CRIM AND LIUM APPROACHES FOR MULTI-GENRE BROADCAST MEDIA TRANSCRIPTION

Vishwa Gupta[1], *Paul Deléglise*[2], *Gilles Boulianne*[1], *Yannick Estève*[2], *Sylvain Meignier*[2], *Anthony Rousseau*[2]

[1] Centre de recherche informatique de Montréal (CRIM)

{Vishwa.Gupta, gilles.boulianne}@crim.ca

[2] LIUM - University of Le Mans, France

firstname.lastname@univ-lemans.fr

ABSTRACT

The Multi-Genre Broadcast Challenge at ASRU 2015 is a controlled evaluation of speech recognition, speaker diarization, and lightly supervised alignment using BBC TV recordings. CRIM and LIUM teams participated in the speech recognition part of the challenge with a joint submission. This paper presents the CRIM and LIUM's contributions. Each team made different choices to develop its ASR system. By the way, it was expected to compare and to evaluate different approaches to diarization and acoustic modeling, and to get complementary ASR systems for effective merging. CRIM's main contributions are the use of a training scenario similar to multi-lingual training to estimate the deep neural net (DNN) acoustic models with most of the data, the use of a pruned trigram model for search, in addition to the use of a genre-dependent quadgram language model for rescoring the lattice from the search. For LIUM, the focus was on fast decoding with high accuracy. The final word error rates (WER) after merging show that it is possible to get reasonable WER with automatically aligned files. The final global WER of 25.1% corresponds to a WER reduction of about 20% absolute in comparison to the ASR baseline system provided by the organizers.

Index Terms— Deep Neural Networks, DNN, change point detection, automatic transcription, multi-genre broadcast transcription.

1. INTRODUCTION

The Multi-Genre Broadcast (MGB) Challenge at ASRU 2015 is a controlled evaluation of speech recognition, speaker diarization, and *lightly supervised* alignment using BBC TV recordings [1]. CRIM and LIUM participated in the speech recognition (or transcription) part of the challenge with a joint submission.

Even though the training data is strongly restricted, and even though main parts of ASR systems built by CRIM and LIUM teams are both based on the Kaldi toolkit [2], each team made different choices to develop its ASR system during this challenge. By the way, CRIM and LIUM were expected to compare and to evaluate different approaches and to produce complementary ASR systems for effective merging.

In acoustic training, CRIM's contribution is to use a training scenario similar to the multi-lingual training [3] to train the acoustic models with most of the data. Training data with more confidence is used to update all the weights during DNN training while the training data with less confidence updates only the weights associated with the hidden layers, and not the weights associated with the output layer. This DNN training strategy reduced the WER significantly. In language modeling, CRIM's contribution is to use a pruned trigram model for search, and to use a genre-dependent quadgram language model for rescoring the lattice from the search. For LIUM, very fast decoding with high accuracy was the *leitmotiv*.

While CRIM optimized their diarization system to minimize the overall diarization error rate (DER), LIUM minimized the false rejection of speech while optimizing the diarization system. LIUM's diarization strategy reduced the WER by 3% absolute. CRIM tried to use most of the acoustic training data while LIUM concentrated on using the high confidence acoustic data. Training with most of the data reduced the WER by 1.2% absolute. CRIM used genre-dependent quadgram language model while LIUM used continuous space language models (CSLM). All these differences effectively reduced the WER for the merged system.

2. ACOUSTIC TRAINING DATA SELECTION

The acoustic training data provided by MGB challenge committee contained lightly supervised alignments based on the transcripts from closed captioning. As a measure of confidence, they also computed phone matched error rates (PMER) and word matched error rates (WMER) [1]. CRIM and LIUM teams have applied different approaches for data selection of acoustic training data.

2.1. CRIM approach

CRIM used the word matched error rates to make the initial selection of the training data, and ran recognition experiments with different values of WMER to see which value gave the lowest word error rate (WER). These experiments used the default language model and the acoustic models (HMMs) as trained by the default Kaldi script provided by the organisers. The WER was computed using the text file. With this scoring style, the lowest WER was achieved with WMER of 50% (Table 1). From the resulting 760 hours of training audio, long word duration audio files - with average word duration (AWD) greater than 1 sec - were removed. This reduced WER by 0.25% absolute. We call the resulting training set A_{CRIM} (containing 747 hours of audio).

Table 1. *%WER with varying WMER percentage and approximate training audio in hours.*

% WMER	10	20	30	40	50	100
hours (approx)	240	400	530	640	760	1210
WER	57.4	57.2	57.1	57.0	56.9	57.3

From the other audio files with WMER greater than 50%, audio files with average word duration greater than 1 sec were removed to create another secondary training set from the remaining audio files. This set (with WMER greater than 50%) has approximately 323 hours of audio. This set was later used in DNN training to update weights of hidden layers, but not the weights associated with the output layer. This secondary training set in combination with set A_{CRIM} was able to reduce the WER on the development set. We call this secondary audio training set B_{CRIM}. Overall, DNNs were trained with 1070 hours of audio files (set A_{CRIM}+B_{CRIM}).

2.2. LIUM approach

LIUM investigated a different approach to extract relevant audio/text alignments to train acoustic models. First, ASR outputs and pronunciation dictionary provided by the organizers were used to train DNN acoustic models. Then, all the audio files provided by the organizers as part of the training corpus were processed by using an internal tool for speaker diarization [10]. Each produced speech segment was transcribed by using the first DNN acoustic models, combined with a 2-gram language model presented in section 3.2. This processing generated a word-graph for each speech segment. Each word-graph was aligned with subtitles made by human annotators and provided with the audio files. Word-graph alignment consists of searching a path within the word-graph that matches with the subtitles, accepting that rough timecode values from subtitles and precise timecode values within the word-graph could be delayed by 20 seconds max. Only long speech segments with no more than one word mismatch (insertion, substitution, or deletion) between subtitles and the closest path in the word-graph were selected. The text associated with a selected speech segment is the one coming from the closest path in the word-graph in regards with the subtitles. The training alignments generated by LIUM result in 700 hours of training audio. We call this set A_{LIUM}.

3. LANGUAGE MODELS

3.1. CRIM language models

Language models were trained on provided, normalized BBC subtitles representing 646M word tokens. The normalization is described in [11]. The hand-transcribed development set from the transcription task was used for interpolation weight tuning and perplexity evaluation. The development set contained 229K word tokens after removing comments, vocal noises and post-processing acronyms. First, trigram and quadgram language models were trained on all this data, with modified Kneser-Ney smoothing, and limiting the vocabulary to the 150,000 word set provided by the MGB Challenge organisers. Their respective perplexities were 126 and 118 on the development set. The quadgram LM was slightly pruned to 21 M 3-grams and 19 M 4-grams. The trigram was more heavily pruned to 2.4 M 3-grams and 3.0 M 2-grams, somewhat larger than the default model of 1.0 M 3-grams, 1.6 M 2-grams.

Since genre labels were common to training, development and evaluation shows, we generated separate language models for each genre. These genre dependent LMs were then used to recognize each show. Using genre labels from the metadata for each show, we split training and development texts into 8 genres (first column of table 2) and trained genre-dependent trigram models. The per genre perplexity for genre-independent trigram and quadgram appears in columns 2 and 3, and the genre-dependent trigram perplexity in column 4. Even though genre-dependent trigrams had higher perplexity, we observed a significant reduction in perplexity when interpolating them with genre-independent LMs (last two columns of table 2).

Table 2. *Development set perplexities for genre-independent trigram, quadgram, genre-dependent trigram and interpolated models.*

Genre	G.I. 3-g	G.I. 4-g	G.D. 3-g	Interp. 3-g	Interp. 4-g
advice	100.3	94.1	122.2	91.8	86.6
childrens	122.4	115.5	135.0	103.2	97.5
comedy	109.9	102.9	149.7	103.7	97.0
competition	120.4	110.9	145.4	105.9	98.1
documentary	138.9	131.4	195.2	132.0	124.8
drama	89.1	81.2	119.9	84.5	77.2
events	147.0	140.2	175.9	121.7	116.0
news	178.6	166.5	179.1	131.6	124.6

3.2. LIUM language models

The LIUM ASR system involved in the MGB challenge uses 2-gram, 3-gram, 4-gram back-off LMs, and a 5-gram back-off used in combination with a 5-gram feed forward neural network model. Back-off LMs were estimated through the SRILM toolkit, while the neural network language model (NNLM) was estimated by using the CSLM toolkit, developed at LIUM and distributed under LGPL license [12]. All these language models created by the LIUM were estimated on the entire normalized data provided by the organizers. No LM adaptation was applied.

LIUM's vocabulary contains 152K words, the most frequent ones in the normalized training data. Classical back-off n-gram models were trained by using the modified Kneser-Ney smoothing, without cutoff nor pruning. The 5-gram LM is composed of 152K 1-grams, 25M 2-grams, 125M 3-grams, 254M 4-grams, and 330M 5-grams. The 5-gram NNLM is composed of a projection layer of 640 units, corresponding to 160-dimensional word embeddings, two hidden layers of 1024 units each, and an output layer providing probabilities for a short-list composed of the 16384 most frequent words.

The use of the LMs is presented in section 4.2.

4. ACOUSTIC MODELS AND SINGLE DECODING

4.1. CRIM acoustic models and single decoding processes

In order to train the best possible acoustic models, the CRIM team tried two different feature parameters and many different deep neural networks (DNNs): TRAP features [4] and cepstral features transformed by an fMLLR transform per speaker. The TRAP features gave significantly lower WER than the fMLLR transformed cepstral features. With TRAP features, the best WER was achieved with training sets A_{CRIM}+B_{CRIM} and was 29.5% (without i-vectors), while the fMLLR transformed cepstral features gave 36.9% WER on the same development set (dev.full + dev.longitudinal). Part of the reason for this big difference in WER may be because the training set diarization was not based on manual segmentation of the audio into speaker turns, but by automatic speaker diarization using Cambridge's RT04 speaker diarization system [5].

To compute TRAP features, the 40-dimensional filterbank features are normalized to zero mean per audio file. 31 frames of these 40-dimensional filterbank features (15 frames on each side of the current frame) are spliced together to form a 1240-dimensional feature vector. This 1240-dimensional feature vector is transformed using a Hamming window (to emphasize the center), passed through a discrete cosine transform and the dimensionality reduced to 40×16 or 640-dimensional feature vector per frame.

Two different DNN activations were tried: p-norm versus sigmoid activation in the hidden layers of the DNN [6]. DNN's with p-norm activation gave lower WER and were kept for the rest of the experiments. DNN's with p-norm activation had 5 hidden layers, each hidden layer had p-norm input dimension of 2500 and p-norm output dimension of 500. The output softmax layer had 3276 outputs. DNN's were kept small in order to be able to train multiple DNNs in the small time frame available for this challenge.

Initially only set A_{CRIM} (747 hours of audio) was used for training the DNNs, with and without i-vectors [7][8][9]. There were 100 i-vectors per speaker, so the input feature vector dimension increased from 640 to 740 for DNNs with i-vectors. The Kaldi toolkit computed the i-vectors for both the training and the development sets. We initially ran experiments with the diarization for the development set using the automatically aligned development set files provided by the organizers. The comparative results after rescoring with unpruned trigram LM are shown in Table 3. We reduce the WER by 0.5% absolute with the i-vectors.

Table 3. *WER on the development set (dev.full) for DNNs with/without i-vector input.*

TRAP features	34.3%
TRAP + i-vectors	33.8%

We achieved our lowest WER on the development set by training DNNs on both set A_{CRIM} and set B_{CRIM} (1070 hours of training audio in total). In this case, we use a multi-lingual style training [3], where we train a DNN with p-norm activation and two output layers. The second output layer is duplicated from the first output layer. The set A_{CRIM} updates weights for the hidden layers and for the first output layer, while the set B_{CRIM} updates weights for the hidden layers and the 2nd output layer. The final model contains the hidden layers and the first output layer only. So in effect, set B_{CRIM} only updates the weights in the hidden layers of this final model. This strategy worked well and gave DNNs with significantly lower WER on the dev set. The overall reduction in WER was 1.2% with multi-lingual style training. We trained 4 different DNNs using the above training strategy:

1. using TRAP features, on set A_{CRIM} + B_{CRIM}
2. using TRAP features + i-vectors, on set A_{CRIM} + B_{CRIM}
3. using TRAP features + i-vectors, on set A_{LIUM} + B_{CRIM}
4. using fMLLR transformed cepstral features, on set A_{CRIM} + B_{CRIM}

For the multi-lingual style training, we start with fully MMI trained models using set A, and then we carry on 5 more iterations using back propagation with multi-lingual style training. This is followed by discriminative multi-lingual style training. The results are shown in table 4.

Note that in this table, the dev set includes both dev.full and dev.longitudinal, and that the diarization is done by LIUM. We used LIUM's diarization [10] as it gave roughly 3% lower WER than using CRIM's diarization [13]. The major difference seems to be that LIUM optimized their diarization to give 0.45% rejection of audio containing speech, while CRIM's diarization rejected 7.45% of audio containing speech. In table 4 we used genre dependent trigram LMs for rescoring (column 5 in table 2).

Table 4. *WER on the Dev set (dev.full + dev.longitudinal) with LIUM diarization and different DNNs with multi-lingual style training.*

Features/training set	TRAP	TRAP+i-vec	Cepstral
A_{CRIM}+B_{CRIM}	29.5%	28.6%	36.9%
A_{LIUM}+B_{CRIM}		29.6%	

We also tried CRIM's diarization for the development set. CRIM's diarization was optimized to give minimum *false alarm+false rejection* of speech, which resulted in roughly 7.45% false rejection of speech. The results for CRIM's diarization are shown in Table 5.

Table 5. *WER on the Dev set with CRIM's diarization with different DNNs with multi-lingual style training.*

Features/training set	TRAP	TRAP+i-vec	Cepstral
A_{CRIM}+B_{CRIM}	32.8%	31.9%	38.4%
A_{LIUM}+B_{CRIM}		32.3%	

In our final decoding, we used genre-dependent quadgram LMs for rescoring (last column in table 2). The genre-dependent quadgram LMs reduced the WER by 0.5% absolute compared to the genre-dependent trigram LMs. The results with the different DNNs are shown in table 6 for LIUM's diarization. The results with CRIM's diarization are shown in table 7. The ctm files and lattices generated by these rescored outputs were then used for final combination with ROVER.

Table 6. *WER on the Dev set (LIUM diarization) after rescoring with genre-dependent quadgram LM and different DNNs with multi-lingual style training.*

Features/training set	TRAP	TRAP+i-vec
A_{CRIM}+B_{CRIM}	29.0%	28.0%
A_{LIUM}+B_{CRIM}		29.3%

4.2. LIUM acoustic models and single decoding processes

Acoustic models estimated by LIUM were based on DNN, and close to the TRAP-based ones built by CRIM. But, some differences are noticeable. For each frame, DNN inputs were composed of 368 TRAP coefficients computed on a sliding window of 31 frames. Each frame was constituted by the outputs of 23 Mel-scale filterbanks. Speaker adaptation was trivial: it only consists on mean subtraction applied on all frames associated to a speaker. The DNN was built following the approach described in [14] and it was composed of six hidden layers with 2048 units, while the output softmax layer had 4627 outputs. It has been trained on the set A_{LIUM}.

Table 7. *WER on the Dev set (CRIM diarization) after rescoring with genre-dependent quadgram LM and different DNNs with multi-lingual style training.*

Features/training set	TRAP	TRAP+i-vec
A_{CRIM}+B_{CRIM}	32.4%	31.3%
A_{LIUM}+B_{CRIM}		31.9%

The LIUM ASR system was a multi-pass system. It was based on the Kaldi system for acoustic decoding and on LIUM tools built from the CMU Sphinx project for linguistic rescoring [15], with some modifications and additional features in order to accelerate the decoding processing and to reach a good accuracy performance. The first pass produces word-graphs by using the DNN acoustic models described above combined with the 2-gram LM presented in section 3.2. Next passes consisted on expanding and rescoring the word-graphs by using 3-gram, then 4-gram back-off LMs, then the 5-gram NNLM (including the 5-gram back-off LM). Last, an accelerated version of the consensus approach [16], which takes into account temporal information to speed up the processing, is applied on the confusion networks built from the 5-gram rescored word-graphs. The results for each pass are summarized in table 8. These experiments were made on a single machine, equipped with one Intel Xeon E5-2690 v2 CPU (10 cores with multi-threading), 128Go RAM, and one NVidia Tesla K40 GPU card.

Table 8. *WER and computation time (in Real Time) on the Dev set (dev.full+dev.longitudinal) of the system which has participated to the MGB Challenge.*

Step	Comment	WER	Comput. time
1	DNN + 2-gram	-	0.065 x RT
2	3-gram rescoring	31.4%	0.0015 x RT
3	4-gram rescoring	30.4%	0.002 x RT
4	CSLM 5-gram rescoring	29.6%	0.1 x RT
5	consensus	29.4%	0.001 x RT
Total	Full process (official submission)	29.4%	0.17 x RT

As shown in table 8, it is possible to get competitive performances with high speed processing, equal to 0.17 x Real Time. While it uses efficiently the GPU power, the CSLM rescoring process slows down the full process. Ta-

ble 9 presents the performances of alternative systems, without CSLM rescoring. It shows that it is possible to get a twice faster system by losing 0.8 point of word error rate. In the framework of an evaluation campaign, we decided to keep the CSLM rescoring.

Table 9. *WER and computation time (in Real Time) on the Dev set of alternative systems in comparison to the one that has participated to the MGB Challenge.*

Comment	WER	Computation time
Full process with 5g CSLM	29.4%	0.17 x RT
Full with classical 5g LM	30.3%	0.075 x RT
Full with classical 4g LM (without 5g LM)	30.2%	0.072 x RT

5. MERGING

The LIUM ASR system was applied twice by using CRIM diarization (32.7% WER) and LIUM diarization (29.4% WER). These two ASR outputs are called, respectively, cslmSegLium and cslmSegCrim.

The initial fusion of the recognition outputs was done by LIUM, with intelligent ROVER on lattices generated by CRIM and LIUM based on a common diarization, either CRIM's or LIUM's. This merging approach was described in [17]. It produced the following results:

1. fusionSegCrim: fusion of LIUM and CRIM lattices, CRIM diarization (30.1% WER)

2. fusionSegLium: fusion of LIUM and CRIM lattices, LIUM diarization (25.8% WER)

The ctm files produced by LIUM were then combined with ctm files produced by CRIM to get the final result. The best result is obtained with a ROVER of 3 ctm files from LIUM and 5 ctm files from CRIM. The final result is close to 25.1% WER for the Dev set. The eight files were rovered in the following order:

1. fusionSegLium: (25.8% WER)

2. CRIM: TRAP features with i-vectors, set A_{CRIM}+B_{CRIM}, LIUM diarization (28.0% WER)

3. CRIM: TRAP features, set A_{CRIM}+B_{CRIM}, LIUM diarization (29.0% WER)

4. CRIM: TRAP features with i-vectors, set A_{LIUM}+B_{CRIM}, LIUM diarization (29.3% WER)

5. CRIM: TRAP features with i-vectors, set A_{CRIM}+B_{CRIM}, CRIM diarization (31.3% WER)

6. CRIM: TRAP features with i-vectors, set A_{LIUM}+B_{CRIM}, CRIM diarization (31.9% WER)

7. cslmSegLium: 29.4% WER

8. fusionSegCrim: 30.1% WER

So through ROVER, WER is reduced by approximately 3% absolute (from 28% to 25.1%). For the Eval set, we did ROVER in the same order and submitted the resulting ctm file as CRIM-LIUM primary system. The fusionSegLium system on the Eval set was submitted as the contrasting system for CRIM-LIUM submission. The final results are shown in table 10. The 26.8% WER on the evaluation set was good enough for *2nd rank* in the MGB challenge transcription evaluation (out of 12 participants). The lowest WER on the Eval set during evaluation was 23.9% [1] .

Table 10. *Final WER on the development (dev.full + dev.longitudinal) and evaluation sets.*

System	Dev	Eval
single (LIUM ASR)	29.4%	30.7%
single (TRAP+i-vect)	28.0%	29.5%
fusion (LIUM seg)	25.8%	27.3%
fusion + rover	25.1%	26.8%

6. CONCLUSION

In the final analysis, we see that merging outputs of two heterogeneous systems results in a significant reduction in WER from 28% for the best single system to 25.1% for the merged system. The LIUM system gives 29.4% WER at 0.17 times real-time. The CRIM system was optimized for performance and showed that we can effectively use acoustic transcription with low confidence in a multi-lingual style DNN training scenario. LIUM's diarization strategy with minimal false rejection of speech resulted in a 3% absolute reduction in WER.

The final WER of 25.1% shows that it is possible to get reasonable WER with automatically aligned files. The WER of 25.1% is probably twice that of what we might expect to achieve with manually transcribed audio. However, transcribing 1600 hours of audio would be a challenging task and require a minimum of 16000 hours of manual transcription.

7. REFERENCES

[1] P. Bell et. al., "The MGB challenge: Evaluating multi-genre broadcast media transcription", in Proc. ASRU 2015.

[2] D. Povey et. al., "The Kaldi Speech Recognition Toolkit", in Proc. ASRU 2011.

[3] N. Vu, D. Imseng, D. Povey, P. Motlicek, T. Schultz, H. Bourlard, “Multilingual deep neural network based acoustic modeling for rapid language adaptation”, in Proc. ICASSP 2014, pp. 7689–7693.

[4] F. Grézl, “TRAP-based Probabilistic Features for Automatic Speech Recognition”, Doctoral Thesis, dept. Computer Graphics & Multimedia, Brno Univ of Technology, Brno 2007.

[5] S. Tranter, M. Gales, R. Sinha, S. Umesh, P. Woodland, “The development of the Cambridge University RT-04 diarisation system,” in Proc. Fall 2004 Rich Transcription Workshop (RT-04), 2004.

[6] X. Zhang, J. Trmal, D. Povey, S. Khudanpur, “Improving deep neural network acoustic models using generalized maxout networks”, in Proc. ICASSP 2014, pp. 215–219.

[7] V. Gupta, P. Kenny, P. Ouellet, T. Stafylakis, “I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription”, in Proc. ICASSP 2014, Florence, Italy.

[8] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors”, in Proc. ASRU 2013, pp. 55-59.

[9] A. Senior, I. Moreno, “Improving DNN speaker independence with i-vector inputs”, in Proc. ICASSP 2014.

[10] S. Meignier and T. Merlin, “LIUM SPKDIARIZATION: An open source toolkit for diarization”, in CMU SPUD workshop, Dallas, Tx, 2010.

[11] P.J. Bell, F. McInnes, S. Gangireddy, M. Sinclair, A. Birch, and S. Renals, “The UEDIN english ASR system for the IWSLT 2013 evaluation”, in Proc. International Workshop on Spoken Language Translation, 2013.

[12] H. Schwenk, “CSLM - A modular Open-Source Continuous Space Language Modeling Toolkit”, in Proc. Interspeech 2013, Lyon, France.

[13] V. Gupta, G. Boulianne, P. Kenny, P. Ouellet, and P. Dumouchel, “Speaker Diarization of French Broadcast News”, in Proc. ICASSP 2008, pp. 4365–4368.

[14] K. Veselý, A. Ghoshal, L. Burget, D. Povey, “Sequence-discriminative Training of Deep Neural Networks”, in Proc. Interspeech 2013, Lyon, France.

[15] P. Deléglise, Y. Estève, S. Meignier, T. Merlin, “Improvements to the LIUM French ASR system based on CMU Sphinx: what helps to significantly reduce the word error rate?”, in Proc. Interspeech 2009, Brighton, UK

[16] L. Mangu, E. Brilland, A. Stolcke, “Finding Consensus in Speech Recognition: Word Error Minimization and other Applications of Confusion Networks”, in Computer Speech and Language, vol. 14, number 4, pp 373-400, 2000.

[17] A. Rousseau, G. Boulianne, P. Deléglise, Y. Estève, V. Gupta, S. Meignier, “LIUM and CRIM ASR System Combination for the REPERE Evaluation Campaign”, in Proc. Text, Speech and Dialogue, Brno, Czech Republic, 2014