iCNC and iROVER: The Limits of Improving System Combination with Classification?

Björn Hoffmeister, Ralf Schlüter, and Hermann Ney

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6 - Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{hoffmeister,schlueter,ney}@cs.rwth-aachen.de

Abstract

We show how ROVER and confusion network combination (CNC) can be improved with classification. The general idea of improving combination with classification is that each word is assigned to a certain location and at each location a classifier decides which of the provided alternatives is most likely correct. We investigate four variations of this idea and three different classifiers, which are trained on various features derived from ASR lattices.

For our experiments, we use highly optimized ROVER and CNC systems as baseline, which already give a relative reduction in WER of more than 20% for the TC-STAR 2007 English task. With our methods we can further improve the result of the corresponding standard combination method.

Index Terms: speech recognition, system combination

1. Introduction

System combination is an indispensable part of state-of-the-art LVCSR systems. It is part of the complex decoding structure of modern recognizers and used to ultimately combine the results from different systems, e.g. [1] and [2]. In this paper we investigate the combination of lattices provided from different sites. The three standard approaches for lattice based system combination considered are ROVER, confusion network combination (CNC), and min.fWER combination. In Section 2 we shortly repeat these approaches and present a tuning scheme which allows us to optimize all free parameters together w.r.t. to minimum WER. Results on the TC-STAR 2007 English task show a reduction in WER of more than 20% relative w.r.t. to the best single system. These results serve as baseline for the improved combination methods we introduce in Section 3.

The idea of improving ROVER has been explored before in [3] and [4]. The first approach used a neural network fed with basic features derived from single best hypothesis only. The latter work introduced iROVER, the idea of improving ROVER with classification and various, lattice-based features. In this paper we further investigate iROVER and we explore two ways of improving CNC with classification. In the first approach we simply let the classifier decide whether the first or the second hypothesis is correct. This approach was successfully applied in [5] for improving CN decoding performance. The alternative approach assigns the iROVER idea to CNC and let the classifier choose between CNC and the single system's CN hypotheses. Other related work is [6], where the authors improve CN decoding by SVMs using features from especial acoustic models.

The features derived from the individual lattices are described in detail in Section 4. Section 5 introduces the three classifiers we apply in this work: Boostexter, random forests, and maximum entropy models. In Section 6 we present results for the various improved combination methods. We show that all improved methods gain over their standard counterparts, where the largest gains are observed for iROVER. The gains for improved CNC are rather small and we finish the section with an analysis of these results. In the final section we draw conclusions and discuss future work.

2. System Combination

The most simple combination technique is ROVER, which combines single best output from multiple ASR systems. The lattices are used only for providing word confidence scores, which we compute according to [7]. ROVER works in two steps. In the first step the hypotheses are aligned via a Levenshtein alignment with the time overlap between words as a local cost. The result of the alignment is a series of slots each containing a word per system together with its confidence. In case of a deletion the empty word together with a default confidence is inserted. Decoding is done for each slot separately, where the word with the maximum total confidence is chosen. Setting all confidences to 1 results in doing a majority vote.

The main idea of confusion network combination as proposed in [8] is similar to ROVER: instead of aligning the single best output, we align confusion networks which are built from the individual lattices. The advantages over ROVER are that each slot contains many hypotheses and hence CNC has a much lower oracle error rate. Each slot in the source CNs provides a posterior distribution over all words. In contrast to ROVER, we can now compute for each slot and word the combined word posterior probability over all systems. The word posterior probability for the combined CN is the weighted average of the individual slot-wise posteriors. In decoding we choose the word with the highest combined posterior probability.

We build CNs directly from lattices using our own algorithm which is based on [9], but uses a pivot path in order to accelerate the CN construction as proposed in [10].

The min.fWER combination method was introduced in [11]. It is motivated by the observation that frame error (fWER) and true WER are highly correlated. In a first step the union over all lattices is built and frame-wise word posterior probability distributions are computed. These are used to rescore each arc in the union lattice with the expected, normalized frame error. In the decoding step the path with the minimum expected frame error is chosen.

In our experiments we have lattices for each system c, with a total of C systems. For each system we are provided with acoustic and language model (LM) scores. For calculating lattice forward-backward scores we set the scale of the acoustic scores to α_c/β_c and the LM scale to α_c, where β_c is the original LM scale of system c and α_c is initialized with 1. When combining posterior probabilities among systems, we assign each system a weight w_c. First, we optimize for each combination method the parameters α_c and w_c, $c \in [1, C]$. Optimization starts with a grid search over the weights followed by the separate tuning of each α_c. Finally, we use the result from the previ-

ous step as starting point for the Nelder-Mead downhill simplex optimization algorithm (AMOEBA). The preceding grid search turned out to be necessary for finding a starting point for downhill simplex that eventually converges to a good local minimum. For min.fWER combination we include the normalization factor for the frame error in the optimization loop.

3. Improving System Combination with Classification

The core idea of ROVER and CNC is to cluster lattice arcs into slots and thus reduce the global decoding task to a local decision problem. The resulting decision rules are simple and rather ad-hoc in the case of ROVER. In contrast, for CNC the decision rule minimizes the expected WER, but the decision is biased by the underlying CNs, which do not provide the true word posterior probabilities, e.g. [12]. Both decision rules use only a fraction of the information provided by the lattices. Thus, we expect to improve combination accuracy by replacing the decision rules by classifiers, which can make favorable use of all the available information.

3.1. iROVER

The idea of replacing the decision rule of ROVER by a classifier was explored in [3] and [4]. In the latter work we aimed to improve performance by classification and by enhancing ROVER with information from min.fWER combination. Here, iROVER refers to the approach where the original ROVER alignment is used and the classifier can choose from the C systems. The idea of coupling min.fWER combination and ROVER is implemented by treating the min.fWER hypothesis as an additional system, which is added to the hypotheses of the C systems. The ROVER alignment is now computed over the $C+1$ hypotheses, where the classifier can also choose from the $C+1$ alternatives.

3.2. iCN and iCNC

We investigate two approaches for improving CNC decoding by classification. The first approach follows [5]: we simply decide among the N-best hypotheses for each slot, where the hypotheses are ordered according to their combined slot-wise word posteriors. We refer to this method as iCN(N=$\cdot$). Choosing N=2 already gives an oracle error rate lower than the corresponding ROVER oracle error rate. The second approach, which we call iCNC, follows directly the iROVER+min.fWER method: we let the classifier choose between the CNC hypothesis and the C hypotheses derived from the CNs of the individual lattices.

4. Features

We generate a feature vector for each slot of the ROVER(CNC) alignment, which is then used as input for classifier training.

4.1. Lattice pre-processing

We normalize the lattices following the procedure described in [4]. Here, in vocabulary normalization we observed a problem in dealing with contractions like "it's". Ideally, we would preserve the form hypothesized by the recognizer, but it turned out that some systems model the difference only in the pronunciation, e.g. the phrase "it_is" is associated with the pronunciations for "it's" and "it is". And we observed that in the reference transcription almost solely the expanded form is used. As a result, when we left the contractions as they are, the classifiers started to learn to expand "it's" to "it is". On the other hand when expanding all contractions we loose accuracy in the time marks which causes a consistent performance decrease of 0.1 or more for min.fWER combination, which highly depends on correct time marks. In the end we decided to do the expansion, because we assume it gives the lower bias to classifier training.

Table 1: *Baseline results for eval07, WER[%].ROVER results in brackets are produced with majority voting.*

System	Viterbi/ ROVER	CN(C)	min.fWER (comb.)
1	9.4	9.0	9.0
2	9.8	9.5	9.6
3	10.2	10.1	10.1
4	9.8	9.8	9.8
1+2	8.1(9.4)	7.4	7.6
1+2+3	7.8(7.9)	7.1	7.4
1+2+3+4	7.5(7.5)	7.1	7.2

4.2. Features Sets

Throughout our experiments we use different feature sets. For each word hypothesis we compute a set of *word features*. The set includes the acoustic and LM score, word duration, the number of characters, and the averaged character duration, which serves as an approximation of the phoneme duration. Furthermore, we add the information if the word is in the list of the 10, 20, or 100 words causing the most errors. In our first work we included the word identity as a feature, but further experiments indicated that the feature is not helpful.

The second set of features includes all features derived from lattice posterior probabilities. This includes CN confidence and CN slot entropy of the system that hypothesized the word and the additional CNC confidence and slot entropy for the iCN(C) approaches. Furthermore, we calculate confidences based on frame-wise posterior probabilities across all systems and from the combined frame-wise posterior probability distributions. *Cross-system confidence* means that we use the statistics from system A to estimate a confidence for a hypothesis of system B. A classifier can use these confidences as an indicator for OOV words. We refer to this feature set as *posterior features*.

For each slot the main feature vector consists of the word features for all hypotheses combined in the slot. We consider the slot context by adding the minimum distance in seconds to the preceding(subsequent) slot and the word features for all hypotheses in the preceding(subsequent) slot. For the iROVER approaches we provide a binary feature indicating whether a hypothesis matches the ROVER or the min.fWER choice (iROVER+min.fWER only). An equivalent feature indicates for iCNC whether a hypothesis matches the CNC vote.

For experiments using posterior features we add for each hypothesis the posterior features from all systems.

5. Classifier

In our previous work, [4], we employed Boostexter (BT), [13]. The idea of BT is to learn a series of weak classifiers (decision stumps) and reweight the training examples using Adaboost, real Adaboost.MH with logistic loss in our experiments. An alternative are random forests (RF), which replace the weak classifier by a more powerful full decision tree and use randomization instead of boosting. The RF implementation used is the Randomized C4.5 as suggested in [14]. Randomization can outperform boosting in the presence of noisy training data, because boosting can start to focus on the wrongly labeled examples. We label the training data by performing an alignment between the aligned CNs (or the aligned single best hypotheses in the ROVER case) and the reference. Here, we have at least three sources of error: the oracle alignment is ambiguous, the CNs only approximate the minimum WER alignment of the hypotheses, and even with a careful pre-processing we still have mismatches due to imperfect text normalization.

Table 2: *Training and test corpora statistics for* TC-STAR *2007.*

	#features	#samples train	#samples test
2 systems			
iROVER	75	3,032	3,215
iROVER+min.fWER	108	3,115	3,301
iCNC	71	647	659
iCN(N=2)	99	28,900	26,961
3 systems			
iROVER	111	4,237	4,386
iROVER+min.fWER	147	4,207	4,416
iCNC	94	1,709	1,801
iCN(N=2)	126	32,624	30,069
4 systems			
iROVER	149	5,320	5,178
iROVER+min.fWER	188	5,346	5,207
iCNC	166	3,696	3,354
iCN(N=2)	157	33,252	30,504

As an alternative to the two decision tree based approaches we present results using a Maximum Entropy (Maxent) model, [15]. In speech recognition Maxent taggers have been successfully applied to ASR post-processing tasks, e.g. for meta data extraction [16].

We build the training and test sets for the classifiers from the confusion network produced by either ROVER or CNC. From an oracle alignment we know for each slot the according reference word. For some slots the reference word is not included and we label these examples as *rejects*. If a reject is hypothesized in decoding, we back off to the baseline hypothesis.

In the iCN approach the remaining examples are labeled with the position of the matching hypothesis, where we use in training as well as in testing only slots having more than one alternative. For the other approaches we label each example with the numbers of all the systems hypothesizing the correct word and thus allowing multi labels. BT can directly handle multi labels in training, unlike RF and Maxent. For the latter we explored two approaches for converting the multi label to a single label problem. The first approach is to build a new label set which consists of one label for each combination of the original labels that occur in training. In preliminary tests this approach worked best for Maxent. In classification the Maxent model assigns a probability to each label, which we count back to probabilities for the original labels.

The investigated alternative is to tackle the multi label problem by doing a *One-vs.-All* classification. For each label we build a binary classifier. In classification we apply each classifier and choose the label with the highest score. This approach worked best for random forests, where the score used is simply the number of trees that voted for the given label.

6. Experiments

We present results for the combination of up to four lattice sets from the English Task of the TC-STAR 2007 Evaluation Campaign. Our TC-STAR project partners kindly provided us their lattices. The task and the applied systems are described in [17], [18], and [1]. We perform parameter tuning and classifier training on the development set. All results presented in this paper are produced on the evaluation set. The baseline results are produced on the pre-processed lattices and are summarized in Table 1.

6.1. Experimental Setup

For each classifier we extract training and test samples as described in Section 5. Table 2 shows the statistics for the different tasks. The feature dimensionality for iCNC is lower than for iROVER, because we consider only the CNC hypothesis as left(right) context. Due to the limited training data we apply 10-fold cross-validation for tuning the classifier parameters. With the optimal parameter set we train the final classifier on the complete data. For Boostexter and Maxent the optimized parameter is the number of iterations, which we choose for each task separately. Random forests proved not to be very sensitive to parameter tuning: in the end we used C4.5 with default options and 100 trees for all tasks.

Table 3: *iROVER combination results for eval07, WER[%].*

iROVER	2 systems	3 systems	4 systems
word features			
Boostexter	7.9	7.6	7.6
Random forests	7.9	7.6	7.4
Maxent	7.9	7.8	-
word and posterior features			
Boostexter	7.6	7.4	7.3
Random forests	7.7	7.4	7.2
Maxent	7.8	7.6	-

Table 4: *Combination results with Boostexter (BT) and random forests (RF) as classifier for eval07, WER[%].*

		2 systems	3 systems	4 systems
iROVER	BT	7.6	7.4	7.3
	RF	7.7	7.4	7.2
iROVER +min.fWER	BT	7.6	7.2	7.0
	RF	7.5	7.3	7.0
iCNC	BT	7.4	7.1	6.9
	RF	7.4	7.1	6.9
iCN(N=2)	BT	7.5	7.2	7.1
	RF	7.4	7.1	7.0

6.2. Results and Analysis

In our first set of experiments we explore the influence of the posterior features on overcoming simple ROVER. From Table 3 we learn that using a classifier with only the simple word features improves considerably over ROVER. Adding the posterior features boost iROVER performance to the level of min.fWER combination.

Table 4 shows the results of a direct comparison of the four i approaches. As expected, iROVER+min.fWER goes beyond iROVER and can take over min.fWER combination but fails on improving over CNC. We suppose that iROVER+min.fWER can compensate for some of the errors described in Section 4. iCNC performs best and can slightly improve over CNC. Encouraged by the results presented in [5] we hoped to see nice improvements for iCN, but disappointingly it is only able to compensate for a few errors.

Boostexter and random forests are mostly on the same level with some advantages for RF. Especially for the hard iCNC and iCN tasks RF seems to be more robust. We applied Maxent only to a few tasks, where the results are slightly worse than BT and RF. Recently, we started some experiments using second order posterior features which seems to help for Maxent, unlike BT and RF. The Maxent toolkit used applies the General Iterative Scaling (GIS) algorithm which causes extremely long runtime, e.g. 100K iterations for the iROVER/2-system task and 1M for the iROVER/3-system task, without giving an advantage over BT and RF, which is eventually the reason why we don't present more Maxent results.

We expected iCNC and iCN to perform better. Oracle error rates show that there is much latitude for improvements for all approaches, with the lowest oracles for iCN(N=2). We tried iCN(N=3) which gives a nice improvement in oracle but not in true error rate.

Table 5: *Error detection and correction results for eval07 for four systems and with a random forest as classifier.*

	error detection		error correction	
	recall	prec.	recall	prec .
iROVER	0.22 (357/1,658)	0.7 (357/514)	0.16 (269/1,658)	0.52 (269/514)
iROVER +min.fWER	0.16 (267/1,629)	0.72 (267/369)	0.12 (189/1,629)	0.51 (189/369)
iCNC	0.14 (153/1,086)	0.71 (153/215)	0.1 (104/1,086)	0.48 (104/215)
iCN(N=2)	0.08 (153/2,018)	0.63 (153/244)	0.06 (113/2,018)	0.46 (113/244)

For a further analysis of the different approaches we looked at the error detection and correction statistics. For a sequence of slots S we denote the reference word for a slot $s \in S$ as $\mathrm{w}_{s,\mathrm{ref}}$, the baseline hypothesis as $\mathrm{w}_{s,\mathrm{base}}$ (either the ROVER, min.fWER, or CNC hypothesis), and the classifier hypothesis as w_s. Now, we define recall and precision for error detection and correction as follows:

$$\mathrm{rec}_{\mathrm{detect}}(S) := \frac{\sum_{s\in S} 1\{\mathrm{w}_s \neq \mathrm{w}_{s,\mathrm{base}} \wedge \mathrm{w}_{s,\mathrm{base}} \neq \mathrm{w}_{s,\mathrm{ref}}\}}{\sum_s 1\{\mathrm{w}_{s,\mathrm{base}} \neq \mathrm{w}_{s,\mathrm{ref}}\}}$$

$$\mathrm{prec}_{\mathrm{detect}}(S) := \frac{\sum_{s\in S} 1\{\mathrm{w}_s \neq \mathrm{w}_{s,\mathrm{base}} \wedge \mathrm{w}_{s,\mathrm{base}} \neq \mathrm{w}_{s,\mathrm{ref}}\}}{\sum_s 1\{\mathrm{w}_s \neq \mathrm{w}_{s,\mathrm{base}}\}}$$

$$\mathrm{rec}_{\mathrm{correct}}(S) := \frac{\sum_{s\in S} 1\{\mathrm{w}_s \neq \mathrm{w}_{s,\mathrm{base}} \wedge \mathrm{w}_s = \mathrm{w}_{s,\mathrm{ref}}\}}{\sum_s 1\{\mathrm{w}_{s,\mathrm{base}} \neq \mathrm{w}_{s,\mathrm{ref}}\}}$$

$$\mathrm{prec}_{\mathrm{correct}}(S) := \frac{\sum_{s\in S} 1\{\mathrm{w}_s \neq \mathrm{w}_{s,\mathrm{base}} \wedge \mathrm{w}_s = \mathrm{w}_{s,\mathrm{ref}}\}}{\sum_s 1\{\mathrm{w}_s \neq \mathrm{w}_{s,\mathrm{base}}\}}$$

Table 5 gives the performance obtained with RF as classifier applied on the four systems task; the results for the other corpora and classifiers show the same tendencies. We see that for iROVER, iROVER+min.fWER, and iCNC the precisions remain almost constant, whereas the recall values decrease. This suggests that the iROVER approaches mostly compensate for errors that were already wiped out by CNC and thus iCNC improves only slightly. Looking at the hypotheses produced by the three approaches support the conclusion. Comparing iCNC and iCN we see that the absolute number of recovered and corrected errors is almost equal for both approaches, but iCN produces many more false positives. Thus, for the tested classifiers and features it helps to apply the ROVER constraint, i.e. to restrict the choice to hypotheses that occurred at least for one system as best hypothesis. That iCN behaves worse because of seeing more data implies that either the feature set or the modeling is still insufficient.

7. Conclusions and Outlook

We showed that classification can improve ROVER even on a set of very basic features available from all ASR systems. The usage of more sophisticated features derived from ASR lattices further improves ROVER. These features in conjunction with classification help as well to improve min.fWER and confusion network combination (CNC).

The improvements over a highly optimized CNC baseline are present but rather small and the question arises whether we are already at the limits of improving combination with classification. For a final judgement further research is required. We plan to overcome shortcomings of the presented features by exploring features of higher degree and from additional knowledge sources. At the classifier-side we consider conditional random fields an interesting approach as they are able to model long-term context dependencies.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

8. References

[1] J. Lööf et. al., "The RWTH 2007 TC-STAR evaluation system for european English and Spanish," in *Proc. Int. Conf. on Speech Communication and Technology*, Antwerp, Belgium, Aug. 2007.

[2] B. Hoffmeister et. al., "Cross-site and intra-site ASR system combination: Comparisons on lattice and 1-best methods," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, HI, USA, Apr. 2007.

[3] R. Zhang and A. Rudnicky, "Investigations of issues for using multiple acoustic models to improve continuous speech recognition," in *Proc. Int. Conf. on Spoken Language Processing*, Pitsburgh, PA, USA, Sept. 2006.

[4] D. Hillard et. al., "iROVER: Improving system combination with classification," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, Rochester, New York, Apr. 2007.

[5] L. Mangu and M. Padmanabhan, "Error corrective mechanisms for speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, USA, May 2001.

[6] V. Venkataramani et. al., "Ginisupport vector machines for segmental minimum bayes risk decoding of continuous speech," *Computer Speech and Language*, vol. 21, July 2007.

[7] F. Wessel et. al., "Using word probabilities as confidence measures," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1998.

[8] G. Evermann and P. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *NIST Speech Transcription Workshop*, College Park, MD, 2000.

[9] L. Mangu et. al., "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, 2000.

[10] D. Hakkani and G. Riccardi, "A general algorithm for word graph matrix decomposition," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, Apr. 2003.

[11] B. Hoffmeister et. al., "Frame based system combination and a comparison with weighted ROVER and CNC," in *Proc. Int. Conf. on Spoken Language Processing*, Pitsburgh, PA, USA, Sept. 2006.

[12] D. Hillard and M. Ostendorf, "Compensating for word posterior estimation bias in confusion networks," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[13] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, 2000.

[14] Th. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, vol. 40, Aug. 2000.

[15] D. Keysers et. al., "Efficient maximum entropy training for statistical object recognition," in *Informatiktage der Gesellschaft für Informatik*, Bad Schussenried, Germany, Nov. 2002.

[16] Y. Liu et. al., "The ICSI-SRI-UW metadata extraction system," in *Proc. Int. Conf. on Spoken Language Processing*, Jeju Island, Korea, Oct. 2004.

[17] D. Falavigna et. al., "The irst english-spanish translation system for european parliament speeches," in *Proc. Int. Conf. on Spoken Language Processing*, Antwerp, Belgium, Aug. 2007.

[18] S. Stüker et. al., "The isl 2007 english speech transcription system for european parliament speeches," in *Proc. Int. Conf. on Spoken Language Processing*, Antwerp, Belgium, Aug. 2007.