Iterative Decoding: A Novel Re-scoring Framework for Confusion Networks

Anoop Deoras [1], Frederick Jelinek [2]

Center for Language and Speech Processing, Johns Hopkins University
3400 N Charles St. Baltimore, MD 21218 USA
[1] adeoras@jhu.edu
[2] jelinek@jhu.edu

***Abstract*—Recently there has been a lot of interest in confusion network re-scoring using sophisticated and complex knowledge sources. Traditionally, re-scoring has been carried out by the N-best list method or by the lattices or confusion network dynamic programming method. Although the dynamic programming method is optimal, it allows for the incorporation of only Markov knowledge sources. N-best lists, on the other hand, can incorporate sentence level knowledge sources, but with increasing N, the re-scoring becomes computationally very intensive. In this paper, we present an elegant framework for confusion network re-scoring called 'Iterative Decoding'. In it, integration of multiple and complex knowledge sources is not only easier but it also allows for much faster re-scoring as compared to the N-best list method. Experiments with Language Model re-scoring show that for comparable performance (in terms of word error rate (WER)) of Iterative Decoding and N-best list re-scoring, the search effort required by our method is 22 times less than that of the N-best list method.**

I. Introduction

In the conventional Maximum a Posteriori (MAP) approach [1] to automatic speech recognition (ASR), one aims to find the most likely word sequence $\mathbf{W}^*$ such that $\mathbf{W}^* = \arg\max_W P(\mathbf{W}|\mathbf{A})$. It has been shown in bayesian decision theory literature [2] that the above mentioned formalism guarantees minimization of the sentence error rate of the recognition hypothesis. Empirically it has been found that there is a lot of co-relation between sentence error rate and word error rate, but the latter metric is more forgiving and also more relevant (to ASR applications). There have been many successful attempts at reformulating this source channel approach in order to directly aim for the word error rate (WER) [3] [4] of the recognition process. Consensus decoding [4], is one such approach in which lattices are converted into linear networks of confusion bins (also referred to as confusion networks), where word posterior scores are computed for each competing word in each confusion bin. Consensus decoding has shown improvements over the MAP approach on several speech corpora [4]. Due to the complexity of the speech recognition task, integration of complex and sophisticated knowledge sources (higher order and/or longer dependency language models, parsers etc) in the decoder search engine is just too computationally intensive and hence practically non-feasible. One way to achieve further reduction in word error rate is to re-score the confusion networks (obtained as the output of the first pass recognition) by external and sophisticated knowledge sources (KS).

Traditionally, for the purpose of language model re-scoring, researchers have relied on lattice dynamic programming re-scoring where the baseline language model scores in the word lattices are replaced by scores obtained from stronger language model following which the viterbi path is found in the re-scored lattice. State of the art language model (LM) toolkits such as SRILM [5] provide functionality for such re-scoring. Since confusion networks have posterior scores on the words, the re-scoring can be achieved either by (a) composing the Finite State Machine (FSM) representation of the LM and confusion networks, combining the posterior scores and new language model scores and then extracting the FSM least cost (viterbi) path *or* (b) by extracting N best lists from confusion networks, recomputing the new language model scores and interpolating them either with the posterior scores or with the acoustic scores for each individual word string and picking the top scoring hypothesis. Re-scoring using finite state machine representation (choice (a) above) is very efficient as compared to the N-best list re-scoring (choice (b) above).

However, in the FSM framework for confusion networks, the incorporation of acoustic information may not be easy and hence may not be a suitable platform for re-scoring. Also, when the external knowledge source is other than a simple n-gram language model (sentence level knowledge sources, longer dependency models, parsers etc), then its conversion into an FSM may not be possible. Hence in such cases, one has to resort to N-best list re-scoring methods. However N-best list re-scoring is sub-optimal because typically N is chosen to be of the order of a few thousand while the total number of paths extracted from confusion networks (or lattices) can potentially be huge. Empirically, it has been found that the oracle path in a confusion network often has many words which are assigned low posterior probability by the baseline recognizer. This indicates that the baseline models (both language model and acoustic model) are not representative enough of the test data to assign oracle paths a higher probability and other erroneous paths a lower probability. Thus during the re-scoring, it becomes crucial to consider even those hypotheses which were originally given lower probability by the baseline system. However the N-best list extracted from

978-1-4244-5480-8/09/$26.00 © 2009 IEEE

a confusion network are the few top scoring hypotheses and hence it is very likely that a corresponding re-scoring method will miss out on some oracle path.

As compared to word lattices, confusion networks are more favorable representations of word hypotheses. They have a much neater and more compact form than their corresponding lattices. Given their linear structure, one can carry out the re-scoring much more efficiently. Also the fact that the oracle word error rate of confusion networks is lower than that of lattices, make them a perfect starting point for re-scoring.

In this paper, we present a simple and efficient framework called *Iterative Decoding*. In this framework, integration of sophisticated and complex knowledge sources is not only easier but it also allows for much faster re-scoring as compared to N-best list re-scoring. To demonstrate the algorithm, we have carried out experiments with re-scoring by a higher order LM (acting as an external source of knowledge).

The rest of the paper is organized as follows: In the next section, we describe the basic principles of consensus decoding and expound some characteristics of a confusion network. In the next section titled 'Iterative Decoding Framework', we describe the details of the proposed algorithm. In the 'Experiments and Results' section, we present results supporting the speedy and efficient performance of Iterative Decoding. We conclude with a section where we talk about possible future directions.

II. Consensus Decoding

As mentioned in the previous section, consensus decoding aims at minimizing the word error rate of the recognition hypothesis. In this particular type of decoding, word lattices (obtained as the output of the first pass of the recognition process) are converted into confusion networks. A confusion network is a concatenation of confusion bins and each confusion bin contains word hypotheses. For every word w of the confusion network, its posterior probability given the acoustics i.e. $P(w|\mathbf{A})$ is computed by summing over the posterior probabilities of all of the links associated with that word.

The words in the confusion bin are ordered according to their posterior probability and for any bin these posterior probabilities sum to 1. Thus if a confusion network $\mathbf{CN}$ is comprised of n confusion bins $\{C_1, C_2, \ldots, C_n\}$, then we have the following identities: $\forall i \in \{1, 2, \ldots, n\}$, $\forall j_i \in \{1, 2, \ldots, |C_i|\}$, where $|C_i|$ is the size of the i^{th} confusion bin,

$$P(w_i(j_i)|\mathbf{A}) \geq P(w_i(j_i + 1)|\mathbf{A})$$

where $w_i(j_i)$ is the j_i^{th} word in the i^{th} confusion bin. $P(w|\mathbf{A})$ is the posterior probability of the word w given the acoustics $\mathbf{A}$.
Also: $\forall i \in \{1, 2, \ldots, n\}, \sum_{j_i=1}^{|C_i|} P(w_i(j_i)|\mathbf{A}) = 1$.

By way of its construction, a confusion network induces many paths which were originally not present in its corresponding input word lattice. Due to the inclusion of these new paths, it has been observed that the oracle word error rate of confusion networks is lower than that of lattices by 1% absolute.

Thus within the paradigm of confusion network decoding, the search for the oracle path can be achieved only after re-scoring the word hypothesis in each bin. Due to the linear ordering of the confusion bins, the global search done over the lattices for finding the least erroneous word string can now be replaced by local search over a small set of word hypotheses enabling the possibility of treating decoding as a multi-class classification problem. Previously researchers have carried out binary classification on a bin by bin basis (independent of the surrounding context) using transformation based learning formalism [6]. Since the surrounding context acts as a very strong clue it should be taken into account for any sort of classification problem.

III. Iterative Decoding Framework

In this section, we describe our algorithm, which will set up a 'framework' for re-scoring (i.e. on a bin by bin basis) by taking into account the surrounding context. Algorithm 1 describes the steps involved in Iterative Decoding.

Input: Confusion Network $\mathbf{CN} = \{C_1, C_2, \ldots, C_n\}$ s.t.
$\forall i \in \{1, \ldots, n\}, \forall j_i \in \{1, \ldots, |C_i|\}, P(w_i(j_i)|\mathbf{A}) \geq P(w_i(j_i + 1)|\mathbf{A})$
and $\sum_{j_i=1}^{|C_i|} P(w_i(j_i)|\mathbf{A}) = 1$
Output: Word String $\mathbf{W}^* = \{w_1^*, w_2^*, \ldots, w_n^*\}$ s.t.
$\forall i \in \{1, 2, \ldots, n\}, w_i^* \in C_i$

PrevHyp: NULL ;
CurrentHyp: Consensus decoding 1 Best ;

while *PrevHyp* $\neq$ *CurrentHyp and #Iter* $\leq$ *Threshold* **do**
  for $i = 1; i <= n; i++$ **do**
    $\forall l \in \{1, \ldots, n\}, l \neq i$, Pick $w_l(1)$;
    Re-score $w_i(1), w_i(2), \ldots, w_i(|C_i|)$ given the context $w_1(1), \ldots, w_{i-1}(1), w_{i+1}(1), \ldots, w_n(1)$;
    Re-rank $w_i(1), w_i(2), \ldots, w_i(|C_i|)$;
  end
  Set PrevHyp = CurrentHyp ;
  Set CurrentHyp = $w_1(1), \ldots, w_n(1)$;
  Set Iter $+= 1$;
end

Algorithm 1: Algorithm for Iterative Decoding

In each iteration, the algorithm starts with the first confusion bin for the purpose of re-scoring its word hypotheses. The top scoring word is extracted from all the other confusion bins. Using the resolved context surrounding the confusion-bin-to-be-decoded, knowledge sources are applied and the task of classification (multi-class) is achieved. For example, if the knowledge source to be applied is a higher order language model and the confusion-bin-to-be-decoded is located at the i^{th} position having size (cardinality) $|C_i|$, then one could compute the likelihood of $|C_i|$ word sequence hypotheses i.e. $\{w_1(1), \ldots, w_i(1), \ldots, w_n(1)\}$, $\{w_1(1), \ldots, w_i(2), \ldots, w_n(1)\}$ ... $\{w_1(1), \ldots, w_i(|C_i|), \ldots, w_n(1)\}$ by combining their new language model scores and acoustic model

scores. These hypotheses share same words in all positions except the i^{th} word position. Likelihood of the word sequence containing the j_i^{th} word ($j_i \in \{1, 2, \ldots, |C_i|\}$) can be a very fair approximation for the likelihood of $w_i(j_i)$ given the surrounding context i.e. $w_1(1), \ldots, w_{i-1}(1), w_{i+1}(1), \ldots, w_n(1)$ as long as this likelihood value is used for finding the relative scoring among the candidate word hypotheses i.e. $\{w_i(1), \ldots, w_i(|C_i|)\}$. Once the classification result is sorted according to the classification scores, the algorithm re-orders the word hypotheses in the confusion bin such that the word with lower index has higher score than the word with higher indices. The algorithm then moves to the next confusion bin and executes similar steps. Thus one by one all the confusion bins are processed and each confusion bin gets its re-ranked, re-ordered word hypotheses. The algorithm then loops over the above mentioned steps to re-do the classification on a bin by bin basis. The iterative process continues as long as the top scoring word sequence extracted from all the confusion bins (i.e. the word sequence formed by the concatenation of the top scoring word from each confusion bin), does not converge and the number of iterations are below a pre-determined threshold.

Since for each confusion bin, we perform the task of classification rather than decoding, it is interesting to note that we can make use of specialized classifiers depending upon the confusion type.

A. Iterative Decoding: A hill climbing formalism

Theorem 3.1: Let the word sequence $\mathbf{W}(0)$ be the original consensus decoded output. Under the re-scoring model (Λ), let the log likelihood of this word sequence be $\mathbf{LL}(0) = \log P(\mathbf{W}(0)|\Lambda)$. Then the likelihood of the output at each iteration of Iterative Decoding monotonically increases. Also, under the new model, the output of Iterative Decoding has a likelihood greater than or equal to the likelihood of the consensus decoding.

Proof: For the first iteration of Iterative Decoding, word sequence $\mathbf{W}(0)$ becomes the starting point. The output sequence $\mathbf{W_1}(1)$ (here subscript indexes the confusion bin and the bracketing indexes the iteration) obtained after decoding the first confusion bin in the first iteration, differs from $\mathbf{W}(0)$ in at-most one word location (first position). From the construction, the algorithm will pick a word in the first bin such that the resulting likelihood (under the re-scoring models Λ) increases over the starting point. Thus clearly

$$\log P(\mathbf{W}_1(1)|\Lambda) \geq \log P(\mathbf{W}(0)|\Lambda)$$

Now when the algorithm moves to second confusion bin, the starting point word sequence becomes $\mathbf{W}_1(1)$ and thus the decoded word in the second confusion bin will be such that the output word sequence will have greater or equal likelihood i.e.

$$\log P(\mathbf{W}_2(1)|\Lambda) \geq \log P(\mathbf{W}_1(1)|\Lambda)$$

Thus by induction we can say that for any two confusion bins indexed by j and $j-1$, we have:

$$\log P(\mathbf{W}_j(1)|\Lambda) \geq \log P(\mathbf{W}_{j-1}(1)|\Lambda)$$

and thus

$$\log P(\mathbf{W}_n(1)|\Lambda) \geq \log P(\mathbf{W}_1(1)|\Lambda)$$

When the algorithm enters second iteration, the output of the previous iteration becomes the starting point and by the construction of the algorithm we get

$$\log P(\mathbf{W}_1(2)|\Lambda) \geq \log P(\mathbf{W}_n(1)|\Lambda)$$

Again by easy induction we can say that the output at the end of two iteration indexed by i and $i-1$, we have:

$$\log P(\mathbf{W}_n(i)|\Lambda) \geq \log P(\mathbf{W}_n(i-1)|\Lambda)$$

and thus if we denote total number of iterations by I, then we get:

$$\log P(\mathbf{W}_n(I)|\Lambda) \geq \log P(\mathbf{W}(0)|\Lambda)$$

Thus, under the re-scoring model, the likelihood of the output of iterative decoding at each step increases monotonically and is greater than or equal to the likelihood of the consensus decoding. However, the algorithm is sub-optimal. ■

B. Analysis of Iterative Decoding

For any confusion network $\mathbf{CN}$ comprising n confusion bins $\{C_1, \ldots, C_n\}$, the total number of distinct word sequences is simply $\prod_{i=1}^{n} |C_i|$. Thus for any optimal N-best list re-scoring, these many hypotheses would have to be considered. As against this, the total number of distinct word sequence hypotheses extracted per iteration in the Iterative Decoding schema is: $\sum_{i=1}^{n} |C_i|(1 - \mathcal{I}(|C_i|, 1)$, where $\mathcal{I}(.,.)$ is an indicator function of two variables, which is 1 if they are equal, and 0 otherwise. Thus if the maximum number of iterations are set to I, then in the worst case, the total number of word sequence hypotheses extracted from a confusion network in Iterative Decoding becomes: $I \sum_{i=1}^{n} |C_i|(1 - \mathcal{I}(|C_i|, 1)$. Typically the number of bins with cardinality greater than 1 is 75% of the sentence length [4]. Thus for instance even if an input sentence of length, say 20, has 15 confusion bins with the least possible cardinality of 2, it can lead to as many as 2^{15} distinct word sequences. This indeed is a prohibitive number and in a real test scenario this number can grow exponentially with the size of the confusion bin and length of the network. As against that, the number of hypotheses considered for search in Iterative Decoding is a linear function of the size of the confusion bin and length of the network.

IV. Experiments and Results

A. Experimental Setup

For the purpose of our experiments, we have used IBM's state of the art speech recognizer: Attila [7]. The transcription system was trained on 330 hours of audio from the 1996 English Broadcast News Speech corpus (LDC97S44) [8] and the 1997 English Broadcast News Speech corpus (LDC98S71).

The acoustic features used by the system are PLP features. The system has acoustic models with roughly 6000 states and 250000 Gaussians. The size of the vocabulary is $90K$. More details on various aspects of the system can be found in [7].

We trained two language models from Broadcast News data. Hereafter we will refer to them as LM-BN-small and LM-BN-big. LM-BN-big is a Kneser Ney smoothed 4-gram language model trained on $163M$ words from 1996 CSR Hub4 language model data and Hub4 acoustic model training transcripts. This LM contains about $44M$ n-grams. LM-BN-small is a pruned tri-gram version of LM-BN-big and contains about $4M$ n-grams. LM-BN-small is used in the recognition first pass while the LM-BN-big is used for re-scoring.

IBM's ASR engine compiles a huge static graph by composing the acoustic phonetic tree and language model and is used during the first pass recognition. Due to the limitations posed by the computer memory, we limit the size of the LM used in the first pass decoding to about $4M$ n-grams.

From the acoustic data not used in the training system, we have used about 5 hours of speech for the test data and another 5 hours of speech as held out data. Both test and held out data corresponds to $2K$ utterances. Number of word tokens in test data is about $52K$.

We compare the performance (as a function of the size of the search space) of Iterative Decoding v/s N-best list re-scoring by finding the word error rate (WER) of the decoded hypotheses from each of the above mentioned methods. The decoding in both methods is carried out using the combination of baseline confusion network posterior scores and re-scoring language model scores. The scaling factors for the combination of the two scores are tuned such that the word error rate on the held out data is minimized. We have not incorporated any word insertion penalty factor in any of our experiments. Incorporation of acoustic score during re-scoring would have been more appropriate, however we wanted to compare the performances of both these methods with the dynamic programming method. In it, incorporation of acoustic scores is not possible, however log posterior scores and re-scoring log language model scores can be interpolated easily by way of composing the finite state machines of confusion network and language model.

Thus in Iterative Decoding, for each confusion bin to be decoded, we extract some hypotheses (equal to the size of the bin). For each of them we interpolate their log posterior scores and log language model scores and then re-rank them. This re-ranking decides the order of the word hypotheses in that confusion bin. Similarly, in N-best list method, we combine the log posterior score and log language model score and choose that hypothesis which gets the maximum score.

B. Results

In the case of Iterative Decoding, the average number of hypotheses generated for the entire test set is given as

$$N_{avg} = \frac{1}{T}\sum_{t=1}^{T} I_t \sum_{t_i=1}^{n_t} |C_{t_i}|(1 - \mathcal{I}(|C_{t_i}|, 1)$$

where, T is the total number of test speech utterances, n_t is the length of t^{th} confusion network, $|C_{t_i}|$ is the cardinality of the i^{th} bin of the t^{th} test confusion network, I_t is the number of iterations required for the convergence during the decoding of the t^{th} confusion network. While in the case of N-best list re-scoring, the average number of hypotheses is given by:

$$N_{avg} = \frac{1}{T}\sum_{t=1}^{T} \min(N, \prod_{t_i=1}^{n_t} |C_{t_i}|)$$

where, N is the maximum size of the N-best list.

We compare the performance of the two methods on the basis of the WER of the decoded output. Consensus decoding output (obtained from the first pass recognition) forms our baseline (first row of Table I). The average number of hypotheses considered during the search by N-best list method and Iterative Decoding method and their corresponding WER is reported (second and third row of Table I respectively). Finally, the minimum word error rate (found using finite state machine composition and least cost path extraction) is reported (last row of Table I). We used publicly available AT&T FSM toolkit [9] to carry out various finite state machine related operations. The maximum value of N in the N-best list re-scoring was kept to be 5000. This was the minimum value of N at which the WER obtained using N-best list re-scoring matched with the optimal WER. Increasing the value of N beyond 5000 did not change the WER.

TABLE I
MAIN RESULTS

Setup	$\mathbf{N_{avg}}$	**WER**
Baseline	-	22.38%
+ N best list re-scoring	4101	20.12%
+ Iterative Decoding	179	20.12%
+ FSM re-scoring	-	20.12%

Comparing the two tables we can see that, Iterative Decoding required on an average, a search space of just 179 utterances per confusion network by converging in as low as 4 iterations to achieve a word error rate of 20.12%. As against this, for a comparable performance, N-best list re-scoring required an average of 4101 utterances per confusion network. The search space required by the latter is at-least 22 times larger than that of the former. In the particular case of n-gram language model re-scoring, we also see that the word error rate obtained using our method matches with that by the FSM re-scoring method.

V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have presented an efficient framework to carry out re-scoring of confusion networks using sophisticated external knowledge sources. We demonstrated the algorithm using a language model as an external knowledge source. For a comparable performance, Iterative Decoding required 22 times smaller search effort than N-best list re-scoring, showing its efficiency. Although for this particular case of n-gram LM re-scoring, our method obtained the optimum results, it should

be noted that there is a possibility for the algorithm to get stuck at local maximum. However, as the starting point of the Iterative Decoding is the consensus decoded output, the search direction may hopefully lead to the optimum solution.

Typically the oracle path is often given a very low likelihood by the baseline models. Thus if one could come up with possibly the best re-scoring models, then the performance of the re-scoring system would implicitly depend upon its efficiency in searching the space of all possible hypotheses. In our method, the number of hypotheses considered during the search is a linear function of the size of the confusion bin and length of the network, while for N-best list, the number of hypothesis grows exponentially as the network becomes bigger and deeper.

As part of the future directions, we plan to use the Iterative Decoding framework for decoding confusion networks using sophisticated sentence level knowledge sources. We also want to evaluate an idea based on the entropy of the confusion bin, where we change the order in which confusion bins are decoded in each iteration of the Iterative Decoding. We are also trying to extend the use of our algorithm directly to the word lattices. Since word lattices retain acoustic information of word hypotheses, it would be mathematically more appropriate to incorporate acoustic scores during re-scoring, instead of the word posterior scores (as is done in confusion network re-scoring).

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their comments. This work was supported by National Science Foundation under grant No OISE-0530118.

REFERENCES

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience Publication, 2000.

[3] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in N-best list rescoring," in *Proc. Eurospeech '97*, Rhodes, Greece, 1997, pp. 163–166. [Online]. Available: citeseer.ist.psu.edu/stolcke97explicit.html

[4] L. L. Mangu, "Finding consensus in speech recognition," Ph.D. dissertation, 2000, adviser-Brill, Eric.

[5] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *Proc. Intl. Conf. on Spoken Language Processing (ICSLP), 2002.*, 2002.

[6] L. Mangu and M. Padmanabhan, "Error corrective mechanisms for speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 29–32.

[7] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 1596–1608, 2006.

[8] D. Graff, "The 1996 broadcast news speech and language-model corpus," in *Proc. 1997 DARPA Speech Recognition Workshop*, 1996, pp. 11–14.

[9] M. Mohri, F. Pereira, and M. Riley, "The design principles of a weighted finite-state transducer library," *Theoretical Computer Science, 231:17-32*, 2000.