SPEECH UTTERANCE CLASSIFICATION

Ciprian Chelba, Milind Mahajan, Alex Acero

Microsoft Research
One Microsoft Way
Redmond, WA 98052

ABSTRACT

The paper presents a series of experiments on speech utterance classification performed on the ATIS corpus. We compare the performance of n-gram classifiers with that of Naive Bayes and maximum entropy classifiers. The n-gram classifiers have the advantage that one can use a single pass system (concurrent speech recognition and classification) whereas for Naive Bayes or maximum entropy classification we use a two-stage system: speech recognition followed by classification. Substantial relative improvements (up to 55%) in classification accuracy can be obtained using discriminative training methods that belong to the class of conditional maximum likelihood techniques.

1. INTRODUCTION

Speech utterance classification as well as text classification are an interesting subproblem in a growing trend of providing natural language user interfaces to automated systems. A straightforward application among many others is call-routing, a problem tackled by various research groups: [1], [2].

Text classification and/or categorization literature is far from scarce since it is a straightforward application of various classification techniques. A good starting point for further reading is [3].

The current work aims at comparing various design decisions when devising an utterance classifier: should one use a one-pass or a two-pass system? Are discriminative training techniques effective? Are richer feature sets useful? This is a broad scope and it is very hard to conduct an exhaustive, even extensive set of experiments. Our goal is to provide experimental results that would guide various choices or further experiments when designing such a system.

We have compared the performance of n-gram classifiers with that of Naive Bayes (NB) and maximum entropy (ME) [4] classifiers. n-gram classifiers lend themselves naturally to being used in a one-pass system whereas NB or ME classifiers operate on a feature vector built from a complete word string and thus incorporating them in a one-pass system is not straighforward.

Another direction explored was training the classifiers using discriminative techniques that belong to the class of conditional maximum likelihood (CML) methods. We have tried using the rational function growth transform (RFGT) [5] for training NB classifiers under the CML criterion. CML NB is found to be equivalent to ME models for the set of features relevant to this problem.

Finally, since the n-gram classifiers intrinsically use n-gram feature sets, we have also evaluated the impact of using 2-gram features along with 1-gram features in the NB and ME classifiers.

The paper is organized as follows: Section 2 describes the problem setup and reviews three common text classifiers: n-gram, Naive Bayes and ME. Section 3 presents various experiments along the directions outlined previously. We conclude with a short analysis of the results.

2. UTTERANCE CLASSIFICATION

To fix notation, we denote a speech utterance with A, the word string that gave rise to it with $W = w_1 \dots w_n$ and the class of the utterance with $C(A)$. The word vocabulary is denoted with $\mathcal{V}$ and the class vocabulary with $\mathcal{C}$. The corpus, split in training and test data, $\mathcal{T}$ and $\mathcal{E}$, respectively, consists of tuples (or samples) s containing: utterance A, transcription W and utterance class $C(A)$. The performance of a given classifier is measured by the class error rate (CER).

2.1. n-gram Classifier

Assume one builds an n-gram model $P(w_i|w_{i-1}, \dots, w_{i-n+1}, C)$ for each class $C \in \mathcal{C}$ by pooling all the training transcriptions that have been labeled with class C.

In a one-pass scenario the decoder search for the most likely path will find

$$(\widehat{C(A)}, \widehat{W}) = \arg\max_{(C,W)} \log P(A|W) + \log P(W|C) + \log P(C)$$

In a two-pass scenario one builds a *pooled* n-gram language model $P(w_i|w_{i-1}, w_{i-n+1})$ from all the training transcriptions in addition to the class specific language models $P(\cdot|C)$. Each test utterance is then assigned a class by doing text classification on the 1-best recognition output using the *pooled* language model:

$$\widehat{C(A)} = \arg\max_C \log P(\widehat{W}|C) + \log P(C)$$

$$\widehat{W} = \arg\max_W \log P(A|W) + \log P(W)$$

We have found smoothing to be a very important issue for all classifiers that we experimented with. For estimating the n-gram models we use the recursive deleted interpolation scheme [6] between relative frequency estimates at different orders.

2.2. Naive Bayes Classifier

For any given event (W, C) in the training or test data, one constructs a binary valued feature vector listing the values each feature takes at this particular point:

$$\underline{f}(W) = (f_1(W), \ldots, f_F(W))$$

Let $\mathcal{F} = \{f_k, k = 1 \ldots F\}$ be the set of features chosen for building a particular model $P(W, C)$. They are binary valued indicator functions $f(W) : \mathcal{V}^* \rightarrow \{0, 1\}$. For convenience we denote $\overline{f_i(W)} = 1 - f_i(W)$. We have explored using features of the form $f_w(W) = 1 \iff w \in W$ (1-gram features) or $f_{w_i, w_{i-1}, \ldots, w_{i-N+1}}(W) = 1 \iff (w_i, w_{i-1}, \ldots, w_{i-N+1}) \in W$ (n-gram features).

Assuming a NB model for the feature vector (see [7]) and the predicted variable C (the utterance class), their joint probability is calculated as

$$P(\underline{f}(W), C) = \theta_C \prod_{k=1}^{F} \theta_{kC}{}^{f_k(W)} \overline{\theta}_{kC}{}^{\overline{f_k(W)}} \qquad (1)$$

where θ_C and θ_{kC} are properly normalized. The class for a given utterance is assigned in two passes.

2.2.1. Maximum Likelihood Parameter Estimation

The parameters θ_C are estimated using maximum likelihood from the training data (relative frequencies). The parameters θ_{kC} are estimated using MAP smoothing:

$$\theta_{kC} = \frac{C(C, f_k) + \epsilon \cdot 1/2}{C(C) + \epsilon}$$

2.2.2. Conditional Maximum Likelihood Parameter Estimation

Another option for training the parameters of the model that is expected to be better correlated with the CER is to maximize the *conditional* likelihood of the training data

$$\sum_{W,C} \tilde{P}(W, C) \log P(C|W; \underline{\theta})$$

where $\tilde{P}$ denotes the empirical distribution over the training set.

We have used the rational function growth transform (RFGT) algorithm described in [5] for estimating the parameters of the model under the conditional maximum likelihood (CML) criterion. Due to the limited amount of space we do not go into the details of the estimation procedure.

It can be easily shown that Eq. (1) can be rewritten as a log-linear model of the type that arises in ME probability modeling. Moreover, under the CML estimation criterion the same objective function is maximized for both NB and ME models.

2.3. Maximum Entropy Classifier

As described in [4], a ME classifier selects a conditional distribution $P(C|W)$ with maximum conditional entropy $H(C|W)$ from a family of distributions which satisfy the set of constraints:

$$\sum_{W,C} \tilde{P}(W, C) \cdot f_k(W, C) =$$
$$\sum_{W,C} \tilde{P}(W) \cdot P(C|W) \cdot f_k(W, C), \forall k = \overline{1, F}$$

where $\tilde{P}$ denotes the empirical distribution over the training set.

We have found smoothing to be extremely important for improving the classification accuracy. As shown in [8] ME models can be smoothed using a Gaussian prior on the feature weights and $\underline{\lambda}^*$ can be selected using the *maximum a posteriori* (MAP) criterion. A modified version of improved iterative scaling (IIS) (as presented in [4]) can be used to find $\underline{\lambda}^*$ under MAP:

$$\underline{\lambda}^* = \arg\max_{\underline{\lambda}} \sum_{W,C} \tilde{P}(W, C) \cdot \log P(C|W; \underline{\lambda}) - \frac{1}{2 \cdot |\mathcal{T}|} \cdot \sum_{k=0}^{F} \frac{\lambda_k^2}{\sigma_k^2}$$

$$P(C|W; \underline{\lambda}) = Z(W; \underline{\lambda})^{-1} \cdot exp(\sum_{k=0}^{F} \lambda_{kC} f_k(W, C))$$

where σ_k^2 represent the variance parameters of the Gaussian prior and $|\mathcal{T}|$ is the size of training set.

3. EXPERIMENTS

3.1. Experimental Setup

All experiments were carried out on the ATIS corpus [9]. We have pooled the ATIS II and ATIS III data after which we extracted the type A utterances (that can be interpreted independent of context) along with their transcriptions. We have used the ATIS III dev94 class A utterances as a development set for tuning the speech recognition system. The

test set was obtained by pooling the ATIS III 93 and 94 test sets such that enough utterances of class A were available for testing our classifier.

The acoustic model was trained on all the ATIS II and III training data, irrespective of utterance class (A, D or X). We have built a standard tied-state cross-word triphone HMM acoustic model using the HTK [10] training tools.

For language model and classifier training we have used only type A utterances along with a class label assigned by taking the argument of the first `SELECT` statement in the SQL querry associated with the utterance. There were 14 classes derived in this manner, their distribution being highly skewed towards the `FLIGHT` class which covers about 74% of the utterances. The training data consisted of 5,822 class A utterances (74,442 words). The test data consisted of 914 class A utterances (10,673 words). The development data consisted of 410 utterances (5,326 words).

The vocabulary derived from the training data had size 780 and out-of-vocabulary rate on test data 0.24%. The pooled 1,2,3-gram language models built on the above vocabulary had test set perplexity 149, 19, 15, respectively.

All speech recognition experiments were carried out by statically compiling word level recognition networks from the various deleted interpolation language models by representing them as finite state networks. The word insertion penalty and language model weight were chosen to minimize speech recognition word error rate (WER) on the development data and were fixed throughout the experiments.

Table 1 compares the performance of the various n-gram classifiers and the ML NB classifier in the two pass scenario. We have also run a set of control experiments in which each classifier is fed the transcription for the test utterances aimed at gauging the gap in performance caused by speech regnition errors. The run-times are normalized to 1GHz CPU.

It is unfair to compare the 2 and 3 -gram classifiers with the Naive Bayes classifier since they use a larger feature set. For the same feature set (be it 1-gram or 1+2-gram, see Table 4 for the ML NB classifier performance when using 1+2-gram features) the two classifiers perform equally well. All classifiers are fairly robust to degradation in WER in the first pass.

3.2. One-pass vs. Two-pass classification

Table 2 compares the performance of the n-gram classifiers when run in a one-pass vs. a two-pass scenario; the one-pass classifiers outperform their 2-pass counterpart at all orders.

A surprising result is that the 2 and 3 -gram one-pass systems perform better than the corresponding classifier when fed the correct transcription (the improvement is significant at level 0.14 and 0.17 according to a sign test, respectively). We consider this to be a side effect of pruning while searching for the most likely path in the recognition network.

1st pass	2nd pass	CER	WER	runtime
1gram	1gram	12.1%	13.0%	4.2hrs
2gram	1gram	11.1%	6.0%	1.8hrs
3gram	1gram	10.7%	5.1%	1.8hrs
Transcript	1gram	10.6%	0%	—-
1gram	2gram	11.2%	13.0%	4.2hrs
2gram	2gram	9.6%	6.0%	1.8hrs
3gram	2gram	10.1%	5.1%	1.8hrs
Transcript	2gram	9.3%	0%	—-
1gram	3gram	11.2%	13.0%	4.2hrs
2gram	3gram	9.4%	6.0%	1.8hrs
3gram	3gram	9.4%	5.1%	1.8hrs
Transcript	3gram	9.6%	0%	—-
1gram	NB	11.6%	13.0%	4.2hrs
2gram	NB	11.6%	6.0%	1.8hrs
3gram	NB	11.4%	5.1%	1.8hrs
Transcript	NB	11.3%	0%	—-

Table 1. Classification error rate (CER), word error rate (WER) and decoding time (1GHz Pentium) for a two-pass system: 1-best word string from n-gram decoder is fed to n-gram/Naive Bayes classifier trained under ML

3.3. Discriminative Training

As explained in Section 2.2.2, CML is expected to be correlated better with the classification error rate. We have trained the NB classifier under the CML criterion using the RFGT algorithm. Table 3 contrasts these results with the ML trained NB classifier as well as the ME one.

As can be seen, discriminative training has a substantial impact on the CER of the NB classifier. The ME classifier performs substantially better than the CML NB classifier despite the fact that the two are equivalent for the feature set used, as outlined in Section 2.2.2. We attribute this difference in performance to superiority of IIS vs. RFGT convergence properties as well as integration of smoothing.

Another observation is that the classifiers trained using discriminative methods are more sensitive to errors in the 1-st pass compared to the ML NB classifier.

3.4. Feature Set

The n-gram classifiers inherently use n-gram features. To compensate for this mismatch with the NB and ME classifiers, we have run an experiment in which we use a feature vector consisting of 1-gram as well as 2-gram features (see Section 2.2) that have been seen in the training data. The results are shown in Table 4.

The addition of 2-gram features improves the performance of the ML Naive Bayes classifier about as much as it does for the ML n-gram classifier (see Table 1) but it has little impact on the discriminatively trained classifiers. One positive effect of adding the 2-gram features is more grace-

n-gram order	no. passes	CER	WER	runtime
1gram	1	11.8%	12.3%	6.6hrs
1gram	2	12.1%	13.0%	4.2hrs
Transcript	x	10.6%	0%	—-
2gram	1	8.5%	6.3%	3.15hrs
2gram	2	9.6%	6.0%	1.8hrs
Transcript	x	9.3%	0%	—-
3gram	1	9.0%	5.5%	3.36hrs
3gram	2	9.4%	5.1%	1.8hrs
Transcript	x	9.6%	0%	—-

Table 2. Comparison between one and two -pass classification error rate (CER) and word error rate (WER) for ML n-gram classifiers: one pass decoder is driven by the n-gram classifier; two-pass system feeds the 1-best word string from n-gram 1-st pass to n-gram classifier

1st pass	CER			WER
	ML NB	CML NB	ME	
1gram	11.6%	7.4%	7.8%	13.0%
2gram	11.6%	7.3%	5.9%	6.0%
3gram	11.4%	7.1%	5.5%	5.1%
Transcript	11.3%	6.7%	4.9%	0%

Table 3. Comparison between Naive Bayes trained using ML and CML and Maximum Entropy classifiers run in a two-pass system

ful degradation in classification accuracy with WER in the first pass for the discriminative classifiers.

1st pass	CER			WER
	ML NB	CML NB	ME	
1gram	10.0%	7.4%	6.1%	13.0%
2gram	9.5%	7.5%	5.4%	6.0%
3gram	9.7%	7.8%	5.3%	5.1%
Transcript	9.5%	7.3%	4.8%	0%

Table 4. Class error rate (CER) and word error rate (WER) for two-pass system using Naive Bayes trained under ML/CML and Maximum Entropy classifiers whose input consists of both 1-gram and 2-gram features

4. CONCLUSIONS

Discriminative training is highly desirable for improving classification accuracy. Depending on the specific algorithm used for estimating the parameters the relative improvement in accuracy on the ATIS task was 35% — 55%. The classifiers trained discriminatively become slightly less robust to speech recognition errors but this shortcoming is vastly outweighed by the improvement in accuracy.

Incorporating the classification in the speech recognition step (one-pass systems) leads to better results at a moderate cost in run-time. Although not fully conclusive, our experiments indicate that the performance of a classifier trained on correct transcriptions for training utterances and evaluated on the correct transcription for test data can be surpassed by a one-pass system that classifies the speech utterance directly.

Using richer feature sets (2-gram and 3-gram) helps, although the improvement for discriminatively trained classifiers is modest. In that case however they add more robustness to speech recognition errors.

5. REFERENCES

[1] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may I help you?," *Speech Communication*, vol. 23, no. 1/2, pp. 113–127, 1997.

[2] Jennifer Chu-Carroll and Bob Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, vol. 25, no. 3, pp. 361–388, 1999.

[3] Christopher D. Manning and Hinrich Schutze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts, 2001.

[4] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–72, March 1996.

[5] P. S. Gopalakrishnan et al., "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 107–113, January 1991.

[6] Frederick Jelinek and Robert Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, E. Gelsema and L. Kanal, Eds., pp. 381–397. 1980.

[7] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, John Wiley and Sons, Inc., second edition, 2001.

[8] Stanley F. Chen and Ronald Rosenfeld, "A survey of smoothing techniques for maximum entropy models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 37–50, 2000.

[9] D. Pallet et al., "1993 benchmark tests for the ARPA spoken language program," in *Proceedings of the Human Language Technology Workshop*, C.J. Weinstein, Ed. Morgan Kaufmann, Plainsboro, NJ, March 1994.

[10] S. Young, "The HTK hidden Markov model toolkit: design and philosophy," Tech. Rep. TR.153, Department of Engineering, Cambridge University, UK, 1993.