THE USE OF CONFIDENCE MEASURES IN VECTOR BASED CALL-ROUTING

Stephen Cox and Gavin Cawley

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

ABSTRACT

In previous work, we experimented with different techniques of vector-based call routing, using the transcriptions of the queries to compare algorithms. In this paper, we base the routing decisions on the recogniser output rather than transcriptions and examine the use of confidence measures (CMs) to combat the problems caused by the "noise" in the recogniser output. CMs are derived for both the words output from the recogniser and for the routings themselves and are used to investigate improving both routing accuracy and routing confidence. Results are given for a 35 route retail store enquiry-point task. They suggest that although routing error is controlled by the recogniser error-rate, confidence in routing decisions can be improved using these techniques.

1. INTRODUCTION

When a customer contacts a medium-size or large business or institution by telephone, the first stage in the process of answering his or her query is to decide to which department or individual the call should be routed. The goal of call routing technology is to use computational speech and language processing techniques to complete this task automatically. An ideal call routing system would be able to decide correctly the "destination" of any call that a human operator could also route. From the user's point of view, call routing technology is highly preferable to the rigid menu-driven systems that are commonly used today, which require the user to respond using touch-tone keyings or single spoken words or phrases. However, it is a challenging task to automate: because of the deliberately open prompt given to the caller (e.g. "How may I help you?", or "How may I direct your call?"), a wide range of responses is elicited from callers. These responses may be very different in length, ranging from single words to long responses that may be syntactically and semantically complex or ambiguous, and that may use a large vocabulary. The call routing task is made feasible by the fact that the number of possible "destinations" for a call is usually quite low (< 40) and the majority of calls can be unambiguously routed to a single destination.

In two previous papers [8, 6], we considered and tested some alternative techniques for the vector-based approach to call routing. In this approach (described in section 2.1), a spoken query is viewed as a "vector" of words and pattern processing techniques are used to route it to the correct destination. The previous papers used the transcriptions of the spoken utterances to form the vectors that are input to the routing engine. To avoid confusion, we refer to these in this paper as the "true-transcriptions". In this paper, we extend the preliminary studies of the previous two papers by using the output from the speech recogniser, which we refer to as the "recogniser-transcriptions", rather than the true-transcriptions. This makes the experiments more realistic but in doing so, introduces "noise" into the text input into the router in the form of word substitutions, deletions and insertions. Routing performance drops compared with performance obtained using the true-transcriptions—the amount of drop depends on the accuracy of the speech recogniser. To combat this loss of performance, we experiment here with estimating and using confidence measures (CMs) for both the words decoded by the speech recogniser (recognition CMs) *and* the destinations decided by the router (routing CMs). The routing CMs can be used in the same way that word CMs are used in intelligent speech-driven systems e.g. to warn the system of an potentially incorrect decision, perhaps because of ambiguous input, and so prompt it to request the user for more input for clarification and confirmation. The recognition CMs can potentially be used both to improve routing accuracy and to estimate the routing CMs.

This paper is organised as follows: in section 2 we outline the essential techniques behind vector-based call-routing used in these experiments and describe the data used. Section 3 details how transcriptions were processed for call-routing. Section 4 describes how the CMs were calculated: section 4.1 describes the recognition CMs and 4.2 the routing CMs. The experiments with the CMs are described in sections 4.3 and 4.4 and results are given in these sections. Finally, we end with a discussion in section 5.

2. BACKGROUND

2.1. Vector-based call routing

The vector-based approach to call routing has been described in e.g. [4, 6] and is summarised as follows. A matrix W is formed using either the true-transcriptions or recogniser-transcriptions of the queries available to train the system. The true (intended) destination of each query is provided by an expert who has listened to the query. The rows of W correspond to different words (or sequences of words) in the vocabulary, and are usually called the *terms*. In the experiments performed here, the columns correspond to the different routes and are usually called the *documents*, a term originating in information retrieval. Element $W(i, j)$ is the number of times term t_i occurred in document d_j. W is then weighted using a weighting scheme that emphasizes terms that are useful for identifying a route and de-emphasizing terms that are not. In these experiments, W was then further trained to give minimum classification error on the training-set using a discriminative algorithm (see section 3.1). The final version of W is referred to as the *routing-matrix*. To route a new query, it is first represented as an additional column vector of W, weighted, and then matched to the other column vectors in W using an appropriate metric. The route assigned to the query is the route corresponding to the column vector of W that is most similar to the query vector. Note that this approach ignores word order in queries (other than the word order given by using a sequence of words as a single term).

2.2. Application 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, but some destinations were used very infrequently. 95% of the calls were routed to the top 35 routes, and these were the calls used in this study. Each call was transcribed and labelled by an expert with the appropriate destination (e.g. "I need my account balance, please" would be routed to *Balance*, "I lost my card" to *LostCard* etc.). These true-transcriptions were divided into a training-set of 6674 queries and a development set of 4713 queries.

Speech recognition was performed using an HMM recogniser whose recognition models had been trained on a large corpus of telephone speech and which had separate models for males and females. The language model used by the recognition system was a trigram model trained on the 6674 training queries: the vocabulary size of the training-set was 1494 words. The recogniser was configured to output the N-best decodings. N was set to be a maximum of 20, but the number of decodings available varied with the length of the input utterance: short utterances generated only a few hypotheses whereas longer utterances generated all N. The word error-rate is measured as (#substitutions + #insertions + #deletions)/(# words in true-transcription) and is given for the training-set and development-set in Table 1. In Table 1, the

	1-best	N-best
Training-set	32.01	26.69
Development-set	40.97	35.25

Table 1: Word error rates on the training and development sets

N-best error-rate is the error-rate obtained by using the decoding that most closely matches the true-transcription.

3. CALL ROUTING AND CONFIDENCE MEASURES

3.1. Call routing technique

In [6], a number of different discriminative techniques for vector-based call-routing were compared. These included Generalised Probabilistic Descent (GPD), Corrective Training (CT) and Linear Discriminant Analysis (LDA). It was found that the GPD technique due to Kuo and Lee was the most effective. The GPD algorithm minimizes classification error on the training-set by adjusting the model parameters of competing classes—see [9] for a full description of the algorithm. The error-rate using GPD quoted in [6] for testing on the training-set and development set true-transcriptions respectively was 4.67% and 12.08%, but these figures were subsequently reduced to 2.67% and 11.08% by further algorithm optimisation. The experiments reported here all used the GPD algorithm to train the discriminative matrices used for routing.

In [6], we described the use of a "stop-list" of words that do not contribute to identification of a destination and hence are excluded from the list of terms. In keeping with the result reported in [9], we found that using a stop-list offered no improvement when using GPD because the algorithm automatically adjusts the weights of terms so that terms that do not contribute to classification accuracy are given a low weight. Using sequences of N words (N-grams) as terms increases performance over using single words, but this technique has the disadvantage that it increases the number of terms considerably, which increases computation time. Using co-locations (word sequences, of variable length, that can be regarded as operating as a single word) suffers from the same problem. Hence in these experiments, no stop-list was used and the terms consisted of single words: there were 1494 terms in the training-set.

After forming the term/document matrix of counts W as described in section 2.1, W is weighted according to the weighting scheme due to Bellegarda [3]. This is a mutual-information based weighting that has been found previously to perform well [8]—details are given in [3] or [8].

The routing module performs route classification by computing the angle between the query test vector and each of the R column vectors of W that represent the 35 different routes and then assigning the query to the route giving the lowest angle.

3.2. Choice of training transcriptions

In previous experiments, we tested our routing algorithms on a set of true-transcriptions, and trained the system on an independent set of true-transcriptions. When testing on the recogniser-transcriptions, the question arises as to whether it is better to train on the recogniser-transcriptions of the training-set queries rather than the true-transcriptions. A priori, this seems an attractive idea: if a word or phrase that occurs in both the training- and development-sets is incorrectly decoded but in a consistent fashion, the result will be the same substituted word or phrase in both sets, and this may mean that routing accuracy is not degraded. Table 2 shows, firstly, that when recognition- rather than

	Router tested with			
	True-trans		Rec-trans	
Router trained on	Train	Dev	Train	Dev
True-trans	2.67	11.08	14.36	21.89
1-best rec-trans			7.54	19.94
N-best rec trans			8.85	20.43

Table 2: Comparison of % routing error-rates under different train/test conditions

true-transcriptions are used, there is a large increase in error-rate. Given the high word error-rates shown in Table 1, this is as expected. However, it also shows that it is advantageous to use the recognition-transcriptions for training. Using true- rather than recogniser-transcriptions gives a large improvement on the training-set (as would be expected), and a significant improvement on the development set. Also, using all N-best transcriptions rather than just the top decoding (1-best) degrades performance. Unless otherwise stated, all experiments reported here used the top recognition-transcription as training for the router.

4. EXPERIMENTS

4.1. Recognition confidence measures

Confidence measures for each word output in the N-best decodings of an input query were estimated using a technique that effectively records the stability of each word within a recognition output lattice [7]. The technique can be described as follows:

For each decoding $D_i, i = 1, 2, \ldots, N$:
  Align D_i with $D_j, j = i+1, 2, \ldots, N$
  using dynamic programming.
  For each word w_i^k in D_i:
    Count the number of times N_i^k that
    word w_i^k in D_i occurs in the same
    position in the other decodings
  end
  Pool all counts of the same word to
  give counts $P_1, P_2, \ldots P_M$. (M = no
  of different words in the decodings).
  $c_l = P_l/(N(N-1)), l = 1, 2, \ldots, M$.
end.

The result of this is that a word that appears in the same position in all N decodings has a confidence $c_l = 1.0$ and a word that appears in only one decoding has $c_l = 1/N(N-1)$. Note that if the same word occurs in different positions in the utterance, it

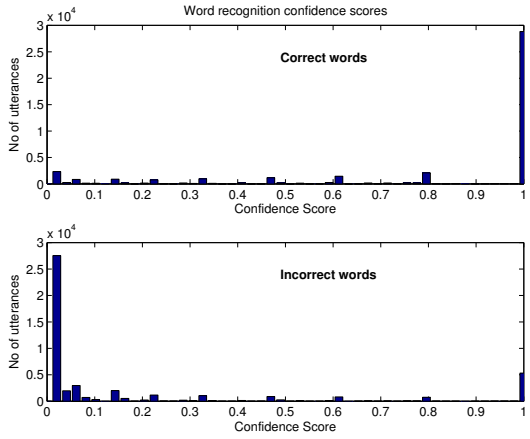

Figure 1: Distributions of recognition confidence scores for correct and incorrect words, all words in N-best decodings

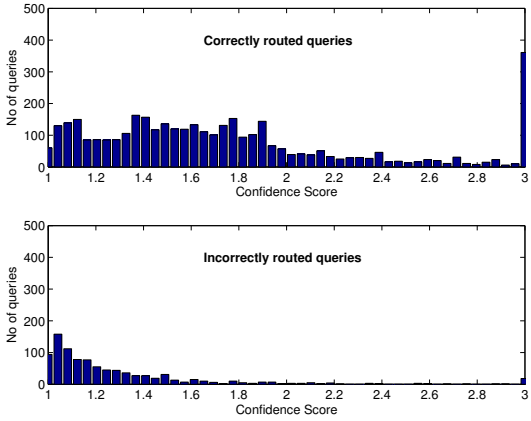

Figure 2: Distributions of routing confidence scores for correctly and incorrectly routed queries in dev. set, top decoding only

is assigned a single confidence measure (maximum value 1.0), as the routing module discards the position of the word within an utterance.

Figure 1 shows the distribution of the recognition confidence scores for correct and incorrect words from all decodings of all utterances in the development-set. Examination of Figure 1 shows that most words (over 70%) have a confidence score of either 1.0 or below 0.05 i.e. the majority of words occur either in all decodings or only once. Furthermore, 67% of correct words have confidence 1.0 and 51% of incorrect words have confidence < 0.05. When all N-best decodings are considered, there are 31085 correct ('C') words and 39249 incorrect ('I'), so by guessing, 55.8% of words would be tagged correctly as either 'C' or 'I'. By classifying all words whose score is above a threshold of 0.52 as 'C' and all others as 'I', this is increased to 78.15%. Hence this CM is a useful tool to aid identification of correctly- and incorrectly-decoded words.

4.2. Routing confidence measures

Each of the N-best decodings of an input query was input to the routing module. The output for each decoding is a set of 35 cosine scores (angles) and each decoding may be classified by assigning to the destination giving the highest cosine score (lowest angle). A confidence score for the top-choice destination associated with the i'th decoding of the j'th input query was estimated as $C_i^j = S_i^j(1)/S_i^j(2)$, where $S_i^j(1)$ is the score of the top choice class and $S_i^j(2)$ the score of the second choice class. When the dot-product is used, as here, $C_i^j \geq 1$.

Figure 2 shows the distribution of the confidence scores for the routes when only the top decoding is routed. The distributions show a clear trend for correctly routed queries to have a higher confidence score than incorrectly routed queries, with a large number of correctly routed queries having a score ≥ 3.0. However, the overlap of the distributions at the low end means that C/I query classification increases from 80.05% by guessing to only 81.37% using the CM, so overall, this CM cannot be described as being very effective.

4.3. Using recognition confidence to increase routing performance

Two ways of using the recognition confidences to boost the routing performance were investigated:

1. Weighting of words by their confidence score.
2. Exclusion of words whose confidence score is below a threshold.

The first technique is based on the premise that when using recognition-transcriptions for routing, the recognition confidence measures described in section 4.1 may be regarded as probabilities that a certain word occurs in the true-transcription of a query. Hence the counts in the routing-matrix W may be replaced by probabilities. The routing-matrix was re-trained using the GPD algorithm, but with recognition confidence measures in the initial matrix rather than counts. For testing, the recognition confidence measures for each word were used (rather than counts) before weighting and matching to the other columns of the matrix. The result was a routing accuracy of 20.43%, about 0.5% worse than using the top decoding (19.94%).

The second technique may increase performance by excluding some incorrect words from the input to the router. However, as the CM is not perfect, some correct words will also be excluded, and some of the excluded incorrect words may not harm routing classification anyway. By varying the threshold, the lowest error-rate obtained was 21.65% when the threshold was set to 0.1, 1.7% higher than using the top decoding with no confidence. We conclude that simply excluding utterances that may be incorrect is not effective.

After this result, it was of interest to "cheat" and find the best performance obtainable using a perfect CM which gives a confidence of 0.0 for incorrect words and a confidence of 1.0 for correct words. This is equivalent to inputting to the router only words in the N-best decodings that appear in the true-transcription. It was tested using both the true-transcriptions and the 1-best recognition-transcriptions to train the router. The results of these experiments were error-rates of 18.19% and 18.49% respectively, which are 1.8/1.5% better than the error-rate obtained using the top decoding with no CM, but still much worse than that obtained using the true-transcriptions for testing (11.08%). The conclusion is that missing words in the N-best recognition-transcriptions are responsible for poor performance, and this problem can clearly only be solved by improving the recogniser. In fact, 85% of the recogniser errors in the figure of 35.25% word error rate (Table 1) are due to substitutions and deletions, which lead to missing words in the recognition output.

4.4. Improving confidence for routing decisions using a classifier

The CM whose score distributions are shown in Figure 2 takes into account only the confidence in the routing assigned to the top decoding of the input query. Just as the recognition confidence measures described in section 4.1 are based on the proportion of occurrences of a word in the N-best decodings, a routing confidence measure can be based on the proportion of occurrences of

a routing in the set of routings of the N-best decodings. A similar but more sophisticated measure is the "diversity" of the routings: our confidence in the routing of a query that produced N different routings from N decodings is clearly lower than our confidence in the routing of a query where all decodings produced the same route. This diversity can be measured using the entropy of the routing decisions: if the N-best decodings produce a set of M routing decisions, $M \leq N$, the entropy E can be computed from estimates of the probability of each different routing, and $0 \leq E \leq \log_2 M$. A feature suitable for use as a CM can then be estimated as $F_{Ent} = 1 - (E/\log_2 M)$: F_{Ent} has a value of 1.0 when all the routings are the same, and 0.0 when they are all different. A further useful feature that can be obtained from the CMs derived from the different decodings of an utterance is F_{mean}, the mean value of the confidence score for the query, C_i^j, as defined in section 4.2.

Both F_{Ent} and F_{mean} give a small decrease in error-rate when used separately to determine whether a routing is correct or not: using F_{Ent}, the error-rate falls to 18.84% and using F_{mean}, to 19.22%, compared with 19.94% when guessing all routes as correct. It is also worth pointing out that 65% of queries have $F_{Ent} = 1.0$ and 89% of the routes assigned to these queries are correct. Inspection of a scatterplot of the two features for correct and incorrect queries showed that F_{Ent} was quantized to only about 30 different values. This suggested using a "piecewise" classifier: for each different F_{Ent} value, determine a threshold on the F_{mean} value that gives optimum C/I classification. We also represented each query as a 2-d feature vector $[F_{Ent}\ F_{mean}]^T$ and tested a support vector machine (SVM) classifier trained on these vectors [5].

However, it was noted that the distributions of the F_{Ent} and F_{mean} features obtained from the training- and development-sets were different. This is probably a consequence of the fact that the training-set queries were used to build the recogniser's language model, and in fact C/I classification error on the training-set queries (14.4%) is much lower than on the development set (19.94%). Because of this difference, when thresholds determined from the training-set were used to classify the development-set, performance was not as good as might be expected if appropriate thresholds were used. To gain a truer picture of the capability of the classifiers, we did cross-validation of the development-set data as follows. The data were divided into 10 equal sets $S_1, S_2, \ldots, S_{10}$, and a training-set consisting of all sets except set S_i made. Parameters for the classifier were estimated from this training-set and tested on set S_i. This was repeated for $i = 1, \ldots, 10$.The error-rates quoted in Table 3 are the mean of the 10 error-rates obtained. When designed using appropriate thresholds, the piecewise classifier gives the lowest error-rate (18.14%). It is perhaps surprising that the SVM classifier performs slightly worse than the simple piecewise classifier—this may be because the SVM is strongly biased towards forming smooth decision boundaries, which is not appropriate for the highly quantized entropy feature.

Classifier	**% C/I error-rate**
No CM classifier	19.94
Threshold on F_{mean} (only)	19.22
Threshold on F_{Ent} (only)	18.84
Piecewise classifier	18.14
SVM, isotropic Gaussian kernel	18.32

Table 3: % C/I classification error-rates on development set queries using the F_{Ent} and F_{mean} features

5. DISCUSSION

In this paper, we have explored the use of two different kinds of confidence measures (CMs) in call-routing: a CM for the words decoded from the query and a CM for the routing. The CM for the decoded words was based on measuring the stability of a word in the N-best list produced by the recogniser and it performed well, increasing the accuracy of tagging a word as "correct" (C) or "incorrect" (I) from 55.8% (by guessing) to 78.15%. However, it was unable to increase routing performance over that obtainable from using the top decoding. We showed that a "perfect" CM that tagged correctly decoded words with 1 and incorrectly decoded words with 0, working on the output from our recogniser, would only decrease routing error slightly (from 19.94% to 18.19%), and concluded that the problem was words missing from the recogniser output, a problem which can only be addressed by increasing the recogniser performance. The CM for routing was based on two features derived from the N-best decodings of a query provided by the recogniser. By combining these features into a CM, the error in tagging a routing decision as C or I was decreased from 19.94% by guessing to 18.14%. These routing error rates are quite high, but it should be pointed out that the queries used in these experiments were labelled with a single destination despite the fact that many were ambiguous and could have been routed by a human to two (and sometimes more than two) destinations. The large increase in routing error when recogniser-transcriptions with a word error-rate of about 40% are used, as opposed to true-transcriptions, is in accord with other results reported in the literature [1, 2]. These show that information retrieval accuracy is maintained until error-rates climb to the 40%–50% area. Our current work is focused on using the constraints of the routing task to improve this accuracy.

ACKNOWLEDGMENT

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

6. REFERENCES

[1] www.informedia.cs.cmu.edu/dli2/talks/aes_9_23_00/.

[2] www.nist.gov/speech/publications/darpa97/html/witbroc1/witbroc1.htm.

[3] J.R. Bellegarda. A multispan language modeling framework for large vocabulary speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6(5):456–467, September 1998.

[4] Jennifer Chu-Carroll and Bob Carpenter. Vector-based natural language call routing. *Computational Linguistics*, 25(3):361–388, 1999.

[5] C. Cortes and V Vapnik. Support vector networks. *Machine Learning*, 20:273 – 297, 1995.

[6] S.J. Cox. Discriminative techniques in call routing. In *Proc. IEEE Conf. on Acoustics, Speech and Signal-processing*, April 2003. (to appear).

[7] S.J. Cox and S Dasmahapatra. High-level approaches to confidence estimation in speech recognition. *IEEE Transactions on Speech and Audio Processing*, 10(7):460–471, November 2002.

[8] S.J. Cox and B. Shahshahani. A comparison of some different techniques for vector based call-routing. In *Proc. 7th European Conf. on Speech Communication and Technology*, September 2001.

[9] H.K.J. Kuo and C Lee. Discriminative training of natural language call-routers. *IEEE Transactions on Speech and Audio Processing*, 11(1):24–35, January 2003.