Proceedings of HLT-NAACL 2003
Main Papers , pp. 1-7
Edmonton, May-June 2003

Effective Utterance Classification with Unsupervised Phonotactic Models

Hiyan Alshawi
AT&T Labs - Research
Florham Park, NJ 07932, USA
hiyan@research.att.com

Abstract

This paper describes a method for utterance classification that does not require manual transcription of training data. The method combines domain independent acoustic models with off-the-shelf classifiers to give utterance classification performance that is surprisingly close to what can be achieved using conventional word-trigram recognition requiring manual transcription. In our method, unsupervised training is first used to train a phone n-gram model for a particular domain; the output of recognition with this model is then passed to a phone-string classifier. The classification accuracy of the method is evaluated on three different spoken language system domains.

1 Introduction

A major bottleneck in building data-driven speech processing applications is the need to manually transcribe training utterances into words. The resulting corpus of transcribed word strings is then used to train application-specific language models for speech recognition, and in some cases also to train the natural language components of the application. Some of these speech processing applications make use of utterance classification, for example when assigning a call destination to naturally spoken user utterances (Gorin et al., 1997; Carpenter and Chu-Carroll, 1998), or as an initial step in converting speech to actions in spoken interfaces (Alshawi and Douglas, 2001).

In this paper we present an approach to utterance classification that avoids the manual effort of transcribing training utterances into word strings. Instead, only the desired utterance class needs to be associated with each sample utterance. The method combines automatic training of application-specific phonotactic models together with token sequence classifiers. The accuracy of this phone-string utterance classification method turns out to be surprisingly close to what can be achieved by conventional methods involving word-trigram language models that require manual transcription. To quantify this, we present empirical accuracy results from three different call-routing applications comparing our method with conventional utterance classification using word-trigram recognition.

Previous work at AT&T on utterance classification without words used information theoretic metrics to discover "acoustic morphemes" from untranscribed utterances paired with routing destinations (Gorin et al., 1999; Levit et al., 2001; Petrovska-Delacretaz et al., 2000). However, that approach has so far proved impractical: the major obstacle to practical utility was the low runtime detection rate of acoustic morphemes discovered during training. This led to a high false rejection rate (between 40% and 50% for 1-best recognition output) when a word-based classification algorithm (the one described by Wright et. al (1997)) was applied to the detected sequence of acoustic morphemes.

More generally, previous work using phone string (or phone-lattice) recognition has concentrated on tasks involving retrieval of audio or video (Jones et al., 1996; Foote et al., 1997; Ng and Zue, 1998; Choi et al., 1999). In those tasks, performance of phone-based systems was not comparable to the accuracy obtainable from word-based systems, but rather the rationale was avoiding the difficulty of building wide coverage statistical language models for handling the wide range of subject matter that a typical retrieval system, such as a system for retrieving news clips, needs to cover. In the work presented here, the task is somewhat different: the system can automatically learn to identify and act on relatively short phone subsequences that are specific to the speech in a limited domain of discourse, resulting in task accuracy that is comparable to word-based methods.

In section 2 we describe the utterance classification method. Section 3 describes the experimental setup and the data sets used in the experiments. Section 4 presents the main comparison of the performance of the method against a "conventional" approach using manual transcription and word-based models. Section 5 gives some concluding remarks.

2 Utterance Classification Method

2.1 Runtime Operation

The runtime operation of our utterance classification method is simple. It involves applying two models (which are trained as described in the next subsection): A statistical n-gram phonotactic model and a phone string classification model. At runtime, the phonotactic model is used by an automatic speech recognition system to convert a new input utterance into a phone string which is mapped to an output class by applying the classification model. (We will often refer to an output class as an "action", for example transfer to a specific call-routing destination). The configuration at runtime is as shown in Figure 1. More details about the specific recognizer and classifier components used in our experiments are given in the Section 3.

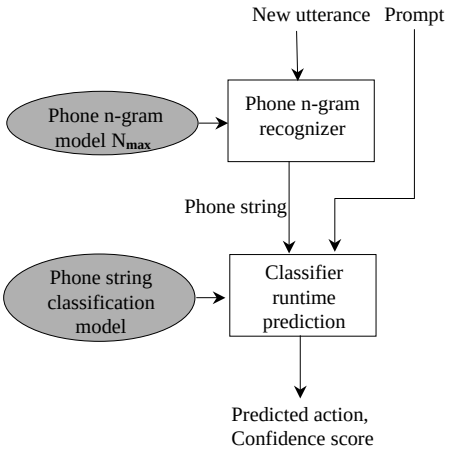

Figure 1: Utterance classifier runtime operation

The classifier can optionally make use of more information about the context of an utterance to improve the accuracy of mapping to actions. As noted in Figure 1, in the experiments presented here, we use a single additional feature as a proxy for the utterance context, specifically, the identity of the spoken prompt that elicited the utterance. It should be noted, however, that inclusion of such additional information is not central to the method: Whether, and how much, context information to include to improve classification accuracy will depend on the application. Other candidate aspects of context may include the dialog state, the day of week, the role of the speaker, and so on.

2.2 Training Procedure

Training is divided into two phases. First, train a phone n-gram model using only the training utterance speech files and a domain-independent acoustic model. Second, train a classification model mapping phone strings and prompts (the classifier inputs) to actions (the classifier outputs).

The recognition training phase is an iterative procedure in which a phone n-gram model is refined successively: The phone strings resulting from the current pass over the speech files are used to construct the phone n-gram model for the next iteration. In other words, this is a "Viterbi re-estimation" or "1-best re-estimation" process. We currently only re-estimate the n-gram model, so the same general-purpose HMM acoustic model is used for ASR decoding in all iterations. Other more expensive n-gram re-estimation methods can be used instead, including ones in which successive n-gram models are re-estimated from n-best or lattice ASR output. Candidates for the initial model used in this procedure are an unweighted phone loop or a general purpose phonotactic model for the language being recognized.

The steps of the training process are as follows. (The procedure is depicted in Figure 2.)

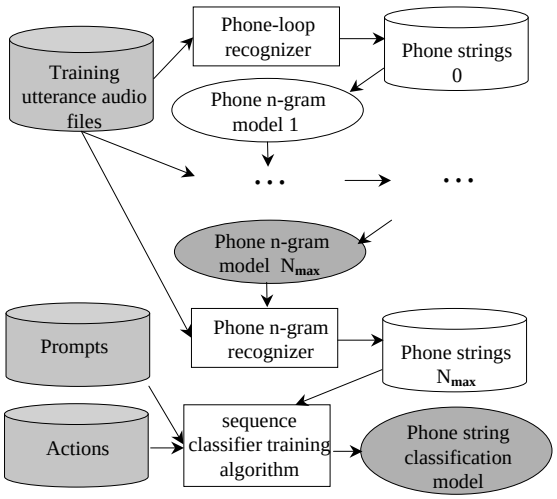

Figure 2: Utterance classifier training procedure

1. Set the phone string model G to an initial phone string model. Initialize the n-gram order N to 1. (Here 'order' means the size of the n-grams, so for example 2 means bi-grams.)

2. Set S to the set of phone strings resulting from recognizing the training speech files with G (after possibly adjusting the insertion penalty, as explained below).

3. Estimate an n-gram model G' of order N from the set of strings S.

4. If $N < N_{max}$, set $N \leftarrow N + 1$ and $G \leftarrow G'$ and go to step 2, otherwise continue with step 5.

5. For each recognized string $s \in S$, construct a classifier input pair (s, r) where r is the prompt that elicited the utterance recognized as s.

6. Train a classification model M to generalize the training function $f : (s, r) \rightarrow a$, where a is the action associated with the utterance recognized as s.

7. Return the classifier model M and the final n-gram model G' as the results of the training procedure.

Instead of increasing the order N of the phone n-gram model during re-estimation, an alternative would be to iterate N_{max} times with a fixed n-gram order, possibly with successively increased weight being given to the language model vs. the acoustic model in ASR decoding.

One issue that arises in the context of unsupervised recognition without transcription is how to adjust recognition parameters that affect the length of recognized strings. In conventional training of recognizers from word transcriptions, a "word insertion penalty" is typically tuned after comparing recognizer output against transcriptions. To address this issue, we estimate the expected speaking rate (in phones per second) for the relevant type of speech (human-computer interaction in these experiments). The token insertion penalty of the recognizer is then adjusted so that the speaking rate for automatically detected speech in a small sample of training data approximates the expected speaking rate.

3 Experimental Setup

3.1 Data

Three collections of utterances from different domains were used in the experiments. Domain A is the one studied in previously cited experiments (Gorin et al., 1999; Levit et al., 2001; Petrovska-Delacretaz et al., 2000). Utterances for domains B and C are from similar interactive spoken natural language systems.

Domain A. The utterances being classified are the customer side of live English conversations between AT&T residential customers and an automated customer care system. This system is open to the public so the number of speakers is large (several thousand). There were 40106 training utterances and 9724 test utterances. The average length of an utterance was 11.29 words. The split between training and test utterances was such that the utterances from a particular call were either all in the training set or all in the test set. There were 56 actions in this domain. Some utterances had more than one action associated with them, the average number of actions associated with an utterance being 1.09.

Domain B. This is a database of utterances from an interactive spoken language application relating to product line information. There were 10470 training utterances and 5005 test utterances. The average length of an utterance was 3.95 words. There were 54 actions in this domain. Some utterances had more than one action associated with them, the average number of actions associated with an utterance being 1.23.

Domain C. This is a database of utterances from an interactive spoken language application relating to consumer order transactions (reviewing order status, etc.) in a limited domain. There were 14355 training utterances and 5000 test utterances. The average length of an utterance was 8.88 words. There were 93 actions in this domain. Some utterances had more than one action associated with them, the average number of actions associated with an utterance being 1.07.

3.2 Recognizer

The same acoustic model was used in all the experiments reported here, i.e. for experiments with both the phone-based and word-based utterance classifiers. This model has 42 phones and uses discriminatively trained 3-state HMMs with 10 Gaussians per state. It uses feature space transformations to reduce the feature space to 60 features prior to discriminative maximum mutual information training. This acoustic model was trained by Andrej Ljolje and is similar to the baseline acoustic model used for experiments with the Switchboard corpus, an earlier version of which is described by Ljolje et al. (2000). (Like the model used here, the baseline model in those experiments does not involve speaker and environment normalizations.)

The n-gram phonotactic models used were represented as weighted finite state automata. These automata (with the exception of the initial unweighted phone loop) were constructed using the stochastic language modeling technique described by Riccardi et al. (1996). This modeling technique, which includes a scheme for backing off to probability estimates for shorter n-grams, was originally designed for language modeling at the word level.

3.3 Classifier

Different possible classification algorithms can be used in our utterance classification method. For the experiments reported here we use the BoosTexter classifier (Schapire and Singer, 2000). Among the alternatives are decision trees (Quinlan, 1993) and support vector machines (Vapnik, 1995). BoosTexter was originally designed for text categorization. It uses the AdaBoost algorithm (Freund and Schapire, 1997; Schapire, 1999), a wide margin machine learning algorithm. At training time, AdaBoost selects features from a specified space of possible features and associates weights with them. A distinguishing characteristic of the AdaBoost algorithm is that it places more emphasis on training examples that are difficult to classify. The algorithm does this by iterating through a number of rounds: at each round, it imposes a distribution on the training data that gives more probability mass to examples that were difficult to classify in the previous round. In our experiments, 500 rounds of boosting were used; each round allows the selection of a new feature and the adjustment of weights associated with existing features. In the experiments, the possible features are identifiers corresponding to prompts, and phone n-grams or word n-grams (for the phone and word-based methods respectively) up to length 4.

3.4 Experimental Conditions

Three experimental conditions are considered. The suffixes (**M** and **H**) in the condition names refer to whether the two training phases (i.e. training for recognition and classification respectively) use inputs produced by machine (**M**) or human (**H**) processing.

PhonesMM This experimental condition is the method described in this paper, so no human transcriptions are used. Unsupervised training from the training speech files is used to build a phone recognition model. The classifier is trained on the phone strings resulting from recognizing the training speech files with this model. At runtime, the classifier is applied to the results of recognizing the test files with this model. The initial recogition model for the unsupervised recognition training process was an unweighted phone loop. The final n-gram order used in the recognition training procedure (N_{max} in section 2) was 5.

WordsHM Human transcriptions of the training speech files are used to build a word trigram model. The classifier is trained on the word strings resulting from recognizing the training speech files with this word trigram model. At runtime, the classifier is applied to the results of recognizing the test files with the word trigram model.

Learned phone sequence	Corresponding words
b ih l ih	**billi**ng
k ao l z	**calls**
n ah m b	**numb**er
f aa n	**phone**
r ey t	**rate**
k ae n s	**canc**el
aa p ax r	**oper**ator
aw t m ay	**what my**
ch eh k	**check**
m ay b	**my b**ill
p ae n ih	com**pany**
s w ih ch	**switch**
er n ae sh	int**ernat**ional
v ax k w	ha**ve a que**stion
l ih ng p	bil**ling p**lan
r ey t s	**rates**
k t uw p	li**ke to p**ay
ae l ax n	b**alan**ce
m er s er	custo**mer ser**vice
r jh f ao	cha**rge fo**r

Table 1: Example phone sequences learned by the training procedure from domain A training speech files.

WordsHH Human transcriptions of the training speech files are used to build a word trigram model. The classifier is trained on the human transcriptions of the speech training files. At runtime, the classifier is applied to the results of recognizing the test files with the word trigram model.

For all three conditions, median recognition and classification time for test data was less than real time (i.e. the duration of test speech files) on current micro-processors. As noted earlier, the acoustic model, the number of boosting rounds, and the use of prompts as an additional classification feature, are the same for all experimental conditions.

3.5 Example learned phone sequences

To give an impression of the kind of phone sequences resulting from the automatic training procedure and applied by the classifier at runtime, see Table 1. The table lists some examples of such phone strings learned from domain A training speech files, together with English words, or parts of words (shown in bold type), they may correspond to. (Of course, the words play no part in the method and are only included for expository purposes.) The phone strings are shown in the DARPA phone alphabet.

Rejection rate (%)	PhoneMM accuracy	WordHM accuracy	WordHH accuracy
0	74.6	76.2	77.0
10	79.5	81.1	81.5
20	84.4	85.8	86.2
30	89.4	90.5	90.9
40	94.1	94.7	94.4
50	97.2	97.3	96.7

Table 2: Phone-based and word-based utterance classification accuracy for domain A

Rejection rate (%)	PhoneMM accuracy	WordHM accuracy	WordHH accuracy
0	80.8	81.6	81.0
10	86.0	86.7	85.3
20	90.0	90.6	89.5
30	93.9	93.7	92.3
40	96.3	96.8	94.7
50	97.5	97.7	96.4

Table 3: Phone-based and word-based utterance classification accuracy for domain B

Rejection rate (%)	PhoneMM accuracy	WordHM accuracy	WordHH accuracy
0	68.2	68.9	69.9
10	73.3	73.7	74.9
20	78.9	79.2	80.2
30	84.8	84.7	85.5
40	89.7	89.3	90.2
50	94.1	93.3	94.5

Table 4: Phone-based and word-based utterance classification accuracy for domain C

4 Classification Accuracy

In this section we compare the accuracy of our phone-string utterance classification method (**PhonesMM**) with methods (**WordsHM** and **WordsHH**) using manual transcription and word string models.

Accuracy Metric

The results are presented as utterance classification rates, specifically the percentage of utterances in the test set for which the predicted action is valid. Here a valid prediction means that the predicted action is the same as one of the actions associated with the test utterance by a human labeler. (As noted in section 3, the average number of actions associated with an utterance was 1.09, 1.23, and 1.07 for domains A, B, and C, respectively.) In this metric we only take into account a single action predicted by the classifier, i.e. this is "rank 1" classification accuracy, rather than the laxer "rank 2" classification accuracy (where the classifier is allowed to make two predictions) reported by Gorin et. al (1999) and Petrovska et. al (2000).

In practical applications of utterance classification, user inputs are rejected if the confidence of the classifier in making a prediction falls below a threshold appropriate to the application. After rejection, the system may, for example, route the call to a human or reprompt the user. We therefore show the accuracy of classifying accepted utterances at different rejection rates, specifically 0% (all utterances accepted), 10%, 20%, 30%, 40%, and 50%. Following Schapire and Singer (2000), the confidence level, for rejection purposes, assigned to a prediction is taken to be the difference between the scores assigned by BoosTexter to the highest ranked action (the predicted action) and the next highest ranked action.

Accuracy Results

Utterance classification accuracy rates, at various rejection rates, for domain A are shown in Table 2 for the three experimental conditions described in section 3.4. The corresponding results for domains B and C are shown in Tables 3 and 4.

The utterances in domain A are on average longer and more complex than in domain B; this may partly explain the higher classification rates for domain B. The generally lower classification accuracy rates for domain C may reflect the larger set of actions for this domain (92 actions, compared with 56 and 54 actions for domains A and B). Another difference between the domains was that the recording quality for domain B was not as high as for domains A and C. Despite these differences between the domains, there is a consistent pattern for the comparison of most interest to this paper, i.e. the relative performance of utterance classification methods requiring or not requiring transcription.

Perhaps the most surprising outcome of these experiments is that the phone-based method with short "phrasal" contexts (up to four phones) has classification accuracy that is so close to that provided by the longer phrasal contexts of trigram word recognition and word-string classification. Of course, the re-estimation of phone n-grams employed in the phone-based method means that two-word units are implicitly modeled since the phone 5-grams modeled in recognition, and 4-grams in classification, can straddle word boundaries.

The experiments suggest that if transcriptions are available (i.e. the effort to produce them has already been expended), then they can be used to slightly improve performance over the phone-based method (**PhonesMM**) not requiring transcriptions. For domains A and C, this would give an absolute performance difference of about 2%, while for domain B the difference is around 1%.

N_{max}	Recog. accuracy	Classif. accuracy
0	54.2	70.0
1	56.6	70.6
2	59.1	71.2
3	59.5	71.5
4	60.0	73.2
5	62.3	74.6

Table 5: Phone recognition accuracy and phone string classification accuracy (PhoneMM with no rejection) for increasing values of N_{max} for domain A.

N_{max}	Recog. accuracy	Classif. accuracy
0	27.9	69.2
1	38.3	70.7
2	48.6	74.7
3	53.3	77.6
4	55.1	79.2
5	55.7	80.8

Table 6: Phone recognition accuracy and phone string classification accuracy (PhoneMM with no rejection) for increasing values of N_{max} for domain B.

Whether it is optimal to train the word-based classifier on the transcriptions (**WordsHH**) or the output of the recognizer (**WordsHM**) seems to depend on the particular data set.

When the operational setting of utterance classification demands very high confidence, and a high degree of rejection is acceptable (e.g. if sufficient human backup operators are available), then the small advantage of the word-based methods is reduced further to less than 1%. This can be seen from the high rejection rate rows of the accuracy tables.

Effectiveness of Unsupervised Training

Tables 5, 6, and 7, show the effect of increasing N_{max} (the final iteration number in the unsupervised phone recognition model) for domains A, B and C, respectively. The row with $N_{max} = 0$ corresponds to the initial unweighted phone loop recognition. The classification accuracies shown in this table are all at 0% rejection. Phone recognition accuracy is the standard ASR error rate accuracy in terms of the percentage of phone insertions, deletions, and substitutions, determined by aligning the ASR output against reference phone transcriptions produced by the pronounciation component of our speech synthesizer. (Since these reference phone transcriptions are not perfect, the actual phone recognition accuracy is probably slightly higher.) Clearly, for all three domains, unsupervised recognition model training improves both recognition and classification accuracy compared with a simple phone loop. Unsupervised training of the recognition model is particularly important for domain B where the quality of recordings is not as high as for domains A and C, so the system needs to depend more on the re-estimated n-gram models to achieve the final classification accuracy.

N_{max}	Recog. accuracy	Classif. accuracy
0	55.4	61.1
1	59.8	61.8
2	65.3	64.3
3	68.1	66.3
4	69.1	67.4
5	69.3	68.2

Table 7: Phone recognition accuracy and phone string classification accuracy (PhoneMM with no rejection) for increasing values of N_{max} for domain C.

5 Concluding Remarks

In this paper we have presented an utterance classification method that does not require manual transcription of training data. The method combines unsupervised re-estimation of phone n-ngram recognition models together with a phone-string classifier. The utterance classification accuracy of the method is surprisingly close to a more traditional method involving manual transcription of training utterances into word strings and recognition with word trigrams. The measured absolute difference in classification accuracy (with no rejection) between our method and the word-based method was only 1% for one test domain and 2% for two other test domains. The performance difference is even smaller (less than 1%) if high rejection thresholds are acceptable. This performance level was achieved despite the large reduction in effort required to develop new applications with the presented utterance classification method.

References

H. Alshawi and S. Douglas. 2001. Variant transduction: A method for rapid development of interactive spoken interfaces. In *Proceedings of the SIGDial Workshop on Discourse and Dialogue*, Aalborg, Denmark, September.

R. Carpenter and J. Chu-Carroll. 1998. Natural language call routing: a robust, self-organizing approach. In *Proceedings of the International Conference on Speech and Language Processing*, Sydney, Australia.

J. Choi, D. Hindle, J. Hirschberg, F. Pereira, A. Singhal, and S. Whittaker. 1999. Spoken content-based audio

navigation (scan). In *Proceedings of ICPhS-99 (International Congress of Phonetics Sciences*, San Francisco, California, August.

J. T. Foote, S. J. Young, G. J. F Jones, and K. Sparck Jones. 1997. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. *Computer Speech and Language*, 11(2):207–224.

Y. Freund and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

A. L. Gorin, G. Riccardi, and J. H. Wright. 1997. How may I help you? *Speech Communication*, 23(1-2):113–127.

A. L. Gorin, D. Petrovska-Delacretaz, G. Riccardi, and J. H. Wright. 1999. Learning Spoken Language without Transcription. In *Proceedings of the ASRU Workshop*, Keystone, Colorado, December.

K. Sparck Jones, G. J. F. Jones, J. T. Foote, and S. J. Young. 1996. Experiments in spoken document retrieval. *Information Processing and Management*, 32(4):399–417.

M. Levit, A. L. Gorin, and J. H. Wright. 2001. Multipass Algorithm for Acquisition of Salient Acoustic Morphemes. In *Proceedings of Eurospeech 2001*, Aalborg, Denmark, September.

A. Ljolje, D. M. Hindle, M. D. Riley, and R. W. Sproat. 2000. The AT&T LVCSR-2000 System. In *Speech Transcription Workshop*, Univ. of Maryland, May.

K. Ng and V. Zue. 1998. Phonetic recognition for spoken document retrieval. In *Proceedings of ICASSP 98*, Seattle, Washington, May.

D. Petrovska-Delacretaz, A. L. Gorin, J. H. Wright, and G. Riccardi. 2000. Detecting Acoustic Morphemes in Lattices for Spoken Language Understanding. In *Proceedings of the Interanational Conference on Spoken Language Processing*, Beijing, China, October.

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

G. Riccardi, R. Pieraccini, and E. Bocchieri. 1996. Stochastic automata for language modeling. *Computer Speech and Language*, 10:265–293.

R. E. Schapire and Y. Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

R. E. Schapire. 1999. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

V. N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, New York.

J. H. Wright, A. L. Gorin, and G. Riccardi. 1997. Automatic acquisition of salient grammar fragments for call-type classification. In *Proceedings of European Conference on Speech Communication and Technology*, pages 1419–1422, Rhodes, Greece, September.