Improving “How May I Help You?” Systems using the Output of Recognition Lattices

James Allen, David Attwater, Peter Durston, Mark Farrell

BTexact Technologies, Adastral Park, Martlesham Heath, Suffolk, IP5 3RE
(James.3.Allen; David.Attwater; Peter.Durston; Mark.Farrell) @bt.com

Abstract

“How may I help you?” systems where a caller to a call centre is routed to one of a set of destinations using machine recognition of spontaneous natural language is a difficult task. Previous BT “How May I Help You” work [1,2] has used top 1 recognition results for classification with much better results when tested on human transcriptions. Classifying using a recognition lattice was found to reduce the gap between results on transcriptions and recognition output. Using features generated from the lattice in addition to the top 1 recognition results gave an improvement in classification of 4% absolute over a baseline system using only the top 1 recognition result. This reduced the gap between classification performance on recognition and transcription by over 25%.

1. Introduction

Gorin et al.[3] first showed that it was possible to route a caller to one of 16 destinations using classification on a recognition transcript of the caller’s spontaneous natural language after an initial prompt, such as “How may I help you?”. His system used a classifier trained on examples of previous calls to a call centre. This classifier took the recognition result from a new utterance and assigned the utterance to one of a set of classes corresponding to call centre destinations or services.

The goal for these systems is to increase Correct Acceptance (CA) at some False Reject Rate (FRR) where:

$$CA = \frac{\text{Number of correctly classified utterances}}{\text{Number of classified utterances}} \quad (1)$$

$$FRR = \frac{\text{Number of incorrectly rejected utterances}}{\text{Number of rejected utterances}} \quad (2)$$

Improvement in the correct accept rate results in more calls being automatically handled, fewer calls to human operators and greater cost savings. In a similar system developed at BTexact [2,4] top 1 recognition results were used. There was a significant gap in performance between results on transcriptions and results on recognition output. This paper describes classification using recognition lattices as an alternative approach attempting to reduce this gap.

When using the top 1 recognition result information is being lost. The recognition has a large vocabulary, needs to run in near real-time and be able to cope with a range of speakers and accents. This being the case recognition errors such as insertions and deletions are common and can be particularly serious for classification based on multi-word fragments. These errors mean that the recognized utterance contains less information on the caller’s intentions than the original utterance. In fact when comparing classification on recognition with recognition on human transcribed utterances a difference of 10-15% in the correct acceptance measure is not unusual.

It was hoped that by using the recognition lattice, which contains multiple sentence hypotheses, for classification some of the lost information could be retrieved and classification performance improved over that from using the top 1 recognition result alone.

The classifier used was a vector classifier modelled on that used in (Chu-Carroll 2000)[4]. Initially classification is based on:

$$class = \max_i \sum_j tf * idf_{i,j} \qquad (3)$$

where i is the class index, j is the index of a fragment collected from the utterance and $tf*idf_{i,j}$ is a normalized $tf*idf$ score for fragment j and its occurrence in class i.

Recognition made use of the Scansoft SpeechPearl2000 recogniser with a custom triphone acoustic model trained on the training set. A bigram language model – also trained on the training set was used.

2. Lattices

2.1. Structure

For our purposes a lattice consists of a set of nodes connected by arcs in a directed non-cyclical graph (Fig 1). Each node exists at a particular time instance (at any time instance there may be more than one node) and is numbered in time order. There exist “start” and “end” nodes. All paths through the lattice begin at the start node and conclude at the end node.

Nodes are connected by one or more arcs. Each arc is directional and connects an earlier node to a later node. Each arc is labelled with a word (or the tag #PAUSE# indicating silence) and a confidence score for the word assigned by the recogniser.

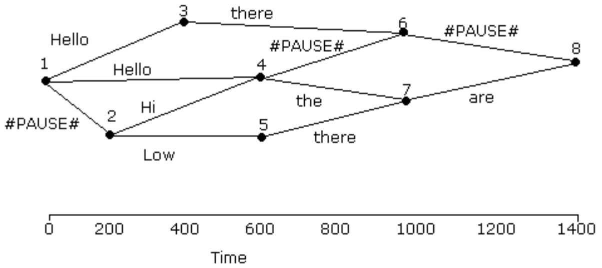

Figure 1: Example Recognition lattice

2.2. Format

The lattice structure shown in Figure 1 is represented by the following file format:

```
BEGIN_LATTICE
1 2 #PAUSE#      0    200  -1.000
1 3 hello        0    400  1.0000
1 4 hello<hel@U> 0    600  1.0000
2 4 hi           200  600  0.6544
2 5 low          200  600  0.0023
3 6 there        400  1000 0.6212
4 6 #PAUSE#      600  1000 -1.000
4 7 the          600  1002 0.0100
5 7 there        600  1002 0.7999
6 8 #PAUSE#      1000 1200 -1.000
7 8 are          1002 1200 0.7232
END_LATTICE
```

Each line corresponds to an arc with the columns corresponding to Start node, End node, Word, Start time, End time and Confidence respectively. The lines are ordered by start-time, to guarantee that later words always start on or after the start time of earlier words. Node numbers are also assigned in monotonically increasing order – thus a lower node number indicates an earlier or identical point in time.

The confidence score is assigned by the recogniser. A score close to 1 indicates high confidence, a score close to 0 indicates low confidence. #PAUSE# tags are not assigned a confidence, indicated by the dummy value –1.

3. Lattice Pruning

The lattices contain a large amount of information that was considered to be redundant for classification purposes. Words may appear multiple times with slightly differing start or end times, the same word might be included with different pronunciations and there are also #PAUSE# tags where the recogniser detected silence. It was decided to remove this redundant information.

First some basic filtering was done. All the #PAUSE# lines were removed and then all variant pronunciations were replaced with the word's base form. Next each word was given a start node, a first end node and a final end node. Given the first occurrence of a wordform its first and final end nodes were set to the end-node of this occurrence.. Subsequent occurrences of this wordform were merged with this initial wordform if the start time of the subsequent word was within one second of the initial wordform. The initial end node was re-assigned if this new word ended prior to the current initial end node. The final end node was also re-assigned if the new word ended after the current final end node.

This left a single entry for each word with its start node being the first node a copy of the word started at. The first end node being the lowest numbered node an instance of the word finished at and the final end node being the latest node a copy of the word finished at. The confidence for the word was defined to be the highest confidence value found for that word.

The resulting lattice/chart structure contains words which potentially overlap one another whilst retaining elements of the original lattice structure.

4. Collecting Fragments

A number of methods for generating fragments for classifier training and testing were used. All of the fragments used for training and test contained pairs of words. In each experiment these word-pair fragments were collected from the pruned lattice in several ways. Analogous methods were used to collect fragments from the top 1 recognition results for comparisons. These features were only used for classification if they appeared in the training set 3 times. Once collected all features were labelled in the form X_Y (with X being the earlier word) and were from this point on treated identically. The features are illustrated in Figure 2.

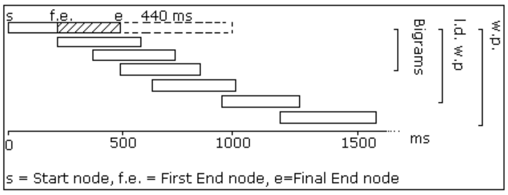

Figure 2: Feature selection from a lattice.

4.1. Bigrams

When classifying top 1 recognition results a bigram is simply any two consecutive words. A "bigram" was generated from the pruned lattice where the start of one word began at or after the first end node of a previous word and at or before the final end node of that previous word.

4.2. Wordpair

When classifying top 1 recognition results a wordpair is simply an ordered pair of words with any arbitrary separation. A "wordpair" could be generated from the pruned lattice where the start of one word began at or after the first end node of a pervious word. A bigram is a type of wordpair.

4.3. Limited Distance Wordpair (l.d.w.p.)

Using top 1 recognition results a limited distance wordpair (l.d.w.p.) was defined to be equivalent to a wordpair that could be separated by zero, one or two words. In the pruned lattice limited distance wordpairs were defined to include all "bigrams" generated as in 3.1 and also any wordpair where the second word started within 440ms of the end of the previous word. The 440ms is based on doubling the average word length of words in the lattice.

An alternative would have been to select word pairs separated by up to 2 intervening nodes. However this was computationally more expensive and this information is lost when pruning the lattice. Without lattice pruning the system would struggle to execute in real-time.

5. Experimental Settings

The corpus was the Oasis corpus used in [2] a set of first utterances spoken to an operator each labelled as one of 16 classes[1]. It consisted of 8000 utterances split into a training set of 7000 utterances and a 1000 utterance test set.

The classifier used in this paper's experiments was a vector classifier based on the one in Chu-Carroll (2000)[4]. Classification matrices were trained on the training data as described in that paper with the exception that reduced dimension matrices were not used as these were not found to significantly affect classification performance.

All examples of a particular feature (e.g. bigrams) were collected from the training set. Classification matrices were then trained using only those features that occurred 3 or more times. The classification matrices used different features: one used bigrams as features, one used wordpairs and the final one used bigrams separated by 0-2 words – i.e. limited distance word pairs. All experiments using bigrams used the bigram classification matrix and so on.

Results [Table 1] are shown as the percentage of correctly accepted (CA) utterances at a 30% false reject rate (FRR).

The false reject rate can be tuned for a particular application with higher FRR generally giving higher correct acceptance. Values between 20-40% FRR have been quoted for systems. Additional results are shown Figures 3 and 4 in the form of ROC curves showing the percentage of correctly accepted utterances against the percentage of falsely rejected utterances as in Gorin et al.[3].

As a baseline for the system classification results using the various features on recognition top 1 are included in the results table.

6. Experiments

6.1.1. Lattice classification.

We first compared the use of analogous features collected from the top 1 recognition or the pruned lattice.

As can be seen in the results table [Table 1] this was disappointing. In addition to the drop in accuracy the number of wordpairs collected from one particular lattice was so large that the classifier had to be modified to limit its use to only 10,000 features generated by an utterance. The corpus contained two classes that were significantly larger than the rest. An examination of the results showed a large proportion of utterances were being misclassified as one of these two most common classes.

It was hypothesised that fragments generated by recognition errors are more likely to appear by chance in the larger classes than in the smaller classes. This implied the significant number of incorrect fragments were overwhelming the number of correct fragments indicating the smaller classes.

6.1.2. Lattice with confidence Weighting

The next step was to add confidence weighting to concentrate on classification using those words the recogniser is most confident about having recognized correctly.

The classification model in (3) was updated to include a weighting term

$$class = \max_i \sum_j tf * idf_{i,j} f(w_j) \qquad (4)$$

where $f(w_j)$ is the confidence allocated to fragment j. The confidence of the two words in the feature was defined to be the weighted average of the two word confidences. This meant that confident features contributed more towards classification than features that the recogniser assigned low confidence to.

This did result in improvements and classification using a lattice generated a higher CA than top 1 recognition at high rates of false rejection (>35-45% depending on the feature used – see Figure 3). However generating the features from the lattice and classifying on them was far more computationally expensive than the top 1 equivalents.

6.1.3. Lattice with confidence weighting and post-collection thresholding

In an effort to reduce the computation time thresholding of the features was introduced. Fragments were filtered such that only fragments with a confidence that exceeded some threshold value were used for classification.

The 7000 utterance training set was further split into 6000 utterances used for training and a range of thresholding values were used in classification of the remaining 1000 utterances, the best threshold proved to be 0.1. This threshold was then used for classification using the full test and training sets. This again resulted in improvements in CA but had little impact on classification computation speed.

6.1.4. Lattice with confidence weighting and pre-collection thresholding

In a further attempt to increase the computational efficiency, words with a confidence below a threshold were removed before beginning to collect features. The development set was used as before and training took place using 6000 utterances. The further 1000 utterances were used to optimise a threshold value for the removal of words. Coincidentally the optimum value turned out to be 0.1 for this form of thresholding as well. This markedly increased classification speed by a factor of 4. This showed an improvement in CA over the baseline bigram and wordpair classifiers for our chosen FRR.

6.1.5. Results for initial tests

As the l.d.w.p. showed the best results in all categories the results for these are shown as ROC curves in Fig. 3 below. The top 1 recognition result using l.d.w.p. classification is included for comparison (Baseline Top 1). The graph labels 6.1.1-6.1.4 refer to the section headings which describe the form of lattice classification used. The range of values of FRR for which the new systems give a higher CA rate than the baseline can be seen clearly here.

[1] This differs from the original Oasis corpus as 3 classes were not found to be useful and were merged with the other/reject class.

Feature Type:	**Matching On**					
	Transcripts	Recognition top 1 **(Baseline)**	Lattice without confidence weighting	Lattice with confidence weighting	Lattice with confidence weighting and post collection thresholding	Lattice with confidence weighting and pre-collection thresholding
Bigrams	85.0	65.9	N/A[1]	64.5	64.2	66.7
l.d.w.p.	89.5	74.3	57.4	70.9	72.2	74.5
Wordpairs	82.1	68.8	52.4	64.1	66.1	71.1

Table 1: Results as Percentage Correct Acceptance at a 30% False Reject Rate.

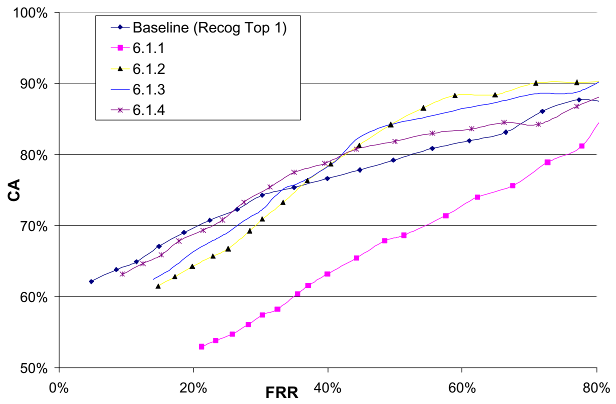

Figure 3: ROC curves for settings of l.d.w.p.

6.1.6. Combined Classification

Finally, to improve performance the scores given to a class by two classifiers were combined by summing the individual scores. The two classifiers used were the lattice based l.d.w.p. using weighting and pre-collection thresholding and the l.d.w.p. classifier using the top1 recognition result (which was the baseline). This summation of the scores, which were the cross products of the features in the utterance against the classification matrix, lead to a change in the top 1 class in many utterances and the improved result can be seen in Fig 4. The top line being the results of l.d.w.p. classification on transcriptions, the middle line being the combined classification result (described above) and the bottom line being the l.d.w.p classifier using the top1 recognition result.

7. Conclusions

Lattices seem to hold promise in reducing the gap between recognition on human transcriptions and recogniser output. The gap between the two was reduced by 4% absolute (25-30% relative) by use of some simple lattice methods in combination with the top 1 recognition result. In particular the use of limited distance wordpairs has proved very successful. The use of these features allows insertions and deletions to be handled more elegantly as in the case of an insertion:

X Y=>X I Y

the co-occurring words X,Y will still appear as a classification feature and in the case of deletions

X Y Z=>X Z

X Z would still potentially be in the classification matrix as a useful feature.

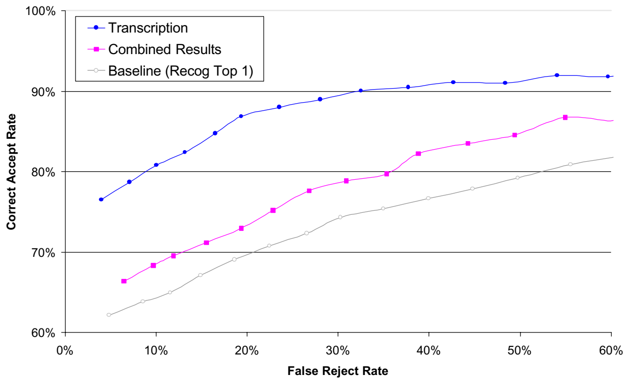

Figure 4: Final ROC curves.

As the greatest improvement was only realised when combining two different classifiers it may be worth spending time developing classifiers on multiple feature sets and combining them.

8. Future Work

Though the l.d.w.p. was successful when used on lattices it was thought that using a probability distribution to weight l.d.w.p. fragments rather than just having a hard cut off at 440 ms may model language more closely and could be worth further investigation. Ways of combining unigram and trigram features in addition to the bigram features used in this paper may also lead to improvement and would be the first avenue of future research.

9. Acknowledgements & Thanks

The authors wish to thank Reinhard Blasig and Dermot Connolly of Scansoft for their support and encouragement.

10. References

[1] Durston , Farrell, Attwater, Allen, Kuo, Afify, Fosler-Lussier, Lee "OASIS Natural Language Call Steering Trial" *Proc. Eurospeech, Denmark Sept 2001.*

[2] Edgington, Attwater and Durston, "OASIS – A framework for Spoken Language Call Steering" *Proceedings Eurospeech 1999*

[3] Gorin, Parker, Sachs "How May I Help You?" *In The Proc. Of Ivtta Philadelphia, October 1996*

[4] Chu-Carroll, Carpenter, "Vector-Based Natural Language Call Routing", *Computational Linguistics., vol. 25, no. 3, pp. 361-388, 1999.*

[1] Lowest FFR for this setting was 39%