RANDOM FORESTS-BASED CONFIDENCE ANNOTATION USING NOVEL FEATURES FROM CONFUSION NETWORK

Jian Xue and Yunxin Zhao

Department of Computer Science
University of Missouri, Columbia, MO 65211 USA
jxwr7@mizzou.edu zhaoy@missouri.edu

ABSTRACT

[1]In this paper, we propose a set of new features for confidence annotation, including three features derived from confusion network and one from statistical significance test. We also propose using Random Forests as confidence classifier. The new features are combined with a set of eight previously proposed confidence features, and the Random Forests is compared with Decision tree and Support Vector Machine. Experiments were conducted on telehealth captioning task with a vocabulary size of 46,489. Average confidence annotation accuracy of 84.69% was achieved on 5 doctors' test set. In addition, Random Forests was shown useful for feature importance ranking. The proposed features are shown important in confidence annotation and Random Forests achieved best results among the three classifiers.

1. INTRODUCTION

[1]Confidence annotation is an important topic in automatic speech recognition. For example, in telehealth automatic captioning system [1], misrecognized words may cause the patients with hearing loss to misunderstand doctors' meaning and cause undesirable problems. Confidence annotation can be used to classify each recognized word into either of two classes, 'correct' or 'incorrect', where 'incorrect' words may be ignored or corrected. In order to achieve accurate confidence annotation, effective features and classifiers for discrimination between correctly and incorrectly recognized words should be provided. Word posterior probability is one of the important features used in confidence annotation [2][3][4]. Word posterior probabilities obtained in CN [5] are drawing more attentions in recent years, where not only words in the best path but also words in competing paths are used in computing the probabilities. The CN based posterior probabilities were mostly used for improving word recognition accuracy, and they were used in confidence annotation in [6][7]. Besides word posterior probability, many speech decoder-based features have been proposed for confidence annotation, such as acoustic-model score, language-model score, language-model type, local word posterior probability based on state posterior probability [8], number of syllables of word, duration of word and so on. Task-specific features have been also proposed, such as parsed-based features in dialog system [9] and semantic features in communicator system [10]. Several classification techniques have been proposed for confidence annotation, for example, decision tree and Support Vector Machine (SVM) [4][9].

In the current work, we propose several novel confidence features, where three of them are based on confusion network. The CN-based features are entropy, posterior bigram and trigram probabilities. In addition, a P-value feature that is normally used in statistical significance test is proposed. For confidence classification, we propose using Random Forests [11][12], which is a large set of decision trees that collectively decide class category of each test sample. We also use Random Forests to rank features based on their importance in confidence annotation. Comparative experiments were carried out with decision tree and SVM based classifiers, where results show that Random Forests delivered best performance.

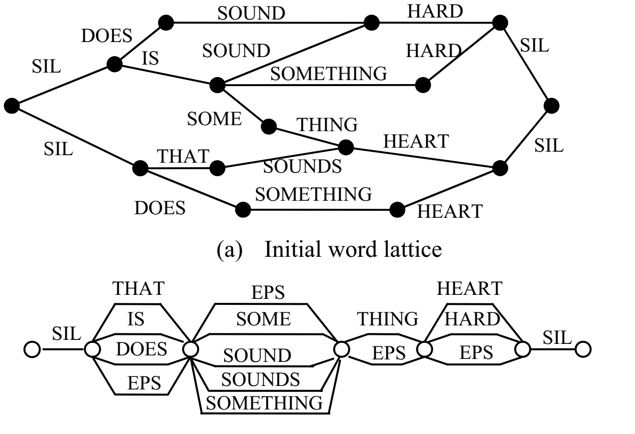
The rest of the paper is organized as follows. Section 2 introduces the novel features we proposed in this work. Section 3 introduces Random Forests and its trait in classification. In section 4 we describe the details of confidence annotation methods and in Section 5 we present experimental results. We conclude our work in Section 6.

2. NOVEL FEATURES

In this section we introduce four novel features for confidence annotation, where three are based on CN and one is from acoustic model.

2.1 Entropy for CN

CN is a linear graph transformed from word lattice [5], which aligns links in the original lattice and transforms the lattice into a linear graph in which all paths pass through all nodes. An example of confusion network is shown in Fig. 1.



(a) Initial word lattice

(b) Confusion Network

Fig. 1 An example of confusion network and its corresponding word lattice

This work is supported in part by National Institutes of Health under the grant NIH 1 R01 dc04340-01A2 and the National Science Foundation under the grant NSF EIA 9911095

Confusion network was utilized to generate word sequence hypothesis that minimizes expected word error rate. Given the alignment and the link posterior probabilities of a confusion network, the word sequence hypothesis with the lowest expected word error is obtained by picking the word with the highest posterior probability at each position in the alignment. The word posterior probability included in the confusion network is a good confidence feature. Besides word posterior probabilities, here we consider the entropy of words for each position of CN based on the word posterior probabilities derived for CN. Entropy measures the difference of word posterior probabilities among words in the same position in the CN, and ambiguity of word identity can be better captured by entropy than the word posterior probability alone. Entropy for CN is defined as:

$$En(w \mid O) = -\sum_{i=1...m} p(w_i \mid O) \log p(w_i \mid O),$$

where w is the word in the recognition output. w_i, i=1…m are the words that are in the same position with w in CN, and w is one of w_i's.

2.2 Bigram, trigram posterior probabilities in CN

Similar to word posterior probability, bigram and trigram posterior probabilities are also word-level posterior probabilities in CN, but these two are conditional probabilities given the context in the recognition output, which are defined respectively as following:

$$P(w_i \mid w_{i-1}, O) = \frac{p(w_i, w_{i-1} \mid O)}{p(w_{i-1} \mid O)}$$

$$P(w_i \mid w_{i-1}, w_{i-2}, O) = \frac{p(w_i, w_{i-1}, w_{i-2} \mid O)}{p(w_{i-1}, w_{i-2} \mid O)},$$

where w_{i-2}, w_{i-1}, w_i are consecutive words in the recognition output w_1, … , w_{i-2}, w_{i-1}, w_i, …, w_n. The joint posterior probability $p(w_i, w_{i-1} \mid O)$ and $p(w_i, w_{i-1}, w_{i-2} \mid O)$ can be computed using forward-backward algorithm in word lattice in a similar way as the single word posterior probability $P(w_i \mid O)$.

2.3 P-value

P-value is a very important concept in classic statistics. For a one-dimension Gaussian distribution $N(\mu, \sigma^2)$ with a known σ^2 and an unknown μ, one need to test if μ equals μ_0 or not. Given an observation x, the P-value is defined as $P_v(x)$ = $\Pr ob(| X - \mu |>| x - \mu | |\mu = \mu_0)$, which is the shaded area in Fig. 2.

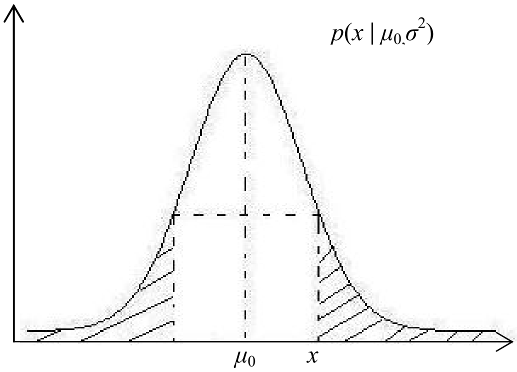


Fig. 2 P-value for a Gaussian distribution

The larger $P_v(x)$ is, the higher confidence we will have that μ equals μ_0.

For classification, we need to test if an observation x belongs to some class C. If the class-conditional distribution is a Gaussian distribution $N(\mu_0, \sigma^2)$, we can use $P_v(x)$ as a confidence feature. Compared with acoustic likelihood score, P-value captures information of distribution spreadness more effectively.

In speech recognition, tied state is modeled by a multivariate Gaussian mixture distribution and diagonal co-variance matrix is often used. We therefore define the P-value $P_{v,i}(\underline{x})$ for each n-dimension Gaussian distribution as the product of the P-values on individual dimensions. The P-value of an n-dimension Gaussian mixture distribution is then defined as

$$P_v(\underline{x}) = \max_i(w_i P_{v,i}(\underline{x})) \quad \text{or} \quad P_v(\underline{x}) = \sum_{i=1...M} w_i P_{v,i}(\underline{x}),$$

where w_i is the ith mixture weight. Our empirical evaluation showed that there was no significant difference between these two definitions.

For a word w in the recognition results, denote x_t, t = 1,2,…,T as the corresponding acoustic observation sequence and M_t, t = 1,2,…,T as the corresponding model sequence. We then define the log P-value for word w as

$$\log P_v(w) = \sum_{i=1}^{T} \log(P_v(x_t \mid M_t))$$

3. RANDOM FORESTS

Random Forests is a classifier based on the algorithm developed by of Leo Breiman and Adele Cutler [11] [12]. The classifier uses large number of decision trees. To classify a new object, the object is sent to each tree in the forest. Each tree gives a classification for the object and the forest chooses the classification having the most votes.

Each tree in the forest is grown as follows. First, choose N samples randomly with replacement from the original training dataset for growing the tree. Second, select m variables randomly out of total M variables and use the best split determined by these m variables to split the node. The value of m is held constant during the forest growing. Third, each tree is grown to the largest extent possible, without pruning.

The Random Forests error rate depends on the following two factors. The first is the correlation between any two trees in the forest, where increasing the correlation increase the error rate. The second is the strength of each individual tree in the forest. A tree with a low error rate is a strong classifier, and increasing the strength of the individual trees decreases the forest error rate.

Reducing m reduces both correlation and the strength, while increasing it increases both. Therefore a proper value for m is needed. In general, m is set to be approximately the square root of M.

Random Forests is considered unexcelled in accuracy among current classification techniques, and can handle thousands of input variables without variable deletion [11]. Random forests do not over-fit as more trees are added.

Random Forests can give estimates of what variables are important in the classification. The value of importance is the total decrease in node impurities from splitting on the variable, averaged over all trees. The node impurity is measured by the Gini index. For further details of Random Forests, please see [11] [12].

4. CONFIDENCE ANNOTATION

In this section we describe our work for confidence annotation in telehealth system, including features and classification techniques we used.

4.1 Features

The complete set of features include the novel features we introduced in Section 2 and eight other features that were previously proposed. The features are categorized as decoder-based features and CN-based.

Decoder-based features

- Acoustic-Score: Acoustic model score of a word.
- Language-Score: Language model score of a word.
- Language-Type: Language model type of a word, backoff or not.
- Total-Score: AS + lmscale*LS, where Lmscale is the weight of language model score used in decoder.
- P-value: P-value of a word, introduced in 2.3.
- Ave-Pvalue: Average P-value, defined as P-value divided by the duration of the word.
- LWPP: Local word posterior score.
- Ave-LWPP: Average LWPP, defined as LWPP divided by the duration of the word.

The feature LWPP was proposed by Dong et al [8]. To define LWPP, the posterior probability of the state s_i conditioned on the observation x is defined as

$$p(s_i \mid x) = \frac{p(x \mid s_i)p(s_i)}{p(x)} = \frac{p(x \mid s_i)p(s_i)}{\sum_{s_j \in D} p(x \mid s_j)p(s_j)},$$

where D is the set of all states survived after pruning. By assuming that the prior probabilities of all states are the same, the formula is simplified as

$$p(s_i \mid x) = \frac{p(x \mid s_i)}{\sum_{s_j \in D} p(x \mid s_j)}$$

Assuming for the word w the state sequence and the observation sequence are $s_m, \ldots s_n$ and $x_m, \ldots x_n$, the LWPP of w is defined as:

$$LWPP(w) = \log[\prod_{i=m}^{n} p(s_i \mid x_i)]$$

CN-based features

- Entropy: Entropy for CN, introduced in 2.1.
- Pos-Score: Word posterior score in CN.
- Bi-Pos: Bigram word posterior score in CN, introduced in 2.3.
- Tri-Pos: Trigram word posterior score in CN, introduced in 2.4.

4.2 Classification Techniques

Three classification techniques were investigated: Random Forests, Decision Tree and SVM. The statistical software R [14] was used in constructing the three classifiers. For Random Forests, we also examined the ranks of the features based on their importance in confidence annotation.

5. EXPERIMENTAL RESULTS

Experiments were conducted on the telehealth system. The system was trained by speech data collected in telehealth and five medical doctors have served as health care providers [1]. An average captioning word accuracy is 78.14%. Our decoder engine is TigerEngine 1.1 [1], and we use fast CN algorithm [13] to get the CNs.

For Random Forests the value of m was 4, the value of N was 0.632 time of sample size, and the number of trees was 500. The kernel function we used for SVM was radial function $K(x_i, x_j) = \exp(-\gamma \mid x_i - x_j \mid^2)$. γ was set to $1/(data \ \dim ension)$ and slack variable ξ was 0.1.

To avoid over-fitting, we used 10-fold cross-validation for each technique, i.e., divided the dataset randomly into 10 equal-sized subsets. We kept one of the 10 subsets as the validation set, and combined the remaining 9 subsets to form the training set. Repeating this 10 times, the accuracy is averaged on the 10 validation sets. The annotation error rate and word error rate are summarized in Table 1, where

$$Annotation \ \ Error \ \ Rate = \frac{Number \ \ of \ \ incorrect \ \ annotation}{Total \ \ number \ \ of \ \ annotation}$$

$$Word \ \ Error \ \ Rate = \frac{Substitutions + Insertions}{Total \ \ number \ \ of \ \ words \ \ in \ \ hypotheses}$$

It is clear that Random Forests achieved best results for each doctor, where the false alarm rate was from 3.58% to 4.76%.

	Word Error Rate	Annotation Error Rate		
		Decision Tree	SVM	RF
Dr. 1	15.54%	13.54%	13.50%	13.16%
Dr. 2	19.01%	18.22%	18.52%	17.55%
Dr. 3	21.46%	18.34%	18.34%	17.83%
Dr. 4	18.35%	16.05%	16.41%	15.90%
Dr. 5	15.50%	13.90%	13.40%	13.26%

Table 1 Performance of confidence annotation

Fig. 2 shows importance of different features ranked by Random Forest, averaged on different data sets. From left to right the features are: Acoustic-Score, Language-Score, Language-Type, Total-Score, P-value, Ave-Pvalue, LWPP, Ave-LWPP, Entropy, Pos-Score, Bi-Pos and Tri-Pos.

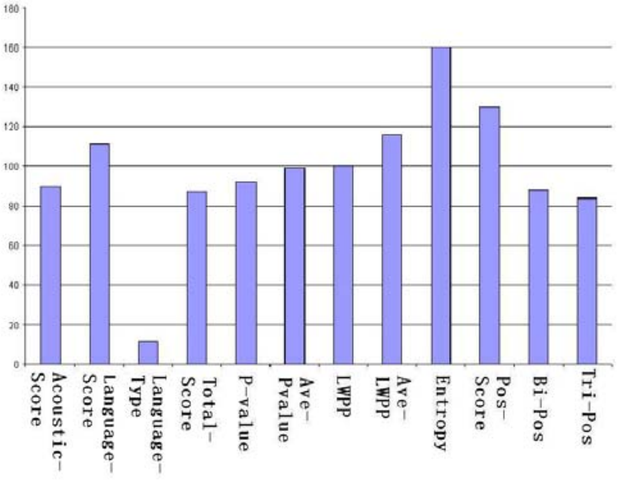


Fig. 3 Chart of importances of features

From Fig. 3 we observe that Entropy is most important. Language-Type almost has no effect and it can be removed. Ave-Pvalue ranks fifth among all of the features.

To further measure the importance of the proposed features, we get the performance of confidence annotation on Dr. 1's dataset using different combination of features by using the Random Forests. Starting from using all 12 features, individual features were selectively removed to see its effect on annotation error rate. The results are summarized in Table 2.

	Annot. Error
Using all features	13.16%
No Entropy	13.50%
No Pos-Score	13.20%
No Entropy and Pos-Score	14.14%
No P-value and Ave-Pvalue	13.40%
No Bi-Pos and Tri-Pos	13.34%
No Entropy, P-value, Ave-Pvalue, Bi-Pos and Tri-Pos	14.16%

Table 2 Performance of confidence annotation using different combination of features

From Table 2 we observe that when Entropy was not used, error rate increased by 0.34%. Since Entropy and Pos-Score are correlated to some extent, when we delete them both, the performance decreased by 1.02%. From table 2 we also observe that when P-value and Ave-Pvalue were not used, error rate increased by 0.24%, and when Bi-Pos and Tri-Pos were not used, error rate increased by 0.18%. So P-value, Bi-Pos and Tri-Pos all played important roles in confidence annotation.

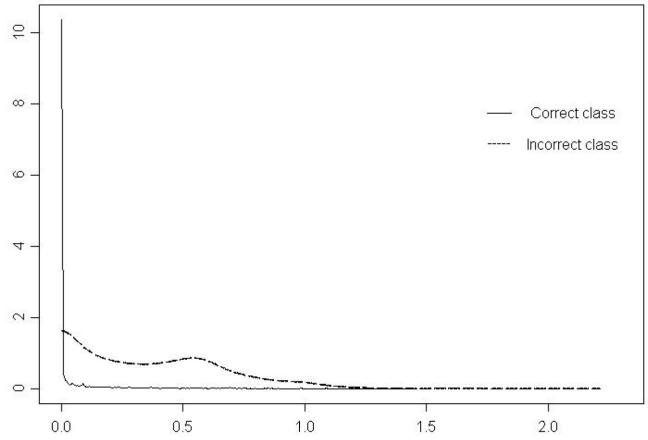


Fig. 4 Distribution of Entropy

To further understand the proposed features for confidence annotation, the distributions of Entropy for "incorrect" class and "Correct" class are shown in Fig. 4. In general, Entropy of words in "Correct" class are smaller than those in "Incorrect" class, and therefore Entropy is an important feature in confidence annotation.

No. of trees	Annot. Error
10	14.60%
20	14.14%
50	13.38%
100	13.20%
200	13.18%
500	13.16%
700	13.16%
800	13.18%

Table 3 Performance of confidence annotation training different number of trees

Table 3 summarized the results of Random Forests on Dr. 1's dataset when training different number of trees. From Table 3 we observe that as the number of trees increased, the error rate decreased to some extent. On our task, we can set the number of trees as 100 or more.

6. CONCLUSION

In this paper we described the work of confidence annotation in telehealth system. We introduced four novel features and the results shows that they are important in confidence annotation. We trained Random Forests, decision tree and SVM on five doctors' dataset and compared their performance. The results show that Random Forests get the best accuracy.

REFERENCES

[1] Y. Zhao et al, "An automatic captioning system for telemedicine", in proceedings of ICASSP 2006, May 2006.

[2] F. Wessel, K. Macherey, and R. Schlueter, "Using word probabilities as confidence measures", Proc. ICASSP, pp. 225-228, 1998.

[3] G. Evermann and P.C. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities", Proc. ICASSP, pp 1655-1658, 2000.

[4] Y. Fu and L. Du, "Combination of multiple predictors to improve confidence measure based on local posterior probabilities". Proc. ICASSP, pp 93-96, 2005.

[5] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minization and other application of confusion network," Computer Speech and Language, vol. 14 (4), pp. 373-400, 2000.

[6] D. Hakkani-Tur, G. Riccardi and A. Gorin, "Active learning for automatic speech recognition". Proc. ICASSP, pp IV-3904 – IV-3907, 2002.

[7] R. Gretter and G. Riccardi, "On-line learning of language models with word error probability distributions". Proc. ICASSP, pp 557-560, 2001.

[8] B. Dong, Q. Zhao and Y. Yan, "A fast confidence measure algorithm for continuous speech recognition". Proc. Interspeech, pp 1457-1460, 2005.

[9] R. Zhang and A. Rudnicky, "Word level confidence annotation using combinations of features", Proc. Eurospeech, pp 2105-2108, 2001.

[10] R. Sarikaya, Y. Gao and M. Picheny, "Word level confidence measurement using semantic features", Proc. ICASSP, pp I-604 – I-707, 2003.

[11] Random Forests classifier description, site of Leo Breiman.

[12] L. Breiman, "Random Forest", Machine Learning 45 (1), pp 5-32, 2001.

[13] J. Xue and Y. Zhao, "Improved confusion network algorithm and shortest path search from word lattice", Proc. ICASSP, pp 853-856, 2005.

[14] The R Project for statistical Computing, http://www.r-project.org.