TOWARDS A DOMAIN-INDEPENDENT ASR-CONFIDENCE CLASSIFIER

Om D. Deshmukh, Ashish Verma

IBM Research India, New Delhi
odeshmuk{vashish}@in.ibm.com

Etienne Marcheret

IBM Watson Research Center, USA
etiennem@us.ibm.com

ABSTRACT

This work addresses the problem of developing a domain-independent binary classifier for a test domain given labeled data from several training domains where the test domain is not necessarily present in training data. The classifier accepts or rejects the ASR hypothesis based on the confidence generated by the ASR system. In the proposed approach, training data is grouped into across-domain clusters and separate cluster-specific classifiers are trained. One of the main findings is that the cluster purity and the normalized mutual information of the clusters are not very high which suggests that the domains might not necessarily be natural clusters. The performance of these cluster-specific classifiers is better than that of: (a) a single classifier trained on data from all the domains, and (b) a set of classifiers trained separately for each of the training domains. At an operating point corresponding to low False Accept, the Correct Accept of the proposed technique is on an average 2.3% higher than that obtained by the single-classifier or the individual train-domain classifiers.

Index Terms— confidence measures, K-means clustering, cluster-purity, IVR systems

1. INTRODUCTION

In various Interactive Voice Response (IVR) systems, an Automatic Speech Recognition (ASR) system is used to automatically recognize and interpret users' speech inputs and decide the future course of interaction.

In such cases the reliability of the ASR hypothesis needs to be quantified. A reliability measure is generated by computing a set of ASR-confidence features and combining them. Some of the common ASR-confidence features are normalized acoustic likelihood scores, LM or LM-backoff scores, N-best based measures such as difference in the top N-1 adjacently ranked recognition scores, word posterior probabilities duration-based features such as HMM state duration, phoneme duration and/or word duration. Computation of posterior probabilities is a non-trivial problem and several approximations have been proposed (c.f. [1]). A statistical hypothesis testing framework referred to as the utterance verification framework is also studied extensively to address the ASR-confidence-based accept/reject decisions (c.f. [2]). Please refer to [3] for a comprehensive review of various features and approaches to quantify the reliability of the ASR hypothesis.

The values of these features for acceptable hypotheses vary across different domains. For example, the number of parallel paths in an ASR system used for accepting credit card details would be much smaller than those in an ASR system used for accepting stock market details. The difference in the best and the second-best hypothesis is likely to be much lower for a system with many parallel paths than a system with fewer parallel paths. This suggests that the classifier to make an accept/reject decision based on these features should be retrained across domains.

In the most ideal situation, a different classifier is trained for every test domain. This assumes availability of substantial amount of labeled data for every test domain. Most deployed systems, however, find it inconvenient to retrain a classifier for different test domains for various practical reasons. In many situations, there is little or no labeled data for several test domains. The requirement of labeled data also increases the preparation period of a deployment cycle. A common practice is to use the same classifier irrespective of the domain of the test utterance. To reduce the sensitivity to the train domain, the classifier is typically trained on data from several different domains.

An obvious question arises: Is there a better, more systematic training approach that can utilize the labeled data from different training domains to improve the performance on unseen domains?

Machine Learning literature is rich with various frameworks that try to address this issue, also referred to as the Domain Adaptation problem [4]. The goal in Transfer Learning framework [5] is to utilize information from a multitude of source domains to improve the performance on a target domain. In [6], the transfer learning framework is applied in a Reinforcement Learning setup to learn rules from existing domains that can easily be applied to target domain(s). In [7], small amount of labeled data from the target domain is used to identify training instances that match the distribution of the target domain. A new classifier is built with higher weights on these 'similar' instances. Sample selection bias approach is proposed in [8] where the requirement of labels on the data from the target domain is relaxed. The objective in multi-task learning [9] is to mine for commonalities across multiple domains and learn all the domains simultaneously.

While all of the above methods expect a certain amount of data from the target domain (either labeled or unlabeled), our focus in the current work is to develop a technique that performs efficiently even when the amount of data from the target domain is not enough to reliably estimate any statistic of the target domain. In our companion paper [10], we apply the Transfer Learning framework to the current problem.

In the current work, we propose an across-domain data-driven clustering approach where data from different Training domains is clustered into a pre-defined number of clusters. The assumption is that there are a few features such that their class-specific distribution is independent of the domain to which their corresponding utterances belong. Indeed, we show that the *purity* and normalized mutual information of these across-domain clusters with respect to the training domains is low. A separate binary classifier is trained for each cluster. For a given test utterance, the closest cluster is identified and the corresponding classifier is used to make the decision.

As part of comparative analysis, the proposed approach is compared with: (a) a single-classifier approach where all the training

data is treated as a single cluster and a binary classifier is trained, and (b) a domain-specific classifier approach where a separate classifier is trained for each domain and the classifier corresponding to the closest one is chosen for evaluation. The implicit assumption here is that features from any particular domain occupy a subspace which is largely non-overlapping with the subspaces of other domains. We show that the data-driven clustering approach outperforms the single-classifier and domain-specific classifier technique for all the test domains evaluated.

Note that if features from different domains indeed occupy very different subspaces then data-driven clustering should also be able to assign separate clusters to each of the different domains.

2. ACROSS-DOMAIN CLUSTERING

In the proposed approach, the training data from all the training domains is clustered into K clusters using the K-means algorithm [11]. The initial *K* centroids are chosen in such a way that each centroid is reasonably far from the other centroids. Careful choice of the initial centroids is critical as the clustering output is quite sensitive to initialization. In the current approach, the centroid of the *i*-th cluster is chosen as follows: distance of all the instances from the *(i-1)* cluster centroids is calculated. N instances that are farthest from all these *(i-1)* centroids are short-listed and one of them is randomly picked as the centroid for the *i*-th cluster. The centroid for the first cluster (*i*=1) is randomly chosen. Short-listing N farthest instances, instead of picking the one farthest instance, reduces the chances of picking outliers as potential centroids. Euclidean distance is used as the distance measure throughout this work. Note that a direct consequence of using Euclidean distance is that the cluster boundaries will be hyper-spheres. Although, the K-means clustering process is known to always converge, we allow for termination either when number of iterations crosses a certain number or when the cumulative difference in the centroids across consecutive iterations falls below a certain threshold. The clustering algorithm was run a few times with different initial centroids. The resultant centroids were similar across different runs indicating that the objective function for the current scenario might approximately resemble a convex function and thus less sensitive to initialization.

One of the issues that needs to be addressed is deciding K: the number of clusters. While a reasonable guess is to let K to be equal to the number of different domains in the training data, we also tried a few values in the vicinity. A few post-clustering refinements were also explored: (a) clusters with number of instances less than a certain percentage of the total number of training instances are discarded, (b) two clusters are merged if the distance between their centroids is less than a certain threshold and if the label of the majority class is the same in both the clusters. The second condition for merging ensures that clusters which capture class-specific feature distribution are not merged.

A separate binary classifier is trained for each cluster. During evaluation, the input test instance is assigned to the cluster whose centroid is the closest to the test instance. The classifier trained for the chosen cluster is used to classify the instance. This method of assigning the test instance to a single cluster and discarding all the other clusters can be thought of as hard-decisioning. A soft-decisioning approach is also explored where the test instance is classified using classifiers trained for each of the clusters. The contribution of each classifier's decision in the final decision is inversely proportional to the distance of the test instance from the corresponding cluster centroid. If the decisions from the K clusters are $[c_1, c_2, \ldots, c_K]$ (note that $c_i = \pm 1$) and the corresponding distances are $[d_1, d_2, \ldots, d_K]$ then these distances are converted to weights as $w_i = exp(-d_i/D)$ where D is the maximum of $[d_1, d_2, \ldots, d_K]$. The above transformation ensures that closer clusters get a higher weight and thus a higher say in the final decision. The final decision C_f is then computed as:

$$C_f = sign[\frac{\sum w_i * c_i}{\sum w_i}]$$

Soft-decisioning reverses the hard-decision in situations where more than one clusters are at similar distances from the test utterances and the decision made by the closest cluster is in minority as compared to the decision made by the other close clusters.

Table 1. *Purity* and NMI for data-driven clusters for various K values. 'as-is' indicates the clusters obtained by the K-means algorithm. 'post-clust' indicates the clusters obtained after post-clustering refinements described in Section 2

	purity		NMI	
K	as-is	post-clust	as-is	post-clust
5	0.277	0.277	0.203	0.203
10	0.299	0.283	0.197	0.196
12	0.314	0.289	0.214	0.211
14	0.324	0.291	0.227	0.211
15	0.322	0.293	0.211	0.213
21	0.338	0.294	0.213	0.215

2.1. Cluster analysis

In the current work, *purity* and *normalized mutual information* measures are used to understand the composition of the clusters in terms of individual domains. *Purity* is computed as follows: Every instance in a cluster is assigned a label corresponding to the domain which is most frequently present in the cluster. *Purity*, which is defined as the accuracy of this assignment, is the ratio of numbers of instances with correct domain assignments and the total number of instances. Situations where every cluster contains instances from a single (and possibly different) domain would lead to a *purity* of 1. On the other hand, situations where every domain is spread out across every cluster would lead to a very low *purity*. Typically, high *purity* can be achieved by increasing the number of clusters. For example, the extreme case of forming a separate cluster for every instance would lead to a *purity* of 1. A different measure that works similar to *purity* but penalizes high number of clusters is Normalized Mutual Information (NMI). NMI is defined as:

$$NMI(D, C) = \frac{I(D; C)}{[H(D) + H(C)]/2}$$

where D is the set of domains, C is the set of clusters, $I(D; C)$ is the mutual information between the set of clusters and the set of domains, $H(D)$ is the domain entropy and $H(C)$ is the cluster entropy. The cluster entropy term in the denominator increases as the number of clusters increases thus penalizing high number of clusters.

The composition of the across-domain clusters at various K values, as indicated by *purity* and NMI, is presented in Table 1. Low values of *purity* and NMI confirm that the domains do not naturally form clusters by themselves and that there is a substantial spread of every domain across clusters. One other interesting observation to be made from the table is that the *purity* and the NMI values for the post-clustering refinement cases are very similar to each other

irrespective of the initial K value. This suggests that irrespective of the starting value of K, the post-clustering refinement steps lead to a near-steady-state clustering output. We have observed that more number of clusters are merged/discarded at higher values of K.

3. EXPERIMENTS

3.1. Database

The data used in this work consists of 13 domains spanning a wide range of perplexity and fan-outs such as 'credit card numbers' domain which has a smaller fanout to 'last names' domain which has a relatively large fanout. Total number of instances is 64,052 with substantial representation from each of the domains: the number of instances for individual domains vary from 2035 to 8477. The binary labels for the training and the test data were assigned by human experts where a positive label indicates the ASR output should be accepted and a negative label indicates the ASR output should be rejected. Total number of instances with positive labels (i.e., accept) is 51,621 and number of instances with negative labels (i.e., reject) is 12,431. The experiments were carried out in a round-robin leave-one-out fashion where every domain is used as the test domain once with all the other domains forming the training domains.

3.2. Experimental Setup

The confidence features used for the clustering and for classifier training include: (1) the difference in the scores of the best and the second-best, (2) the difference in the scores of the best and the third-best, (3) the difference in scores of the best and a phone-loop garbage model which captures hesitations and disfluencies, (4) average acoustic score of the decoded phones, (5) the percentage of frames recognized as silence, (6) number of phones in the decoded utterance. Each of these features is individually normalized so that the dynamic range is restricted to $[0-1]$ in the training data. The normalized feature n_i (where i is the dimension) is given by:

$$n_i = \frac{x_i - min_i}{max_i - min_i}$$

where x_i is the corresponding original feature, max_i is the maximum value of the feature in the training data and min_i is the minimum value.

Number of clusters, K, was varied from 5-21. N, the number of farthest instances short-listed is set to 100. Maximum number of iterations of the K-means algorithm was set to 500 and the terminating threshold on cumulative difference in centroids across consecutive iterations was set to 0.001. The classifier used is the WEKA implementation [12] of the logistic-regression classifier [13].

3.3. Evaluation Metric

The performance of the overall classification task is evaluated using the False Accept (FA) and Correct Accept (CA) metrics. FA is defined as the ratio of number of negative-labeled instances which the classifier predicts as positive (i.e., accept) to the total number of negative-labeled instances. CA is defined as the ratio of number of positive-labeled instances which the classifier predicts as positive to the total number of positive-labeled instances. FA and CA are typically multiplied by 100 to present them as percentages. The FA rate is controlled by varying the threshold on the ASR-confidence above which the ASR outputs are accepted. This also leads to the Receiver Operating Curve (ROC) which is used to compare the performance of different techniques at various FA/CA values. The metric used by the deployment team is the average of CA values corresponding to FA values of 2-, 3- and 4%.

Table 2. Average CA values for FA of 2,3 and 4%. The four methods compared are: (A) Matched-condition; (B) single-classifier; (C) individual-domain; (D) data-driven clustering for K = 12;

domain	(A)	(B)	(C)	(D)
1	54.0	48.4	37.7	49.9
2	64.4	58.4	53.8	63.2
3	67.0	61.4	58.8	66.5
4	97.4	93.3	92.7	92.0
5	98.4	98.0	94.6	98.2
6	27.1	20.5	17.1	19.4
7	50.5	43.6	36.3	45.0
8	33.2	26.6	46.3	44.8
9	23.1	20.4	17.3	21.1
10	97.3	95.5	85.5	94.9
11	80.7	81.7	66.8	80.4
12	51.7	48.0	42.5	48.0
13	96.7	95.3	92.6	96.4
average	64.7	60.8	57.1	63.1

3.4. Baseline Techniques

The performance of the data-driven clustering technique is compared with the following baseline setups:

Single-classifier: In this setup, data from all the training domains is grouped together and a single classifier is trained. This is the setup that current deployments employ.

Train-domain classifiers: In this setup, a separate classifier is trained for each train domain. Given a test utterance, the closest train domain is identified by computing the distance between the utterance and the centroids of each of the train domains. The classifier trained on this closest domain is used to classify the test utterance.

The performance of the proposed technique is also compared with the ideal *match-condition* case where a separate classifier is trained for each test domain using data from the same domain. Results for this condition are presented on 10-fold cross-validation.

4. RESULTS

Table 2 compares the average CA values at FA of 2, 3, and 4% for each of the domains using the different techniques described above. As expected, the performance of the matched-condition is superior to all the other approaches in all but one domains. The performance of the individual-domain case is much inferior to that of the matched-condition or the single-classifier cases.

The performance of the proposed across-domain clustering was evaluated for various number of clusters, with and without the post-clustering refinements, and with hard- and soft-decisioning. For most of these scenarios, an improvement in performance is observed as compared to that of the single-classifier case. In many cases, the combination of post-clustering refinements and soft-clustering improves the performance as compared to when neither of them is used. These improvements are only about 0.5% though. The best average performance across all domains, however, is for the 12-cluster, no post-clustering refinements and hard-decisioning case. In rest of the paper, results only from the 12-cluster case are presented. As can be inferred from Table 2, the performance of the 12-cluster case is better

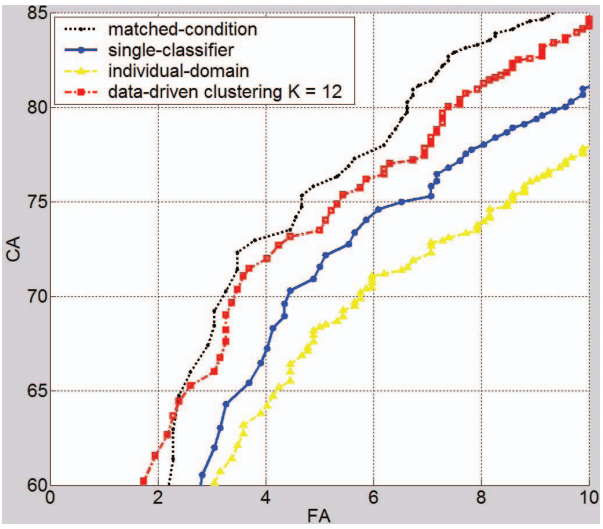

Fig. 1. Comparison of ROC for domain 3 from Table 2 for various methods: (a) matched-condition (black dotted-line); (b) single-classifier (solid blue line with circles); (c) individual-domain classifier (yellow line with triangles); (d) proposed across-domain clustering with K = 12 (red line with squares);

Table 3. Comparison of Average CA values for FA of 2,3 and 4% when number of training domains (N_T) is varied; 'std-dev' implies standard deviation. (Test domain is domain-2 from table 2)

N_T	single-classifier		proposed method	
	avg-CA	std-dev	avg-CA	std-dev
1	26.2	25.9	26.2	25.9
3	44.3	19.7	45.7	17.7
5	46.9	14.5	56.6	9.4
7	48.5	9.1	56.4	6.8
9	56.7	3.2	61.9	4.6
10	57.4	2.5	62.9	1.3
12	58.4	-	63.2	-

than that of the single-classifier or the individual-domain technique and comes very close to the performance of matched-condition case for many domains. The CA value averaged across all the domains for the match-condition case is 64.7% while it drops to 60.8% for the single-classifier case. The proposed 12-cluster case bridges this gap by about 58.9% [$= (63.1 - 60.8)/(64.7 - 60.8)$].

The effect of the amount of training data on the performance of the different techniques is analyzed next. To that effect, the number of training domains (N_T) is gradually increased from 1 to 12. For example, for $N_T = 3$, any 3 domains are chosen from the available 12 domains to train classifiers. There are $\binom{12}{3}$ such possibilities. 20 random combinations are chosen and the performance averaged across the combinations. Table 3 presents the average CA values when N_T is varied from 1 to 12. The test domain is domain-2 from Table 2. The standard deviation in the performance across the 20 combinations is also presented. Note that, as expected, increase in N_T increases the performance of the single-classifier as well as the proposed technique. The deviation in performance across the multiple combinations also drops as N_T is increased. Also, note that the performance of the proposed technique is always superior to that of the single-classifier case implying that the proposed across-domain clustering better utilizes the available data.

Figure 1 compares the ROC for the single-classifier, match-condition, individual-domain and the 12-cluster case for domain 3 of Table 2. Note that the ROC for the 12-cluster case is consistently above that of the single-classifier and touches that of the match-condition at several points. Similar trend is observed for the other domains presented in Table 2.

5. DISCUSSION AND FUTURE WORK

In summary, we show that the performance of the task of ASR-confidence based automatic acceptance/rejection of the ASR hypothesis on unknown domains can be improved by forming across-domain clusters of the training data and training a separate classifier for each cluster. The decision on the test utterance is formed by using the classifier corresponding to the closest cluster. The performance of such a technique is shown to approach that of the matched-condition case for several domains. Our on-going efforts in exploring Mixture Regression techniques and the Transfer Learning framework are presented in a companion paper [10].

6. REFERENCES

[1] F. Wessel et. al., "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. on Speech and Audio Proc.*, vol. 9(3), pp. 288–298, Mar. 2001.

[2] H. Jiang and C. H. Lee, "A new approach to utteranceverification based on neighborhood information in model space," *IEEE Trans. on Speech and Audio Proc.*, vol. 11(5), pp. 425–434, May 2003.

[3] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, pp. 455–470, 2005.

[4] J. Blitzer and H. Daume, "Overview of domain adaptation," 2010, Tutorial on Domain Adaptation at ICML-2010.

[5] S. Thrun and T. Mitchell, "Learning one more thing," in *Proc. of the Intl. Joint Conf. on Artificial Intelligence*, San Mateo, California, USA, 1995.

[6] M. E. Taylor and P. Stone, "Cross-domain transfer for reinforcement learning," in *Proc. of the Intl. Conf. on Machine Learning*, Corvalis, Oregon, 2007, pp. 879–886.

[7] W. Dai et. al., "Boosting for transfer learning," in *Proc. of the Intl. Conf. on Machine Learning*, Corvalis, Oregon, 2007, pp. 193–200.

[8] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. of the Intl. Conf. on Machine Learning*, Banff, Alberta, Canada, 2004, pp. 114–121.

[9] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28(1), pp. 41–75, 1997.

[10] E. Marcheret et. al., "A framework for unsupervised transfer learning and application to dialog decision classification," in *to appear in Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 2012.

[11] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd edition, New Yorkk; Wiley, 2001.

[12] S. Garner, "Weka: the watiko environment for knowledge analysis," in *Proc. New Zeland Computing Science Research Students Conference*, 1995, pp. 57–64.

[13] P. Komarek, *Logistic Regression for Data Mining and High-Dimensional Classification*, Ph.D. thesis, Pittsburgh, PA, May 2004.