LANGUAGE MODEL COMBINATION AND ADAPTATION USING WEIGHTED FINITE STATE TRANSDUCERS

X. Liu[1], *M.J.F. Gales*[1], *J.L. Hieronymus*[2], *P.C. Woodland*[1]

[1]Cambridge University Engineering Dept,
Trumpington St., Cambridge, CB2 1PZ U.K.
[2]NASA Ames Research Centre
Mountain View, CA 94035, USA
Email: {xl207,mjfg,jlh83,pcw}@eng.cam.ac.uk

ABSTRACT

In speech recognition systems language model (LMs) are often constructed by training and combining multiple n-gram models. They can be either used to represent different genres or tasks found in diverse text sources, or capture stochastic properties of different linguistic symbol sequences, for example, syllables and words. Unsupervised LM adaptation may also be used to further improve robustness to varying styles or tasks. When using these techniques, extensive software changes are often required. In this paper an alternative and more general approach based on weighted finite state transducers (WFSTs) is investigated for LM combination and adaptation. As it is entirely based on well-defined WFST operations, minimum change to decoding tools is needed. A wide range of LM combination configurations can be flexibly supported. An efficient on-the-fly WFST decoding algorithm is also proposed. Significant error rate gains of 7.3% relative were obtained on a state-of-the-art broadcast audio recognition task using a history dependently adapted multi-level LM modelling both syllable and word sequences.

1. INTRODUCTION

In current ASR systems language model (LMs) are often constructed by training n-gram components models [7] on data from a set of diverse sources representing different genre, epoch or other higher level attributes. In order to incorporate more linguistic constraints, it is also possible to train and combine LMs that model different unit sequences, for example, syllables and words [4]. Interpolated LMs with context free weighting are normally constructed using special purpose tools, for example, the SRILM toolkit [12]. In order to capture local variation of modelling resolution, generalization, topics and styles among component LMs, history context dependent LM interpolation and adaptation can be used [8]. These techniques often require extensive software changes. An alternative approach considered in this paper is to combine and adapt LMs using *semi-ring* based weighted finite state transducers (WFSTs) [9, 10]. Unless otherwise stated, *tropical semi-ring* based WFSTs are considered in this paper.

As this approach entirely based on well-defined WFST operations, minimum change to decoding tools is required. It is highly flexible and can be used for a wide range of combination configurations. It not only supports the use of global, context free weights in LM combination, but also a more general case when context dependent weights are employed. Thus LM adaptation using history context dependent interpolation can be conveniently implemented.

The rest of the paper is organised as follows. The use of WFST based LM representation in current ASR systems is reviewed in section 2. LM combination schemes using WFST operations are introduced in section 3. WFST based context dependent LM adaptation are presented in section 4. An efficient on-the-fly WFST decoding approach using context dependent LM adaptation is proposed in section 5. Experimental results on a state-of-the-art Mandarin broadcast speech transcription task are presented in section 6. Section 7 gives the conclusion and suggests possible future work.

2. WEIGHTED FINITE STATE TRANSDUCERS

A WFST is a finite state machine that associates weights such as probabilities, durations, penalties, or any other quantity that accumulates linearly along paths within a directed graph, to each pair of input and output symbol sequences. A set of classic finite automata operations to combine, optimize and compact WFSTs during search are available. Many types of modelling information used in speech recognition systems, such as HMM topology, lexicon and n-gram LMs, involve a stochastic finite-state mappings between symbol sequences. WFSTs provide a generic and well-defined framework to represent them. More precisely, n-gram LMs can be represented by weighted finite state acceptors (WFSA). These are special cases of WFSTs when the input and output symbol sequences are identical. Take two simple back-off 2-gram LMs for a three word vocabulary $\{A, B, C\}$ as examples, their WFST representations, $L_G^{(1)}$ and $L_G^{(2)}$, are shown in figure 1(a) and 1(b). In both transducers, n-gram log probabilities appear as the negated arc weights. The 1-gram back-off weights are represented by non-emitting epsilon arcs without output symbols, as marked with "<e>" in the figure.

3. LANGUAGE MODEL COMBINATION

Component LMs can be combined using linear, or log-linear model interpolation. In machine learning, they are commonly referred to as mixtures of experts (MoE) and products of experts (PoE) [5, 6].

Linear Model Combination: As a *union* of all the individual experts, it tends to to give a broader distribution than individual components alone. Hence, this form of model combination may help overcome the sparsity issue when training individual component models and thus improve generalization. Let w_i denote the i^{th} word of a

This work was in part supported by DARPA under the GALE program via a subcontract to BBN Technologies. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

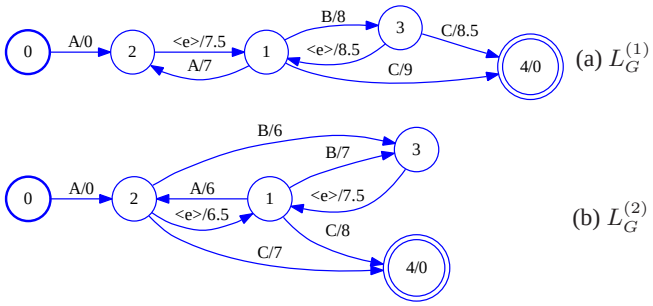

Fig. 1. WFST representation of two simple 2-gram back-off LMs.

L word long sequence $\mathcal{W} =< w_1, w_2, ..., w_i, ..., w_L >$. The LM log-probability for the complete word sequence is given by

$$\ln P(\mathcal{W}) = \sum_{i=1}^{L} \ln \left(\sum_{m=1}^{M} \lambda_m P_m(w_i|h_i^{n-1}) \right) \quad (1)$$

where h_i^{n-1} represents the i^{th} word's history of $n-1$ words maximum, $< w_{i-n+1}, ..., w_{i-1} >$, and λ_m is the global, context free weight for the m^{th} component under a positive and sum-to-one subject. These weights indicate the "usefulness" of each source for a particular task. To reduce the mismatch against the target domain, these weights may be perplexity tuned on held-out data.

Assuming component LMs model the same type of symbol sequences, for example, words, the WFST representation of the linearly combined LM can be derived using a component level *composition* between the n-gram and interpolation weight transducers prior to a final *log semi-ring* based WFST *union* operation. Hence,

$$L = \left(L_G^{(1)} \circ L_{\boldsymbol{\phi}}^{(1)} \right) \cup ... \left(L_G^{(m)} \circ L_{\boldsymbol{\phi}}^{(m)} \right) \cup ... \left(L_G^{(M)} \circ L_{\boldsymbol{\phi}}^{(M)} \right) \quad (2)$$

where $L_G^{(m)}$ is the n-gram model transducer, and $L_{\boldsymbol{\phi}}^{(m)}$ the interpolation weight transducer for the m^{th} component. Take the component LMs of figures 1(a) and 1(b) as examples, their context free interpolation weights, $\lambda_1 = 0.3, \lambda_2 = 0.7$ (1.220 and 0.360 as negated natural log) may be represented by the two transducers in figure 2. They proportionately reflect the probability contribution from the two component LMs on a context free basis, as the second model has more 2-grams than the first one. It can be shown that context free, global interpolation simply increases the costs of all emitting arcs within each component n-gram model sub-transducer by its own weight in the bottom of figure 2. The interpolated probability of any n-gram is represented by the marginalization over the probability of all partial paths in the union transducer that departs from a state representing context h_i^{n-1} and outputs word symbol w_i. In order to improve efficiency during search, a *closure* operation can be used to create a single terminal state before being further compressed via *determinization* and *minimization* operations.

It is also possible to linearly combine LMs modelling sequences of different linguistic units, for example, syllables and words. In order to have compatible transducer symbols during the union operation of equation (2), the syllable level component transducers must be first *composed* with a lexicon transducer, which provides subword to word mapping, and then *projected* onto word level.

Log-Linear Model Combination: In contrast, log-linear interpolation provides an *intersection* of individual experts. It yields a high likelihood only when all component models agree. For the example above, the log-linearly interpolated LM probability on word sequence level, with the probability normalization term ignored, is

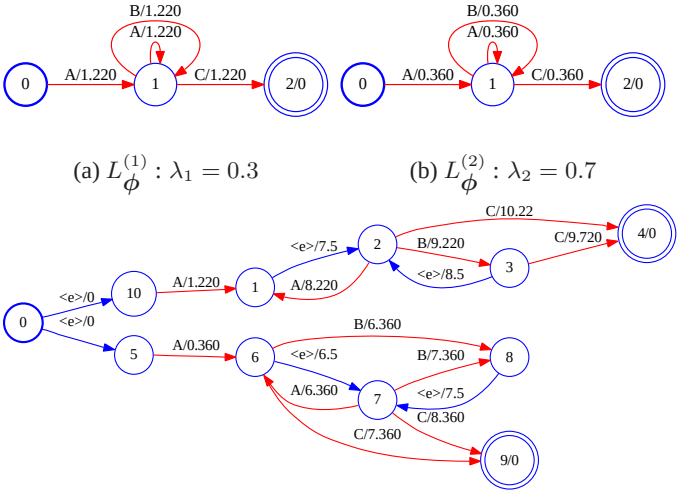

Fig. 2. The WFST representation of the context free linear interpolation weights for component LMs of figures 1(a) and 1(b) and the linearly interpolated LM derived using operations in equation (2).

$$\ln P(\mathcal{W}) = \sum_{i=1}^{L} \sum_{m=1}^{M} \lambda_m \ln P_m(w_i|h_i^{n-1}) \quad (3)$$

where λ_m is the context free log-linear weight for the m^{th} component. They are no longer subject to a positive and sum-to-one constraint. They may be optimized under a discriminative framework as in *maximum entropy* models [3]. In this paper these weights are fixed as equal. This form of model combination exploits the consensus among product experts. Hypotheses with very different log-likelihood ranking among component models will be penalized.

Assuming compatible symbols are used for all transducers, a log-linear model combination may be efficiently implemented using a sequence of WFST *composition* operations between component n-gram model transducers after an *arithmetic scaling* of arc costs by their respective log-linear weights. This is given by

$$L = \left(L_G^{(1)} \times \lambda_1 \right) \circ ... \left(L_G^{(m)} \times \lambda_2 \right) \circ ... \left(L_G^{(M)} \times \lambda_M \right) \quad (4)$$

The precise nature of component language models determines which of the two combination schemes is more appropriate. For example, when building word level LMs, in order to improve context coverage and generalization, a linear interpolation between component LMs trained over a diverse set of text sources can be used. When introducing additional sub-word level linguistic constraints to increase discrimination, word and syllable level LMs can be log-linearly combined [4]. In order to achieve a good balance between generalization and discrimination, it is also possible to leverage from both forms of combination using a product between mixtures of experts, or equivalently a composition between union-ed LMs.

4. LANGUAGE MODEL ADAPTATION

In order to improve robustness to varying styles or tasks, unsupervised test-time LM adaptation to a particular broadcast show, for

example, may be used. As directly adapting n-gram probabilities is impractical on limited amounts of data, standard adaptation schemes only involve updating the context free, linear interpolation weights of equation (1).

However, this approach can only adapt LMs to a particular genre, epoch or other higher level attributes. Local factors that determine the "usefulness" of sources on a context dependent basis, such as modelling resolution, generalization, topics and styles, are poorly modelled. Take 2-gram distribution $P(C|B)$ in figure 1 as an example, the first component LM of figure 1(a) gives a 2-gram log-probability of -8.5, while a lower score of -15.5 is assigned by the second one via a back-off path in figure 1(b). In this case the probability contribution from the two component LMs clearly contradicts the assignment of context free interpolation weights of 0.3 and 0.7 in figure 2. To handle this issue, context dependent LM interpolation and adaptation can be used [8]. A set of discrete context dependent back-off weights are used to dynamically adjust the contribution from component LMs. Thus equation (1) is extended to

$$\ln P(\mathcal{W}) = \sum_{i=1}^{L} \ln \left(\sum_{m=1}^{M} \phi_m(h_i^{n-1}) P_m(w_i|h_i^{n-1}) \right) \quad (5)$$

where $\phi_m(h_i^{n-1})$ is the m^{th} component weight for context h_i^{n-1}. Both maximum likelihood and discriminative schemes are available to robustly estimate context dependent interpolation weights [8].

The WFST representation of equation (2) also holds for LMs constructed using context dependent interpolation weights. The difference between context free and dependent LM interpolation in equations (1) and (5) lies in the precise nature of weight transducers. Again take the two component LMs of figures 1(a) and 1(b) as examples, the WFST representation of their context dependent interpolation weights are shown in figure 3(a) and 3(b). As is shown in the figure, when the history varies, more flexibility is allowed to component LM weighting than the context free case of figure 2. For 2-gram $P(C|B)$, a duly higher weight of 0.8 (0.219 as negated natural log) is now assigned to the first component LM.

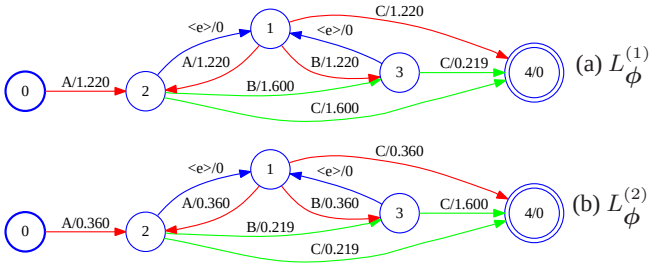

Fig. 3. WFST representation of context dependent linear interpolation weights for component LMs of figures 1(a) and 1(b).

5. ON-THE-FLY WFST NETWORK EXPANSION

When using context dependent interpolation in decoding, there is a flexible choice between a static, off-line application, and dynamic, on-the-fly application of the weights. During test-time LM adaptation, every broadcast show or snippet, for example, may have its own set of interpolation weights. When modelling a large number of contexts using the transducer topology of figure 3, the composition between component n-gram and their weight transducers can lead to a significant network expansion. This is highly inefficient and makes the subsequent network compression operations very expensive. A similar issue exists during the composition between component n-gram transducers in the log-linear combination of equation (4). Hence, it is preferable to dynamically perform the composition, union and compression operations of equation (2) in one single step on-the-fly. Related approaches have been previously shown effective for the composition between one single back-off n-gram LM and a lexicon [1, 2]. The basic idea to only create a new path on request during search, if and only if it carries context information different from others. For context dependent adaptation, the LM state associated with context history is *jointly* determined by component n-gram models and interpolation weights in the form of a context doublet. Using an on-the-fly lattice expansion algorithm, there are two advantages. First, under the lattice constraint, no dead-end state [1], which has no successful path to the terminal state, will be created during expansion. Secondly, redundant paths representing unused lower order back-off distributions will be automatically filtered out.

6. EXPERIMENTS AND RESULTS

The CU-HTK Mandarin ASR system was used to evaluate performance of multi-level combined and adapted LMs. The baseline system of word level recognition units was used in an initial "P2" lattice generation stage followed by a "P3" lattice rescoring stage using re-adapted acoustic models. The overall structure of the system was similar to that described in [11]. A 63k word list consisting a total of 52k multiple character Chinese words, 6k single character Chinese words and 5k frequent English words was used. An interpolated 4-gram word level baseline LM and adapted gender dependent cross-word triphone MPE acoustic models with HLDA projected PLP and pitch features trained on 1673 hours of broadcast speech were used in decoding. A total of 4.3 billion characters from 27 text sources were used in LM training. These account for 2.8 billion words after a longest first based character to word segmentation. Information on corpus size, cut-off settings, smoothing schemes and component weights for the top 10 heavily weighted text sources are given in table 1. Three GALE Mandarin broadcast speech development sets were used: 2.6 hour dev07, 1 hour dev08 and 2.6 hour p2ns. Manual audio segmentation was used. The word level baseline LM component weights were perplexity tuned on dev07, dev08 and additional held-out data.

Comp LM	#Char (M)	#Word (M)	Train Config	Intplt Weight
bcm	14.26	9.21	kn/111(11)	0.260058
bnm	12.29	7.41	kn/111(11)	0.147834
gigaxin	483.65	362.74	kn/112(22)	0.132539
phoenix	144.57	91.38	kn/112(22)	0.107920
gigacna	891.13	604.98	gt/123(33)	0.072665
voarfa	63.54	35.31	kn/112(22)	0.072299
ibmsina2	382.34	253.59	kn/112(22)	0.055601
bbndata	301.39	186.3	kn/112(22)	0.046213
galeweb	556.41	390.8	kn/122(22)	0.045918
agilece	336.78	204.5	kn/112(22)	0.031497

Table 1. Text source size, cut-off settings, smoothing scheme used and interpolation weights for top 10 heavily weighted text sources.

Confusion network (CN) decoding performance of the baseline word level LM at P2 stage is shown in the first line of table 2. In

order to incorporate additional sub-word level constraints in LMs, A character level LM was also constructed. This provided an indirect way of modeling syllable sequences, as syllable segmented and labelled Chinese texts are generally unavailable in large quantities. Due to data sparsity, only 6-gram character level LMs were built and linearly interpolated. Their cut-off settings are shown in bracket of table 1. On average the word based system produces approximately 1.5 characters per word. Hence, a 6-gram character level LM has a comparable context span to word level 4-gram LMs. CN performance of this system is shown in the second line of table 2. As expected, with a stronger constraint, the word level 4-gram baseline significantly outperformed the character 6-gram LM by 0.4%-1.2% absolute. When combining syllable and word constraints using an equal weighted log-linear interpolation of equation (3) and the WFST representation of equation (4), consistent performance improvements were obtained over the word level baseline. This is shown in the 3rd line of table 2. It gave statistically significant CER reductions of 0.5% and 0.3% on dev08 and p2ns respectively.

P2 System	LM Adapt	CER% dev07	dev08	p2ns
w.4g	-	9.7	9.6	9.6
c.6g	-	10.9	10.0	10.3
w.4g ∘ c.6g	-	9.5	9.1	9.3
w.4g	CI	9.6	9.3	9.4
w.4g	CD	9.5	9.2	9.3
w.4g ∘ c.6g	CD	9.4	8.9	9.1

Table 2. P2 CN performance of language models on dev07, dev08 and p2ns. "'∘" denotes the WFST composition operation.

The second section of table 2 shows performance of three adapted LMs using the WFST representation in equation (2). The 1-best outputs from the un-adapted word level baseline system was used as the supervision in perplexity based LM adaptation. Standard LM adaptation using context independent interpolation weights gave CER reductions of 0.1%-0.3% absolute across three test sets (4th line of table 2). Using three word history based context dependent adaptation of equation (5) with a hierarchical smoothing prior, further CER improvements of 0.1% absolute were obtained for all test sets (5th line of table 2). Adapting both word and character level LMs using context dependent weights before a final log-linear combination gave the best performance in the table. Absolute CER gains of 0.4% and 0.3% on dev08 and p2ns were obtained over the baseline word level LM adapted using context free interpolation. The total performance improvements over the unadapted word level baseline are 0.3% on dev07, 0.7% dev08 (7.3% rel) and 0.5% on p2ns (5.2% rel) respectively, all being statistically significant.

P3 System	LM Adapt	CER% dev07	dev08	p2ns
w.4g	-	9.3	8.7	9.1
w.4g	CI	9.1	8.6	9.1
w.4g	CD	9.0	8.5	8.8
w.4g ∘ c.6g	CD	8.8	8.4	8.6

Table 3. CN performance of P3 acoustic rescoring of P2 lattices generated by various language models on dev07, dev08 and p2ns.

Table 2 shows the performance of multi-level combined and adapted LMs at P2 lattice generation stage. Now it's interesting to examine if the performance improvements can be maintained at the P3 stage where re-adapted acoustic models are used to rescoring P2 lattices generated by various LMs in table 2. These are shown in table 3. Performance gains from the adapted multi-level combined LM (last line of table 3) over the word level baseline (first line of table 3) were largely maintained. Statistically significant CER reductions 0.3%-0.5% absolute were obtained over all test sets, in particular, 0.5% absolute (5.5% rel) for dev07 and p2ns.

7. CONCLUSION

Flexible LM combination and adaptation using weighted finite state transducers have been investigated in this paper. A wide range of model combination configuration are supported. An efficient on-the-fly WFST decoding algorithm was also proposed. Experimental results on a state-of-the-art large vocabulary speech recognition task suggest the proposed methods may be useful for speech recognition. Future research will focus on using efficient WFST representation for language models of more complicated forms. Transducer based acoustic model combination will also be considered.

8. REFERENCES

[1] D. A. Caseiro & I. Trancoso (2006). A Specialized On-the-fly Algorithm for Lexicon and Language Model Composition. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1281-1291, 2006.

[2] O. Cheng, J. Dines & M. M. Doss (2007). A Generalized Dynamic Composition Algorithm Of Weighted Finite State Transducers For Large Vocabulary Speech Recognition, in *Proc. ICASSP'07*.

[3] J. Darroch & D. Ratcliff (1972). "Generalized iterative scaling for log-linear models", Ann. Math. Statist., vol. 43, 1972.

[4] J. L. Hieronymus, X. Liu, M. J. F. Gales & P. C. Woodland (2009). Exploiting Chinese Character Models to Improve Speech Recognition Performance, in *Proc. Interspeech'09*.

[5] G. Hinton. Products of Experts, in *Proc. ICANN*, 1999.

[6] G. Hinton. Training Products of Experts by Minimizing Contrastive Divergence, *Neural Computation*, 14:1771–1800, 2002.

[7] S. M. Katz (1987). Estimation Of Probabilities From Sparse Data For The Language Model Component Of A Speech Recognizer. *IEEE Trans. ASSP* 35 (3), 400401.

[8] X. Liu, M. J. F. Gales & P. C. Woodland (2009). Use of Contexts in Language Model Interpolation and Adaptation, in *Proc. Interspeech'09*.

[9] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:2, 1997.

[10] M. Mohri & M. Riley. Network optimizations for large vocabulary speech recognition. *Speech Communication*, 25:3, 1998.

[11] R. Sinha, M. J. F. Gales, D. Y. Kim, X. A. Liu, K. C.Sim, and P. C. Woodland (2006). The CU-HTK Mandarin broadcast news transcription system, in *Proc. ICASSP'06*.

[12] A. Stolcke (2002). SRILM - An Extensible Language Modeling Toolkit, in *Proc. ICSLP'02*.