Abstract

Call-routing is the technology of automatically classifying the type of a telephone call from a customer to a business or an institution in order to transmit the call onward to the correct “destination”. Making transcriptions of calls to provide training data for automatic routing in a particular application requires considerable human effort, and it would be highly advantageous for the system to be able to learn how to route calls from training utterances that were not transcribed. This paper introduces several techniques that can be used to build call routers from an untranscribed training set, and also without any prior knowledge of the application vocabulary or grammar. The techniques concentrate on identifying sequences of decoded phones that are salient for routing, and introduces two methods for doing this using language models that are specifically tailored for the routing task. Despite the fact that the phone recognition error-rate on the calls is over 70%, the best system described here achieves a routing error of 13.5% on an 18 route task.

1 Introduction

The aim of call-routing is to provide an “automated operator” to deal with incoming telephone calls to a business. A call-router uses speech and language processing techniques to classify a call as one of a small number of call-types, and then transmits it to the appropriate “destination”. The destination might be another automatic system if the transaction required by the caller is a simple one, or a human operator for a sensitive or complex transaction. For example, in an application where the router receives calls associated with a credit-card account, the utterance “I lost my card” would be routed to LostCard and the utterance “What is my account balance?” to Balance. In this paper, we assume that routing must be done on the basis of a single utterance from the caller i.e. the system has no dialogue facility.

Call-routing was pioneered by Gorin and his colleagues at AT&T for use in the AT&T network, resulting in a system called “How May I Help You?”.  (Gorin, Riccardi, &
Qiang and Cox

Wright, 1997; Gorin, Petrovska-Delacretaz, Riccardi, & Wright, 1999; Gorin, Hanel, Rose, & Miller, 1994). Their (initial) approach was to extract salient phrase fragments from transcriptions of callers’ requests and incorporate these into the finite state language model in their speech recognizer. Call classification was done by computing a posteriori probabilities for all possible call types, or by passing the weighted fragments through a neural network classifier. Later, Wright et. al. introduced a method of acquisition of syntactic and semantic fragments from the word sequence using a greedy detection algorithm (Wright, Gorin, & Riccardi, 1997). In 1999, Chu Carroll and Carpenter applied vector-based information retrieval technology to call routing by treating each destination as a “document” and representing the utterance within a vector space (Chu-Carroll & Carpenter, 1999). Cox refined this method by applying Linear Discriminant Analysis (LDA) (Cox & Shahshahani, 2001b) and incorporating confidence measures (?). The technique was further improved by Kuo and Lee, who applied discriminative learning of destinations (Kuo & Lee, 2000). Tur et. al. at AT&T have explored various techniques for exploiting only a subset of the labelled training data to build a router (Tur & Hakkani-Tr, 2003; Tur, Schapire, & Hakkani-Tr, 2003).

Commercial systems that perform call-routing are now in use, but they currently require considerable human effort to transcribe example calls from a particular application for training purposes. By contrast, the task of labelling each example call with its destination can be performed quickly by an expert in the application. It would thus be of great utility to be able to train a call router using a set of example utterances and their associated destinations, without any actual transcriptions of the utterances. However, this task is much more difficult when transcriptions are unavailable, as in the absence of any information about the vocabulary and syntax of the application, it is necessary to use phoneme recognition. Phoneme recognition is inherently less accurate than word recognition, and is especially difficult in call-routing applications, owing to the poor quality of the telephone speech signals. The signals are subject to effects of distortion, bandwidth restriction, electronic and environmental noise, and the difficulties caused by these effects are compounded by the fact that callers often speak casually and spontaneously, and sometimes with a strong accent.

In this paper, we present techniques that enable the basic problem caused by the inac-
accuracy of the phone transcriptions to be addressed. The key idea is that we can assign auto-
matically a probability of association of segments of the call waveform with destinations,
without any requirement to estimate word or phrase identities from the noisy phonetic de-
coding. The paper is organized as follows: section 2 gives the details of the data, baseline
recogniser and router used in our experiments. In section 3, a technique for finding salient
sequences of phones that is based purely on segmentation and clustering of phone streams
output by the recogniser is described. Section 4 explores the use of “mixture language
models”, in which individual phonotactic language models (PLMs) are combined with a
general PLM. In section 5, a more extreme version of this technique is utilised in which
multiple route-specific PLMs (one for each route) are used together with acoustic match-
ing to identify salient sequences more accurately. We conclude with a discussion in section
6 and some suggestions for further work in this field.

2 Data, Baseline Recogniser and Router

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. All utterances were
transcribed and labelled with a single route. For these experiments, we used a set of
4511 transcriptions of utterances for training and 3518 for testing, in which 18 different
call types were represented. Some of these call types were quite easily confusible e.g.
PaymentDue and PaymentDate, PaymentAddress and ChangeAddress. For the experiments
described in section 3 only, extra utterances were used to build a language model.

Phoneme recognition of the input speech queries was performed using an HMM recogn-
iser whose acoustic models had been trained on a large corpus of telephone speech and
which had separate models for males and females. In later experiments, transcriptions
from the Wall Street Journal (WSJ) database were used to generate an initial 6-gram PLM,
with standard back-off procedures when data was sparse. The baseline accuracy of the
recogniser depended crucially on the type of PLM used and the data on which this model
was based, and details are also given in each section.

The vector-based approach to call routing is used in this work. A word-based version
of this technique has been described in e.g. (Cox & Shahshahani, 2001b). Using the training material, a matrix $W$ is formed in which the rows correspond to different words or sequences of words in the vocabulary ("terms"), and the columns to the different routes. Element $W_{ij}$ is the number of times term $t_i$ occurred in route $r_j$. $W$ is then weighted using a weighting scheme that emphasizes terms that are useful for identifying a route and de-emphasizes terms that are not. The weighting used here is due to Bellegarda (Bellegarda, 1998) and is as follows:

$$W_{ij} \rightarrow \log \left( \frac{W_{ij}}{n_j} \right) \left( 1 + \frac{1}{\log N_r} \sum_{j=1}^{N_r} \Pr(r_j|t_i) \log(\Pr(r_j|t_i)) \right).$$

In equation 1,

- $N_r = \text{number of routes (columns of } W \text{ matrix)}$
- $n_j = \text{number of terms occurring in route } r_j$

The first bracketed term in equation 1 is sometimes known as “inverse document weighting” in the field of Information Retrieval. It gives a greater weight to elements in a route that has fewer different terms associated with it. The second bracketed term is one minus the normalized conditional entropy of the route given term $t_i$. It ranges from zero to one, having a value of zero if $t_i$ is equiprobable in every route, and a value of one if $t_i$ occurs only with a single route.

The process of routing a new query is as follows:

1. the query is represented as an additional column vector of $W$;
2. the complete new matrix $W$ is re-weighted according to equation 1;
3. the new column vector is matched to the other column vectors in $W$ using an appropriate metric (dot product was used in these experiments);
4. the route assigned to the query is the route corresponding to the column vector of $W$ that is most similar to the query vector. In this work, the dot product distance between the vectors was used as a measure of similarity.

In this paper, we use “salient sequences” of phones derived from the recogniser output, rather than words, as the terms. By “salient sequences”, we mean sequences of phonemes that are useful for classifying a call: a sequence that occurs frequently but only ever in a
certain route is highly salient, whereas a sequence that occurs in every route is not salient. This property is captured by combining the estimated mutual information between the sequence and the routes and the number of occurrences of the sequence.

3 Iterative Language Modelling of Salient Sequences

3.1 Overview

Our first approach to assigning routes to noisy phonetic sequences was based on iterative refinement of an initial set of salient sequences. Using the phonetic recogniser output, a set of salient sequences can be postulated. A bigram model of these sequences is then estimated and the utterances are re-segmented using this model, producing a new set of sequences. This process is iterated. Salient sequences are also clustered, and all sequences within the same cluster are replaced by the central sequence of that cluster. When there is no change in the set of postulated salient sequences, the iteration is halted. A record is kept of the route of the phrase from which each sequence originated, and this enables the matrix $W$ to be constructed as explained in section 2. Finally, linear discriminant analysis (LDA) is applied to $W$ to enhance classification accuracy.

3.2 Data and Language modelling

For the experiments described in this section (only), phoneme transcriptions of 8000 further utterances (containing a total of about 236 000 phonemes) were used to train n-gram phonotactic language models (PLMs) for recognition. These 8000 utterances were distinct from the 4511 used for training the router. This procedure is, of course, incompatible with our stated aim of making the routing process fully automatic, as transcriptions are required to build the PLM. However, in our initial experiments, we wished to see how well some basic techniques would perform, and this demanded a reasonable level of recognition and hence a good PLM for the recogniser. In later work, the PLM was built from material other than the transcriptions (see sections 4 and 5). 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%, with similar numbers of deletions and insertions.
We experimented with two PLMs: PLM1 was built by using a dictionary to transcribe each word in an utterance as a phoneme string, and concatenating the strings (if multiple pronunciations were present in the dictionary, the most likely one was used). PLM2 was built in the same way but with a silence included between each word. Recognition results for these two models, for different length n-grams, are given in Figure 1.

Figure 1 about here.

Figure 1 shows that recognition accuracy is higher when PLM1 is used and peaks for an N-gram of length $N = 7$. However, it was found that when PLM2 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 PLM1 using the word boundaries found by using PLM2. A typical result of this process is shown in Figure 2.

Figure 2 about here.

The dashes in the recognised output of Figure 2 indicate where the recogniser has found breaks. Although the phone string of the recognised output shown here is inaccurate, the word breaks found are mostly correct, and this was found to be the case in general.

### 3.3 Iterative Algorithm

The algorithm to discover salient sequences is based on that described in (Levitt, Gorin, & Wright, 2001). Firstly, an initial set of salient sequences is estimated using the phone sequences output by the recogniser that have been segmented using PLM1 and PLM2 as described in section 3.2. This is done in the following way:

For each utterance
   For each segment in the utterance
      Find all possible segmentations of the segment
      into 3, 4, ..., 9 phonemes.
   End
Find the mutual information (MI) between each (new) segment and the routes. Exclude all segments whose number of occurrences \(< T_O \) and whose MI \(< T_{MI} \).

Here, \( T_O \) and \( T_{MI} \) are thresholds that are manually set (we used \( T_O = 4 \) and \( T_{MI} = 0.8 \)). Because of recogniser errors, the process above generates a very large number of different sequences, but there is a core of sequences that occur many times, and some of these sequences tend to be associated with a small set of routes (perhaps only one). These sequences form the initial set of salient sequences. The MI of a sequence, i.e. a term \( t_i \), is estimated as

\[
MI(t_i) = \sum_j \Pr(r_j) \log \frac{\Pr(t_i, r_j)}{\Pr(t_i) \Pr(r_j)}
\]

(2)

The initial set of salient sequences is then refined iteratively. The algorithm is explained graphically in figure 3.

Two methods of constraining the search for appropriate sequences are employed:

1. use of a bigram “language” model of sequences to guide re-segmentation of the output of the recogniser.
2. clustering and then correction of sequences that are noisy versions of the same underlying sequence;

The bigram PLM operates on the lattice output by the recogniser and constrains the selection of sequences to those that are most consistent with its parameters. The sequences are further constrained by the clustering process. For clustering, a modified form of the Levenstein Distance is used to estimate a “score” between every pair of unique sequences (salient and non-salient). This modified Levenstein distance attempts to address the problem of comparing the lengths of strings with sub-strings, and hence, indirectly, morphological variants of a word. If, for instance, \( r\,i\,y\,p\,l\,e\,y \) (“replay”) is compared with \( r\,i\,y\,p\,l\,e\,y\,s\,m\,e\,h\,n\,t \) (“replacement”), the Levenstein distance is 5, even though
riypley is a substring of r i y p l e y s m e h n t. However, if r i y p l e y is compared with r i y p e a ("repair") the distance is 2, despite the fact that the two strings are different words. We use a similarity score rather than the Levenshtein distance, in which matching rather than differing symbols are counted: the higher this score, the closer the strings match. A matrix is formed in which element $i, j$ of the matrix is the score between sequences $s_i$ and $s_j$, and this is used as the basis for clustering the sequences, which is done using a standard k-means algorithm. After clustering, sequences are “corrected” by rewriting all the sequences mapped to a certain cluster as the sequence that is identified as the “most central sequence” in the cluster. This identification is based on the “minimax” criterion (Wilpon & Rabiner, 1985).

It was of interest to observe the correspondence between the salient sequences and the transcribed words as iteration was performed. For each identified salient phoneme sequence, the closest word or phrase in the vocabulary was found by applying the similarity score described above to the dictionary transcriptions of words and commonly-occurring phrases. The probability distributions over the salient sequences and over their corresponding words/phrases were estimated, and the Kullback-Leibler (KL) distance between these two distributions estimated. Figure 4 shows the difference between the KL distance at the first iteration and the KL distance at the fifth iteration. This difference is almost always negative, indicating that the probability distributions of the paired salient sequences and words/phrases have become more similar as iteration proceeds. This implies that the mapping from salient sequences to words/phrases improves over most sequences during iteration. From this, we can infer that the technique is automatically learning salient words/phrases from the recogniser output.

Finally, linear discriminant analysis (LDA) is applied to the matrix $W$. Applying LDA to $W$ was found to be useful in improving routing accuracy in (Cox & Shahshahani, 2001b). It also has the beneficial effect of reducing the dimensionality of the column vectors of $W$ to $N_r − 1 = 18$ in our case and hence of reducing considerably the computational task.
3.4 Results

Experimental results comparing the routing accuracy using the transcribed words with the accuracy obtained using phoneme salient sequences are given in Table 1. This table shows that the error-rate using salient sequences is many times that obtained using the word transcriptions.

Table 1 about here.

Table 2 shows how the routing error-rate changes as the iterative refinement of salient sequences described above proceeds. The training-set shows the most marked improvement, indicating that the iterative process is capable of recovering some salient words and phrases that were used to train the classifier. Improvement is less marked for the test-set, but is still very significant. However, at 46.9%, the error-rate is still far too high for the system to be useful.

Table 2 about here.

4 Use of Mixture Language Models

4.1 Overview

The vocabulary and to some extent the syntax used within utterances that are associated with a particular route can be characterised as a mixture of items that are common to all routes (e.g. “Can you tell me...”) with items that are specific to the route (“...my current balance”). This suggests that a mixture of a common “background” language model and a route-specific language model may be more effective at decoding in this task than a single language model. Mixture language models have been used extensively in information retrieval (Lavrenko, 2002; Zhai & Lafferty, 2001), and the idea of mixing background and document (route) specific models is explored in (Miller, Leek, & Schwartz, 1999) and in (Yokoyama, Shinozaki, Iwano, & Furui, 2003). A difference between this work and the work reported here is that we use as our language model phone sequences derived automatically from the recogniser output rather than word or phrase transcriptions.
In this section, we also describe how we dispensed with the prop of using transcriptions to build a PLM, and performed the whole routing process automatically using PLMs derived from the recogniser output. For recognition, we used the technique of an iterated PLM first proposed in (Alshawi, 2003) and extended in (Huang & Cox, 2004). A seed phonotactic PLM derived from a phonetic transcription of a corpus of speech (the Wall Street Journal (WSJ) corpus) is used to enable an initial recognition pass on the data. The output strings from the recogniser are then themselves used to refine the PLM, and this process is iterated. The process is fully described in section 4.3.

4.2 Training and Testing Procedure

The mixture of PLMs used here is a simple interpolative one:

$$P_j^r(k) = \lambda P_j(k) + (1 - \lambda)P_G(k)$$

where $P_j^r(k)$ is the probability of the $k$’th sequence in the mixture (adapted) route-specific PLM (MRS-PLM) for the $j$’th route, $P_j(k)$ is the corresponding probability for the original route-specific PLM (RS-PLM) for the $j$’th route, $P_G(k)$ the corresponding probability for the general (background) PLM (G-PLM) and $\lambda$ an interpolation coefficient. Notice that all phone sequences within an RS-PLM will also be present in the G-PLM, so equation 3 can always be applied. The parameter $\lambda$ was set to be the same for all routes and is selected by maximising the routing accuracy on the training-set.

The main steps in building and using the model proposed here are outlined below.

Training:

1. Build a G-PLM based on the complete set of utterances from all call types and iterate to improve phone recognition performance—this is described in detail in section 4.3.
2. Use the utterances from route $r_j$ and the same technique as above to build a set of $N_r$ RS-PLMs.
3. Combine the specific and the general models using equation 3 to produce a set of $N_r$ MRS-PLMs.
4. Build a set of $N_r$ speech recognisers, each with a different MRS-PLM.
5. Recognise the utterances associated with route $r_j$ using the recogniser trained with MRS-PLM$_j$. 

10
6. Cluster phone sequences in the output to obtain a set of clustered keywords (as described in section 3.3).

7. Train a vector-based call-routing classifier (as described in section 2).

Testing:

1. Recognize the unknown utterance using each of the $N_r$ recognisers.

2. Reduce the recognized phoneme sequences from each recogniser to a sequence of cluster identities.

3. Using the vector-based call-routing classifier, classify the phone sequence from the $j$'th recogniser, $j = 1, 2, \ldots, N_r$.

4. Pool the resulting $N_r$ classifications, and classify the unknown utterance as the class most heavily represented in this set.

4.3 Construction of Salient Sequences using an Iterative Phonotactic Language Model

Figure 5 shows schematically how a PLM is iterated. The recogniser is initially run with a 6-gram PLM built from a phonetic transcription of the sentences in the WSJ corpus. Such a PLM functions as a basic model for the phoneme statistics of spoken English as the WSJ utterances have only a small overlap with most of the utterances spoken in the routing data. However, it enables a first recognition pass to be made on the routing utterances. The output of this recognition pass is then used to train a new 6-gram PLM, which is substituted for the previous PLM, and this process is iterated until there is no change in accuracy of the output strings or until a maximum number of iterations has been reached. The technique described here differs slightly from that presented in (Alshawi, 2003) in that we use a constant value of $N$ ($N = 6$) for the N-grams rather than starting with a low value of $N$ and increasing $N$ on each iteration. The technique appears to work because, although the recognition accuracy is low on the first pass, there are enough correctly decoded substrings to form the basis for an improved PLM for the second pass; hence the recognition accuracy increases on every iteration, which leads to an improved PLM model etc. etc.
The set of phonetic sequences output after the final iteration of this process is then subject to the same process for identifying salient sequences as was described in section 3.3.

4.4 Results

Table 3 shows the routing error-rates achieved when a G-PLM is used ($\lambda = 0$), RS-PLMs are used ($\lambda = 1$) and MRS-PLMs are used (optimal (global) $\lambda$). The differences between the three different techniques are significant at at least the 1% level.

It is of interest to see how the call-routing accuracy and the phone accuracy are related. Figure 6 shows how the call-classification accuracy for a single route varies with the mixing parameter $\lambda$ of equation 3, and Figure 7 shows the phone recognition accuracy for the same data. Phone accuracy is low when a G-PLM is used, higher when RS-PLMs are used, and peaks when a MRS-PLM with $\lambda = 0.7$ is used. By contrast, routing accuracy is high when a G-PLM is used, lower when RS-PLMs are used, and peaks when a MRS-PLM with $\lambda = 0.3$ is used. One might expect the call classification accuracy to follow the phone recognition accuracy, but in practice, although recognition accuracy is higher when $\lambda = 1$, there are many more insertion errors and these affect the call classification performance adversely. The improvement shown by MRS-PLMs when $\lambda = 0.3$ compared with a G-PLM is due to better recognition of salient phoneme sequences. Compared with RS-PLMs, there is actually a little degradation in recognition performance on salient phoneme sequences. However, there is also better recognition performance on other phonemes, and the overall effect on accuracy is slightly positive.

Table 6 about here.

Table 7 about here.
5 A Two-Pass System Using Route-Specific Language Models

5.1 Overview

The results of the previous section showed that although RS-PLMs gave better performance than a G-PLM, they “over-generate” i.e. they decode every input utterance as a phone sequence particular to their own route, so that any correctlydecoded sequences in the output are swamped by false positives. This, of course, was expected, since any utterance, when processed using a PLM specific to a certain route, will be decoded in terms of the vocabulary and syntax specific to that route. We therefore sought a way of detecting and eradicating the false positives produced by this technique whilst at the same time preserving the correctly decoded sequences. This problem could be addressed by adjustment of the “grammar factor” which balances the acoustic and language model probabilities during recognition, but we have found that using an extra stage of acoustic matching to validate the output works better. This technique works as follows.

During the estimation of salient sequences described in section 4.3, the sequence of frames corresponding to each phonetic sequence processed is stored, and after the final iteration of the algorithm of section 3.3, the frame-sequences associated with a certain cluster are used to form an HMM. Each salient phonetic sequence was modelled by a five-state left-to-right HMM with no skips and each state characterised by a mixture Gaussian state observation density. A maximum of 3 mixture components per state was used. The Baum-Welch algorithm was used to estimate the parameters of the Gaussian densities of the HMMs. HMMs for 41 salient phonetic sequences whose number of occurrences was larger than a threshold (we used 30) were built and stored. At recognition time, these HMMs are used to decode the speech frames corresponding to salient sequences detected by an RS-PLM. A sequence that matches well to the appropriate HMM is retained as being correctly decoded, but a sequence that matches poorly is rejected as a “false positive”. Further details are in section 5.2.
5.2 Routing

For recognition, we use a "divide and conquer" approach. Using a G-PLM in the recogniser, utterances that produce one or more salient sequences in the recognition output that are identified with the same route, $r^*$, are classified immediately as $r^*$. Utterances whose output is ambiguous, in that

1. they produce no salient sequences, or,
2. they produce salient sequences associated with more than one route, or,
3. their recognition confidence is too low to trust

are subject to a more detailed recognition pass in which separate PLMs for each route are used. This has the advantage of only applying the extra computational effort required to use multiple PLMs for those utterances that need this. In practice, if lattices are used, the additional computational effort is not too great. The recognition process is shown diagrammatically, together with the number of utterances from the test-set that were involved at each stage, in figure 8.

Table 8 about here.

Recognition/routing using multiple route-specific PLMs works as follows. 18 recognized phonetic sequences are output, one from each recognizer, and salient phonetic sequences may or may not be detected in each recognizer output. The HMMs described in section 5.1 are then used to recognise the section of the utterance waveform that corresponds to a hypothesized salient sequence. Validation of phrases is achieved by using a form of key phrase detection as described in (Rohlick et al., 1993; Kawahara, Lee, & Juang, 1998). The forward probability of the data given the model at any time is monitored and a search made for peaks. If full-likelihood recognition is used, we estimate the score:

$$S_f(w, t) = \frac{\alpha(e_w, t)}{\sum_s \alpha(s, t)}.$$  \hspace{1cm} (4)

where $e_w$ is the last state of salient sequence $w$ and the summation is over all states $s$. We search for local maxima in these scores profiles to determine putative salient phoneme sequences. (Rohlicek et al., 1993).
5.3 Results

Figure 9 compares the phone accuracies when

1. a single PLM built from recogniser output is iterated;
2. 18 RS-PLMS built from recogniser outputs are iterated;
3. a single PLM built from dictionary transcriptions is iterated.

It is interesting that performance decreases with iteration when the transcriptions are used, but increases when the recognised strings are used. This is probably because, when the recognised strings are used, the initial PLM, which is trained on WSJ transcriptions, does not reflect the distribution of n-grams in the data, and so performance is poor. However, the vocabulary in the data is quite small, so that after even a single recognition pass, although the error-rate is high, the new PLM is a better reflection of the n-grams in the data. This has the effect of improving the phone recognition performance, and this improvement continues with each iteration. When an initial language model built using dictionary phoneme transcriptions is used, the performance is initially much better than using an LM trained on an independent corpus, as would be expected. However, because of the small vocabulary size and the relatively high number of occurrences of a few phonetic sequences, any errors in recognition of these sequences dominate, and this leads to an increasing overall error-rate. These results are not as good as those reported in (Alshawi, 2003) using an iterative language model. This may be because of the difference in the speech recognisers, or, more likely, in the average length of the phrases in the different vocabularies, which are much shorter than the phrases used here.

Table 4 shows the call-routing classification error-rate when a single PLM is used and the PLM is iterated. What is of note here is that an apparently small decrease in phone error-rate on iteration gives rise to a large decrease in call-routing error-rate. This is because although the overall phone error-rate improves only slightly, the error-rate on the key phonetic sequences is greatly improved, leading to improved classification performance. The
error-rate on this dataset when the dictionary-derived translations of the transcriptions of the utterances are used is 6.3%. It is interesting to note that this is better than performance obtained when words themselves are used for routing (7.52%). We attribute this to the fact that we did not use any stemming in our word-based router, and when words are represented as phoneme strings, some substrings capture the base forms of inflected forms that were excluded on grounds of frequency from the word-based router. (A similar finding for character- vs. word-based text classification was noted in (Peng & Schuurmans, 2003.).)

Table 5 compares the call-routing classification accuracies. The error-rate achieved using the two pass system (G-PLM + RS-PLMs), 13.5%, is much better than that using a G-PLM, 17.4%, but still double that obtained by using the dictionary transcriptions.

It could be argued that it is not possible to say whether the improvement shown in column 4 of Table 5 compared with column 3 is due to the use of multiple PLMs or to the use of the HMM acoustic post-processing. However, when a single PLM is used, the situation is that either there are one or more fairly unambiguous output sequences from a single call type, or there are many noisy and ambiguous sequences whose positions are not well-defined. It is very difficult to process these putative sequences with the HMMs of key phonetic sequences. Using multiple LMs has the effect of producing a set of relatively unambiguous sequences from only a small subset of call-types, whose position in the waveform is quite well-defined. This reduces the number of HMM sequences that need to be used and hence also the difficulty of application.

6 Summary and Discussion

In this paper, we have addressed the problem of building a call-routing system using a set of utterances whose destinations are known but which are not transcribed. Initial experiments that did use transcribed utterances to build a phonotactic language model (PLM) for recognition purposes indicated that it was possible to estimate and to cluster salient phone sequences from the output of the recogniser and use these to build an effective router. These phone sequences were demonstrated to have a similarity to phonetic transcriptions of the words and phrases in the transcriptions. Subsequent experiments successfully re-
placed the PLM derived from call transcriptions with a model that was initially estimated from transcriptions of an independent text corpus, and then iteratively refined. This made the routing process truly transcription-free. Two techniques for improving the call-routing accuracy were then investigated, both of them relying on using route-specific language models and using multiple decodings of the input utterance, one for each route. The first of these used a mixture language model, in which a poorly-estimated model of the vocabulary and syntax specific to a certain route is smoothed using a general “background” language model that models phrases common to all routes, and which is better estimated. The average routing error-rate on an 18 call-type task using mixture language models was improved from a baseline of 27.4% error when a general language model was used to 22.1% with a mixture model. However, a more successful technique used a two-pass system. Utterances were first recognised using a general PLM, and those that gave recogniser output indicating an unambiguous choice of a single route were classified. The remaining utterances were classified using unsmoothed route-specific language models for recognition, and the output from each recogniser was then re-analysed using a set of HMMs that modelled salient phone sequences. “False positives” (incorrectly hypothesized salient phone sequences) were then identified in the recogniser outputs and removed, and utterances could then be classified. Using this approach, routing error-rate fell to 13.5%. This is still over twice the error-rate that can be obtained when transcriptions of the training utterances are available (6.3%), but a considerable improvement on our initial system that had an error-rate of 46.93% even though it used transcriptions to build the PLM. A surprising outcome of this work is that high-accuracy routing can be achieved when the accuracy of the phone recogniser is very low, in our case only about 30%. Of course, the accuracy here is measured relative to a reference that consists of phoneme strings concatenated from dictionary transcriptions of words, and these transcriptions would not reflect the actual transcriptions of the spontaneously and rapidly spoken speech that we were using. But even allowing for this, the recogniser accuracy is very low. The two-pass technique developed here circumvents this difficulty by using route-specific language models to decode the speech. This forces the decoding to be made in terms of phone strings that are particular to a route, and the second pass uses acoustic models to discard spuriously identified sequences.

Future work will concentrate on adapting and modifying the ideas used in this paper.
for use in a somewhat easier (and more realistic) scenario in which a small proportion of the training utterances available have been transcribed, as in (Giuliani & Federico, 2001). We intend to exploit further the idea of a two- (or multi-) pass system in which “easy” utterances are classified immediately and more sophisticated classifiers are used to classify more complex utterances, and also to identify utterances that cannot be classified unambiguously into one of the destination routes. It would also be interesting to see whether similar techniques could be used to learn language models suitable for use in a full dialogue system.

ACKNOWLEDGMENT

We thank Ben Shahshahani of Nuance Communications for providing the data for these experiments.

References


Figure 1: Phone Accuracy as a function of n-gram length for LM1 (no silences included between words) and LM2 (silences included).

![Graph showing phone accuracy as a function of n-gram length for LM1 (no silences included between words) and LM2 (silences included).]

Figure 2: Comparison of dictionary transcription and recognition output with breaks

Word Transcription: I would like to change my credit limit.
Dictionary Transcription: ay w uh l ay k t uw ch ey n jh m ay k r eh d ih t l ih m ih t
Recognised Output: h aw - l iy k - t aa - ch ey n t - m ey - k eh t ih t - l ih m ih t

Table 1: Routing error-rate on training- and test-set, word transcriptions and phoneme salient sequences

<table>
<thead>
<tr>
<th></th>
<th>Error-rate using transcribed words</th>
<th>Error-rate using phoneme salient sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training-set</td>
<td>5.61</td>
<td>36.56</td>
</tr>
<tr>
<td>Test-set</td>
<td>7.52</td>
<td>46.93</td>
</tr>
</tbody>
</table>

Table 2: Variation of routing error-rate with iteration on training- and test-set

<table>
<thead>
<tr>
<th>Iteration</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training-set</td>
<td>55.68</td>
<td>39.40</td>
<td>37.65</td>
<td>36.69</td>
<td>36.56</td>
</tr>
<tr>
<td>Test-set</td>
<td>54.32</td>
<td>49.27</td>
<td>47.64</td>
<td>47.12</td>
<td>46.93</td>
</tr>
</tbody>
</table>
Figure 3: An iterative algorithm for refining estimates of salient sequences

- **Initial estimate of salient sequences**
- **Current salient sequences**
- **Converged?**
  - Yes: **END**
  - No: **Build bigram language model**

- **Cluster**
- **Re-segment recogniser output**
Figure 4: The difference of the Kullback-Leibler distance between salient sequences and the corresponding closest word or phrase in the vocabulary at the first iteration and at iteration five.
Qiang and Cox

Figure 5: An iterative algorithm for refining a phonotactic language model

![Algorithm Diagram]

Table 3: Routing error-rate on test-set using a general PLM (G-PLM), route-specific PLMs (RS-PLM) and mixture route-specific PLMs (MRS-PLM)

<table>
<thead>
<tr>
<th>Model</th>
<th>Routing Error-rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-PLM ($\lambda = 0$)</td>
<td>27.4</td>
</tr>
<tr>
<td>RS-PLM ($\lambda = 1$)</td>
<td>42.5</td>
</tr>
<tr>
<td>MRS-PLM (optimum $\lambda$)</td>
<td>22.11</td>
</tr>
</tbody>
</table>

Table 4: Phone recognition error-rate and call routing error-rate

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>Phone Error-rate (%)</th>
<th>Routing Error-rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>74.3</td>
<td>56.7</td>
</tr>
<tr>
<td>2</td>
<td>72.9</td>
<td>39.6</td>
</tr>
<tr>
<td>3</td>
<td>70</td>
<td>30.6</td>
</tr>
<tr>
<td>4</td>
<td>69.4</td>
<td>27.9</td>
</tr>
<tr>
<td>5</td>
<td>69.0</td>
<td>27.4</td>
</tr>
</tbody>
</table>

Table 5: Comparison of routing error-rates

<table>
<thead>
<tr>
<th>Name</th>
<th>Trans-Phone 1 PLM</th>
<th>1 PLM + Multiple PLMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Error-rate (%)</td>
<td>6.3</td>
<td>27.4</td>
</tr>
<tr>
<td></td>
<td>13.5</td>
<td></td>
</tr>
</tbody>
</table>
Figure 6: Call-routing error-rate as a function of the mixing parameter $\lambda$
Figure 7: Phone error-rate as a function of the mixing parameter $\lambda$.
Figure 8: The Recognition Process

Test utterances (3515 utts) → Recogniser with G-PLM → Salient phonetic sequence detector → HMMs of key phonetic sequences → REJECT

No sequences detected (368 utts) → Call-route classify → 487 utts correct 107 utts incorrect

1. No sequences detected
2. Sequences from several routes detected
3. Conf-measure too low
   (962 utts)

All sequences are from a single route (2553 utts) → Call-route classify → 2553 utts correct 0 utts incorrect

Total utts = 3515
# classified correct by 1 LM = 2553
# classified correct by 18 LMs = 487

Figure 9: Phone error rate using different PLMs.