NATURAL LANGUAGE CALL ROUTING: TOWARDS COMBINATION AND BOOSTING OF CLASSIFIERS

Imed Zitouni, Hong-Kwang Jeff Kuo, Chin-Hui Lee *

Bell Labs, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974, U.S.A.
{zitouni, kuo, chl}@research.bell-labs.com

ABSTRACT

In this paper, we describe different techniques to improve natural language call routing: boosting, relevance feedback, discriminative training, and constrained minimization. Their common goal is to reweight the data in order to let the system focus on documents judged hard to classify by a single classifier. These approaches are evaluated with the common vector-based classifier and also with the beta classifier which had given good results in the similar task of E-mail steering. We explore ways of deriving and combining uncorrelated classifiers in order to improve accuracy. Compared to the cosine and beta baseline classifiers, we report an improvement of 49% and 10%, respectively.

1. INTRODUCTION

Topic identification systems attempt to reproduce human categorization judgments. We investigate in this paper the application of natural language call routing, in which the caller may say what he/she wants and is automatically routed to the right department or directed to a human operator when the system is unable to determine the caller's intent with certainty. In probabilistic approaches, call routing is treated as an instance of document routing, where a collection of labeled documents is used for training and the task is to judge the relevance of a set of test documents. Each destination in the call center is treated as a collection of documents (transcriptions of calls routed to that destination), and a new caller request is evaluated in terms of relevance to each destination [1].

In this paper, we first present two baseline classifiers which have been used in the past for call routing and E-mail routing. We first show that each of these two baseline classifiers can be improved individually through a variety of techniques. Using *boosting*, multiple classifiers of the same type can be trained on reweighted data. These classifiers will have errors which are not totally correlated and can therefore be combined to produce a more powerful classifier than any of the individual ones. *Discriminative training* optimizes the classifier to achieve the minimum classification error criterion on the training data. Finally, *automatic relevance feedback* reformulates the user query to move it towards relevant classes during testing.

In addition to combining classifiers of the same type, combining those of different types can also further improve classification accuracy. *Linear interpolation* is a simple method for combining classifiers. *Constrained minimization*, which uses three classifiers, is also introduced; in this scheme, the decision is made by the first two classifiers if they agree, and arbitrated by the third when they disagree.

2. COSINE CLASSIFIER

The cosine classifier is a popular vector-based classifier used in information retrieval that has been adopted for natural language call routing [1, 2]. The training process involves constructing a routing matrix R $(m \times n)$. A list of ignore words are eliminated and a list of stop words are replaced with place holders The rows of R represent the m terms (e.g., words) and the columns the n destinations. The routing matrix R is the transpose of the term-document matrix, where r_{vw} is the frequency with which term w occurs in calls to destination v. Each term is weighted according to term frequency inverse document frequency (TFIDF) and are also normalized to unit length [3]. New user requests are represented as feature vectors and are routed based on the cosine similarity score.

Let $\vec{x}$ be the m-dimensional observation vector representing the weighted terms which have been extracted from the user's utterance. One possible routing decision is to route to the destination with the highest cosine similarity score:

$$\text{destination } \hat{j} = \arg\max_j \cos\phi_j = \arg\max_j \frac{\vec{r_j}\cdot\vec{x}}{\|\vec{r_j}\|\|\vec{x}\|}. \quad (1)$$

3. BETA CLASSIFIER

The beta classifier is a probabilistic method which has previously been shown to give the best results in a study on E-mail routing [4]. Each topic is represented by a word vocabulary and for each word we compute its probability in the topic and its weight [4]. This weight is assigned according to a function inversely proportional to the number of topic-vocabularies in which this word is present. A query $W_1^N = w_1, w_2, \ldots, w_N$ is routed to the destination j with the highest similarity measure:

$$destination\ \hat{j} = arg\max_j T_j(w_1, w_2, \ldots w_k, \ldots, w_N)$$
$$= arg\max_j \left(\beta^{\delta_1} \frac{\sum_{k=1}^{N} P(w_k/T_j)(\eta(w_k))^{\delta_2}}{N}\right), \quad (2)$$

where N denotes the number of words in the query, $P(w_k/T_j)$ the probability of w_k in topic T_j, and $\eta(w_k)$ the weight assigned to w_k. Parameters δ_1 and δ_2 are estimated on a development corpus to boost the accuracy. In our experiments, we obtain a value of 0.3 for δ_1, a value of 2 for δ_2 and we take into account words that occur at least three time in the corpus. The term β_j is the weight assigned to topic T_j:

$$\beta_j = \frac{\sum_{t=1}^{N_j} \eta(w_t)}{\sum_{k=1}^{J}\sum_{t=1}^{N_k} \eta(w_t)}, \quad (3)$$

*Now a visiting professor at School of Computing, National University of Singapore, email contact at chl@comp.nus.edu.sg.

where N_k represents the number of words in the k^{th} topic-vocabulary.

4. BOOSTING TECHNIQUE

The basic idea behind this approach is to improve a weak learning boolean algorithm [5]. The procedure of the adopted algorithm we use is as follow:

- Input: $(x_1, y_1), \ldots, (x_m, y_m)$, where m denotes the number of documents in the training corpus, $x_i \in X$ is a document and $y_i \in Y$ is the corresponding topic;
- let distribution D be initialized to $D_1(i) = \frac{1}{m}$ and classifier set $\Re$ be initialized to null;
- for $k = 1, \ldots, K$:
  - ① train classifier c_k using the distribution D_k, building $S_{c_k}(j, q)$ which is the similarity distance between a topic j and a query q;
  - ② compute the classifier error rate ε_k on the training corpus according to the distribution D_k:
  $$\varepsilon_k = Pr_{i \sim D_k}[c_k(x_i) \neq y_i] = \sum_{i: c_k(x_i) \neq y_i} D_k(i)$$
  - ③ compute $\alpha_k = \frac{1}{2} \, ln\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right)$;
  - ④ add the classifier c_k to the set $\Re$ only if $\Re$ is empty or if the classifier C_T, which is the combination between classifiers in $\Re$ and c_k, improves the accuracy on the training corpus:
  $$C_T(q) = arg\max_j \left(\alpha_k S_{c_k}(j, q) + \sum_{c_t \in \Re} \alpha_t S_{c_t}(j, q)\right)$$
  - ⑤ update the distribution D_k:
  $$D_{k+1}(i) = \frac{D_k}{Z_k} \times \begin{cases} e^{-\alpha_k} & \text{if } c_k(x_i) = y_i \\ e^{\alpha_k} & \text{if } c_k(x_i) \neq y_i \end{cases}$$
  where Z_k denotes a normalization factor chosen so that D_{k+1} will be a distribution;
- Output: the final classifier C such that:
$$C(q) = arg\max_j \left(\sum_{t \in \Re} \alpha_t S_{c_t}(j, q)\right).$$

Compared to the baseline version [6], which combines all classifiers computed at each iteration $(1, \ldots K)$, we combine only classifiers which improve the classification error rate on the training data (cf. step 4). We found that this yields better results. Schapire *et al.* proposed a theoretical analysis of the number of rounds needed for boosting [6], but it tends not to give practical answers. Therefore, in our case, we use heuristics to estimate this number K: if the value α_k is always positive, the value of K is set to the smaller value of the number of features and the number of documents in each topic. We do not allow α_k to be negative. These rules yield good results.

5. RELEVANCE FEEDBACK TECHNIQUE

Researchers realized that it is hard for an average user to formulate a "good query." Therefore, for successful routing, aids for good query formulation should be provided to users. Hence, one of the most effective ways to improve the performance of a classifier is to find a manner to improve user queries.

Assume that $\vec{q}_{orig}$ represents the original user call, T the topic number, $\vec{t_i}$ the vector representing the t_i^{th} topic and Rel the set of relevant topics such that $|Rel| = R$. Hence, the classifier starts by computing the R best topics for the user-query $q_{\vec{orig}}$, builds the set Rel and then reformulates the query as follow [7]:

$$\vec{q}_{new} = \vec{q}_{orig} + \alpha_1 \frac{1}{R} \sum_{t_i \in Rel} \vec{t_i} - \alpha_2 \frac{1}{M-R} \sum_{t_i \notin Rel} \vec{t_i}, \qquad (4)$$

where α_1 and α_2 denote interpolation parameters $(\alpha_1 + \alpha_2 = 1)$. Intuitively, α_1 represents how far the new vector should be pushed toward the relevant documents, and α_2 represents how far it should be pushed away from the non-relevant ones. Therefore, the output is the best topic given by the classifier to this new query $\vec{q}_{new}$.

Note that this technique is applied only during the test phase. Although, originally used to refine user requests by asking them the rate documents, we employ it to reweight the query vector without additional user input. We use only the relevant topics derived from the initial classifier.

6. DISCRIMINATIVE TRAINING TECHNIQUE

Discriminative training has recently been proposed for natural language call routing [8] and has been shown to be highly effective in simplifying the classifier design and improving portability [9]. Instead of simple counting in conventional maximum likelihood training, the minimum classification error criterion is used in discriminative training of the routing matrix parameters. Classification accuracy and robustness are improved by adjusting the models to increase the separation of the correct class from competitors. The same framework is used in this paper.

7. CONSTRAINED MINIMIZATION TECHNIQUE

The possibility of building multiple classifiers and then combining them to obtain a more accurate one is of considerable interest. We consider here a combination strategy for three classifiers [10].

Suppose we have two uncorrelated classifiers C_1 and C_2 which predict the topics t_1 and t_2 respectively for a query q. When both classifiers agree $(t_1 = t_2)$, the topic result is the same as each of the classifiers. When they disagree, a third classifier is invoked as an arbiter. This third classifier may be explicitly trained on disagreements of the first two using minimum error training and can also make a choice only on a subset of topics. This subset may be computed according to the N-best topics proposed by each of the first two classifiers or according to a confusion measure. For exemple, when the first two classifiers disagree, we take the set of confusible topics $ST_1 = \{t_{i_1}\}$ in C_1 and $ST_2 = \{t_{i_2}\}$ in C_2. Then, the third classifier C_3 chooses among the topics in the union of these subsets $(ST_1 \cup ST_2)$.

Let $\hat{t}_{C_i}$ denote the best topic chosen by the classifier C_i to the query q. Hence, the set of confusible topics for C_i are those $ST = \{t_i\}$ with a distance to $\hat{t}_{C_i}$ smaller than a threshold computed according to the average distance between $\hat{t}_{C_i}$ and the correct one on the training set. In this paper we chose the Kullbach-Leibler distance.

8. EXPERIMENTS

8.1. Database

Experiments were performed on two call routing tasks, a banking task with USAA and a UK operator task. We used the same training and test sets collected for the USAA call routing task as reported in [1], consisting of a total of about 4000 calls, routed to

23 destinations. In addition, experiments were performed using all of the training and test sets from the OASIS corpus as described in [11], consisting of $7,400$ tokens for training (T_1) and $1,000$ for testing (T_2), routed to 15 destinations. We denote this database BT. Some results are not the same as previously reported in [9] because a different set of unigram features is used in this paper.

Experiments are reported on both human transcriptions and ASR recognized strings. We use real-time recognition results which have a word error rate of about 30% for USAA and 48.1% [11] for the BT task.

8.2. Single Classifier Improvement

The main issue we want to investigate here is the accuracy improvement of one single type of classifier. We present in Table 1 the classification error rate of the beta and cosine classifiers using boosting, automatic relevance feedback (ARF) and discriminative training (DT). We show the impact of these techniques on each classifier separately. Hence, we do not investigate a combination between the beta and cosine classifiers in this table.

	Baseline	DT	Boosting	ARF
BT Human Transcription				
Cosine	47.3%	24.7%	35.5%	40.2%
Beta	26.6%	25.3%	25.4%	26.2%
BT ASR Recognized Strings				
Cosine	53.2%	37.0%	45.7%	43.8%
Beta	38.7%	38.2%	38.3%	38.4%
USAA Human Transcription				
Cosine	9.4%	6.1%	7.1%	9.4%
Beta	12%	5.5%	8.7%	12.0%
USAA ASR Recognized Strings				
Cosine	12.0%	8.4%	10.0%	12.0%
Beta	14.9%	7.8%	12.0%	15.3%

Table 1. Classification error rate of cosine and beta classifiers with the use of improvement techniques.

Results show that all the techniques improve the classifier accuracy, especially discriminative training. On the BT data this technique increases the accuracy of the cosine baseline by 30% on ASR recognized strings and 45% on human transcription. The improvement can be as high as 47% when we use discriminative training on the USAA data with the beta classifier. For this reason we present on Table 2 the impact of discriminative training when combined with other techniques.

We think that the beta and cosine classifiers have reached a local optimum with discriminative training, which explains the lack of significant improvement when combined with other techniques. We also arrive at the same conclusion as Schapire for boosting [6]: the improvement in accuracy is clearly dependent on the data and the classifier. The better the initial classifier, the less the improvement from boosting; the cosine classifier is improved by 25%, although the beta classifier is improved by only 5%. For the relevance feedback technique, we also conclude that the reformulation of the user request can help the classifier, specifically when the accuracy is low. Indeed, in the case of the cosine classifier, we get an improvement of 15% on human transcription and of 18% on ASR recognized string. This improvement becomes small or negligible when the accuracy increases, e.g. for beta or when we use discriminative training. Note that results are sensitive to the value of the number of relevant topics; with a value of R equal to 1 or 2 we obtain the best results (cf. §5).

	Baseline+DT	Boosting+DT	ARF+DT
BT Human Transcription			
Cosine	24.7%	26.3%	24.7%
Beta	25.3%	24.9%	25.2%
BT ASR Recognized Strings			
Cosine	37.0%	37.3%	37.0%
Beta	38.2%	37.6%	38.0%
USAA Human Transcription			
Cosine	6.1%	5.5%	6.1%
Beta	5.5%	5.2%	5.2%
USAA ASR Recognized Strings			
Cosine	8.4%	8.1%	8.4%
Beta	7.8%	8.4%	8.1%

Table 2. Classification error rate of cosine and beta classifiers using discriminative train and other techniques.

8.3. Multiple Classifier Combination

In the following, we investigate different methods to combine the two classifiers beta and cosine. First, linear interpolation (LI) of the two baseline classifiers is used. Second, we interpolate the cosine and beta classifiers after boosting (LI+Bost).

In addition, the constrained minimization technique is investigated to combine the beta and cosine classifiers: we consider C_1 and C_2 as the beta and cosine classifiers respectively, trained on T_1. When both classifiers agree, the topic result is the one agreed upon. When they disagree, a third classifier C_3 is invoked as an arbiter. This classifier is trained on the disagreements of the first two; the classifier type for C_3 is chosen to be the one with the smaller classification error rate, which is beta in our case. This classifier C_3 proceeds only on a subset of topics computed according to a confusion measure (cf. §7); we denote this approach *CM_D*. Another experiment is also done in which C_3 makes a choice between the N-best topics from C_1 and C_2; denoted *CM_N*. In our experiment, the value of N is set to 2 which gives the best result.

We present in Table 3 the classification error rate of the combined classifiers; we show results with and without the use of discriminative training.

Experiments show that the combination between these two classifiers is a good way to improve the performance. In fact, just a linear interpolation between them (25.5%) on BT data increase the accuracy of the best baseline classifier (26.6%) by 4%. Moreover, the use of boosting (24.6%) allows us to get more than 7% improvement. On the other hand, constrained minimization did not give the improvement we had expected on BT data. The reason is the error rate of the third classifier C_3 trained on the disagreement between the cosine and beta classifiers is quite high on the entire test set, about 65%.

Hence, to better use this technique, we build a new classifier with a higher accuracy for C_3. Let classifiers C_1 and C_2 represent the cosine and beta classifiers, respectively, discriminatively trained on the entire training corpus T_1. Then, let C_3 represent the better classifier between beta and cosine trained also on T_1. During classification, C_3 disambiguates among only a subset of topics computed according to a confusion measure (cf. §7). We

	LI	LI+Bost	CM_D	CM_N
BT Human Transcription				
cosine + beta	25.5%	24.6%	24.7%	25.0%
(cosine+beta) + DT	24.4%	24.2%	24.3%	24.6%
BT ASR Recognized Strings				
cosine + beta	38.9%	37.9%	38.0%	38.2%
(cosine+beta) + DT	37.2%	36.7%	37.0%	37.3%
USAA Human Transcription				
cosine + beta	10.4%	7.1%	7.8%	8.1%
(cosine+beta) + DT	5.8%	5.2%	5.8%	5.8%
USAA ASR Recognized Strings				
cosine + beta	12.7%	9.1%	10.0%	10.4%
(cosine+beta) + DT	7.8%	8.7%	8.7%	8.7%

Table 3. Classification error rate of different combination techniques between cosine and beta with and without the use of discriminative training.

denote this combined classifier CM^*. We present in Table 4 the classification error rate of this combination as well as a linear interpolation between these three classifiers (C_1, C_2, C_3).

	CM^*	Linear interpolation
BT Human Transcription		
Combined Classifier	23.8%	24.5%
BT ASR Recognized Strings		
Combined Classifier	36.5%	36.6%
USAA Human Transcription		
Combined Classifier	5.5%	5.8%
USAA ASR Recognized Strings		
Combined Classifier	7.8%	7.5%

Table 4. Classification error rate of three classifiers using constraint minimisation (CM^*) and linear interpolation.

As expected, the accuracy of this combination (CM^*) is better than those cited before in Table 3 and is also slightly better (2% on average) than a linear interpolation between the different classifiers. The combination of multiple classifiers CM^* (23.8%), improves the baseline version of beta (26.6%) by 10% approximately as well as the baseline version of cosine (47.3%) by 49% on the human transcription with BT data. The use of relevance feedback technique on this combined classifier does not result in significant improvement. We note also that in the constrained minimization technique, the use of a confusion measure for C_3 outperforms the accuracy of the combined classifier compared to the use of N-best topics; this is true in all experiments we have done.

9. CONCLUSION

In this paper, we first showed that boosting, automatic relevance feedback, and discriminative training can be used to improve the accuracy of a single classifier. Experiments showed a significant improvement in the accuracy of the system compared with baseline: 15% with automatic relevance feedback, 25% with boosting, and 45% with discriminative training. In general, the better the initial classifier, the less the improvement from these techniques. Discriminative training seemed to give better performance than boosting or relevance feedback. We also reported results of combining multiple types of classifiers using linear interpolation and constrained minimization. A further improvement of approximately 3% was obtained when three different classifiers were combined using either linear interpolation or constrained minimization. Although constrained minimization did not give significantly better accuracy than linear interpolation, a different training method for the arbitrating third classifier may give better results. One principle which can be drawn from the results in this paper is that a combination of uncorrelated classifiers can improve classification accuracy. Uncorrelated classifiers can be derived through different models or feature sets (such as cosine or beta) or through training with re-weighted data sets (boosting).

10. ACKNOWLEDGEMENTS

We gratefully acknowledge USAA and BT Retail and BTexaCT for the use of data collected for their tasks.

11. REFERENCES

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

[2] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, pp. 513–523, 1988.

[3] G. Salton, A. Wong, and C.S. Yang, "A vector space model for information retrieval," *Communication of the ACM*, vol. 11, no. 18, pp. 613–620, November 1975.

[4] B. Bigi, A. Brun, J.P. Haton, K. Smaili, and I. Zitouni, "A comparative study of topic identification on newspaper and e-mail," in *String Processing and Information Retrieval-SPIRE, IEEE Computer Society*, 2001.

[5] M.J. Kearns and L.G. Valiant, "Cryptographic limitations on learning boolean formulae and finite automata," *Journal of the Association for Computing Machinery*, vol. 1, no. 41, pp. 67–95, January 1994.

[6] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The annals of statistics*, vol. 5, no. 26, pp. 1651–1686, October 1998.

[7] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *Journal of the American Society for Information Science*, vol. 4, no. 41, pp. 182–188, 1990.

[8] H.-K. J. Kuo and C.-H. Lee, "Discriminative training in natural language call routing," in *Proc. ICSLP-2000*, Beijing, China, Oct. 2000.

[9] H.-K. J. Kuo and C.-H. Lee, "A portability study on natural language call steering," in *Proc. Eurospeech-01*, Aalborg, Denmark, 2001.

[10] P. Niyogi, J.B. Pierrot, and O. Siohan, "Multiple classifiers by constrained minimization," in *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, 2000.

[11] P. Durston, H.-K. J. Kuo, M. Farrell, M. Afify, D. Attwater, E. Fosler-Lussier, J. Allen, and C.-H. Lee, "OASIS natural language call steering trial," in *Proc. Eurospeech-01*, Aalborg, Denmark, September 2001.