The LIUM ASR and SLT Systems for IWSLT 2015

Mercedes García-Martínez, Loïc Barrault, Anthony Rousseau,
Paul Deléglise, Yannick Estève

LIUM, University of Le Mans, France
firstname.lastname@lium.univ-lemans.fr

Abstract

This paper describes the Automatic Speech Recognition and Spoken Language Translation systems developed by the LIUM for the IWSLT 2015 evaluation campaign. We participated in two of the proposed tasks, namely the Automatic Speech Recognition task (ASR) in German and the English to French Spoken Language Translation task (SLT). We present the approaches and specificities found in our systems, as well as our results from the evaluation campaign.

1. Introduction

This paper describes the ASR and SLT systems developed by the LIUM for the IWSLT 2015 evaluation campaign. We participated in the two tasks mentioned above, with German language for the ASR task; and English to French for the SLT task.

The remainder of this paper is structured as follows: in section 2.1, we describe the data used for both tasks and how the selection was performed. In section 2, we present the architecture of our ASR system and the results obtained on the various corpora used during the campaign. Then in section 3, we expose the architecture of our SLT system, along with its results during the campaign. Lastly, the section 4 concludes this system description paper.

2. Automatic Speech Recognition Task in English

In this section, we will describe the Automatic Speech Recognition system developed by the LIUM for the IWSLT 2015 campaign, as well as present the results (both in-house and official) obtained on various corpora.

2.1. Data selection for the ASR task

Performance of Natural Language Processing (NLP) systems like the ones we are going to present here can often be enhanced using various methods, which can occur before, during or after the actual system processing. Among these, one of the most efficient pre-processing method is data selection, *i.e.* the fact to determine which data will be injected into the system we are going to build. For this campaign, many data selection processing was done, both in ASR and SLT tasks.

2.1.1. Data selection for acoustic models training

For our acoustic modeling we used as a primary source the Euronews ASR 2013 dataset [1] provided by the campaign organizers. In order to strengthen this base, we added data from various in-house sources. Then, we also collected a set of TEDx talks in German and carefully removed the off-limit talks. The Table 1 summarizes the characteristics of the data included in our ASR system acoustic models.

Corpus	Duration	Segments	Words
Euronews	69.1h	22 707	506 019
In-house	207.2h	42 316	2 018 262
TEDx	38.0h	42 633	312 142
Total	314.3h	107 656	2 836 423

Table 1: Characteristics of the acoustic data used in the LIUM ASR system acoustic models.

2.1.2. Data selection for language models training

Since language models training data is constrained for the ASR task, we applied our data selection tool XenC [2] on each allowed corpus at our disposal: basically all of publicly available WMT15 data and a collection of TEDx Talks closed-captions. Based on cross-entropy difference from a corpus considered as in-domain and out-of-domain data, our tool allows to perform relevant data selection by scoring out-of-domain sentences regarding their closeness to the in-domain data. The table 2 summarizes the characteristics of the monolingual text data used to estimate our system language models.

2.2. Architecture of the LIUM ASR system

Our architecture is based on two separate systems, mainly based on the Kaldi open-source speech recognition toolkit [3] which uses finite state transducers (FSTs) for decoding. A first pass is performed by using a bigram language model and deep neural network acoustic models. This pass generates word-lattices: an in-house tool, derived from a rescoring tool included in the CMU Sphinx project, is used to rescore word-lattices with a 3-gram, then a 4-gram back-off LM and

Corpus	Original # of words	Selected # of words	% of Orig.
IWSLT14	2.85M	2.85M	100.00
Common Crawl	48.04M	4.24M	8.82
Europarl	47.40M	3.20M	6.74
News Crawl	1.4G	130.60M	9.26
News-Comm.	5.06M	0.62M	12.25
Total (w/o IWSLT14)	1.5G	138.66M	9.18

Table 2: Characteristics of the monolingual text data used in the LIUM ASR system language models.

5-gram Continuous Space Language Model [4]. Last, an accelerated version of the consensus approach [5], 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.

2.2.1. Acoustic modeling

The GMM-HMM (Gaussian Mixture Model - Hidden Markov Model) models are trained on 13-dimensions PLP features with first and second derivatives by frame. By concatenating the four previous frames and the four next frames, this corresponds to $39 * 9 = 351$ features projected to 40 dimensions with linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT). Speaker adaptive training (SAT) is performed using feature-space maximum likelihood linear regression (fMLLR) transforms. Using these features, the models are trained on the full 314.3 hours set, with 9 500 tied triphone states and 325 000 gaussians.

On top of these models, we train two separate deep neural networks (DNNs). The first one is based on TRAP features: 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. It has been trained on the full 314.3 hours set. The DNN was built following the approach described in [6] and it was composed of 6 hidden layers with 2048 units, while the output softmax layer had 4627 outputs. The second one is based on the same fMLLR transforms as the GMM-HMM models and on state-level minimum Bayes risk (sMBR) as discriminative criterion. Again we use the full 314.3 hours set as the training material. The resulting network is composed of 6 hidden layers with 2 048 units, while the output dimension is 7 827 units and the input dimension is 440, which corresponds to an 11 frames window with 40 LDA parameters each.

To speed up the learning process, each DNN is trained using general-purpose graphics processing units (GPGPU) and the CUDA toolkit for computations.

2.2.2. Language modeling

For language modeling, we rely on the SRILM language modeling toolkit [7] as well as the Continuous Space Language Model toolkit. The vocabulary used in the LIUM ASR systems is composed of 131 435 entries. The language models are trained on the data presented in section 2.1.2 and separate sets are trained for each system.

With the SRILM toolkit, a 2-gram LM is estimated for each corpus source, with no cut-offs and the modified Kneser-Ney discounting method. These 2-gram LM are then linearly interpolated to produce the final 2-gram LM which will be used in the final system, using the German IWSLT 2013 test corpora. To rescore the word-lattices produced by Kaldi, a trigram and a quadrigram language models (namely 3G and 4G) are estimated with the same toolkit, again by training one LM by corpus source and then linearly interpolating them. A 5G continuous-space language model (CSLM) is also estimated for the final lattice rescoring, with no cut-offs and the same discounting method as for the bigram language model. Table 3 and table 4 details the interpolation coefficients for the 2G, 3G and 4G language models as well as the final perplexity for each final model used in the two systems, respectively for the TRAP-based and the fMLLR-based system.

Corpus	Coefficients		
	2G	3G	4G
manual transcriptions of speech	0.21	0.16	0.16
Common Crawl	0.03	0.05	0.05
News Crawl	0.21	0.18	0.17
Europarl	0.04	0.06	0.07
News-Comm.	0.51	0.55	0.0.55
Perplexity	379	279	264

Table 3: Interpolation coefficients and perplexities for the bigram (2G), trigram (3G) and quadrigram (4G) language models used in the LIUM ASR TRAP-based system.

2.3. Word-lattice merging

Both systems used the same audio segmentation, provided by the LIUMSpkDiarization[8] speaker diarization toolkit. Using the same segmentation makes easier the merging between the two ASR outputs: final outputs were obtained by merging word-lattices provided by both ASR systems.

Both systems provide classical word-lattices with usual information: words, temporal information, acoustic and linguistic scores. Before merging lattices, for each edge, these scores are replaced by its *a posteriori* probability. Posteriors are computed for each lattice independently, then weighted

Corpus	Coefficients		
	2G	3G	4G
IWSLT14	0.016	0.014	0.012
Common Crawl	0.028	0.023	0.020
Europarl	0.075	0.090	0.097
News Crawl	0.872	0.866	0.865
News-Comm.	0.008	0.008	0.006
Perplexity	514	349	326

Table 4: Interpolation coefficients and perplexities for the bigram (2G), trigram (3G) and quadrigram (4G) language models used in the LIUM ASR fMLLR-based system.

by $\frac{1}{n}$, where n is the number of word-lattices to be merged (here, $n = 2$). In our experiments, we did not find significant improvements by using more tuned weights.

For each speech segment, the use of weighted posteriors allows to merge starting (respectively ending) nodes from both lattices together into a single lattice in order to process directly with an optimized version of the consensus network confusion algorithm. This optimization reduces very significantly the computation time by managing temporal information during the clustering steps.

2.4. Results

The LIUM ASR system officialy achieved a Word Error Rate score of 17.8 on the 2015 test set, however, at this time of writing, ranks for each participant and full results have not been disclosed, thus we are not able to provide comparisons.

3. Spoken Language Translation Task

In this section, the architecture of our Statistical Machine Translation (SMT) system used in the SLT task is described.

3.1. Architecture of the LIUM SLT system

The SMT system is based on the Moses toolkit [9]. The standard 14 feature functions were used (*i.e.* phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, word and phrase penalty and target language model (LM) probability). In addition to these, a 5-gram Operation Sequence Model (OSM) [10] have been trained and included in the system.

3.2. Data processing and selection for the SLT task

All available corpora have been used to train the different components of the SMT system. The source side of the bitexts have been processed in order to make it more similar to speech transcriptions. After a standard tokenization, the processing mainly consisted in applying regular expressions to delete punctuations and unwanted characters, convert capital letters in lowercase and rewrite numbers in letters.

Once the processing performed, monolingual and bilingual data selection has been applied using XenC [2]. For this purpose, the TED corpus has been used as in-domain corpus (to compute in-domain cross-entropy). The development corpus (named *liumdev15*) was used to determine the quantity of data by perplexity minimization. It is composed of the following corpora : dev2010, tst2010 tst2013.

3.2.1. Translation model

The translation models have been trained with the standard procedure. First, the bitexts are word aligned in both directions with GIZA++ [11]. Then the phrase pairs are extracted and the lexical and phrase probabilities are computed. The weights have been optimized with MERT.

3.2.2. Language modeling

The language model is an interpolated 4-gram back-off LM trained with SRILM [7] on the selected part of the French corpora made available. The vocabulary contains all the words from the development sets, the target side of bitexts and only the more frequent words from the large monolingual corpora. The interpolation coefficient have been optimized using the standard EM procedure. The perplexity of this model on *liumdev15* was 67.02.

Besides, two large context CSLM [12] have been trained, each with a different architecture. Those models are used to rescore the n-best list of SMT hypotheses. Table 5 shows the details of the architectures of the CSLMs as well as the perplexities obtained on the development corpus *liumdev15*.

Name	Order	Proj. size	#hidd. x size	PPL
CSLM11	11	512	3 x 1024	41.98
CSLM19	19	320	3 x 1024	41.38

Table 5: Architecture of the CSLM trained for rescoring the n-best list of SMT hypotheses. The third and fourth columns show the projection layer size and the number and size of the hidden layers, respectively. The last columns contains the perplexities obtained with each model on *liumdev15*.

3.2.3. Neural network machine translation system

In addition to the phrase-based SMT system, we trained a neural network machine translation (NNMT) system based on [13] during 4 days. It is implemented in the Groundhog framework. It consists in a bidirectionnal encoder-decoder deep neural network equiped with an attention mechanism, as described in Figure 1.

We performed the translation with different values for the beam size. We can observe in Table 6 that the more the beam size is increased, the lower the results in BLEU.

An explanation to this is that the BLEU score differs from the internal score calculated by the model (at the output of the softmax layer). Consequently, a partial hypothesis with

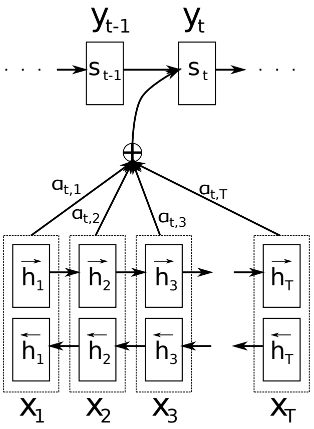

Figure 1: Architecture of the encoder-decoder deep neural network machine translation system equiped with an attention-based mechanism. Taken from [13].

Corpus	*Beam size*		
	10	100	1000
liumtst15	36.79	36.1	35.24
liumdev15	31.62	30.95	30.12

Table 6: Results obtained with the deep NNMT system with different values of beam size.

a low score which is pruned with a small beam size, is kept and extended when the beam size is greater. Moreover, the NN output probability distributions are known to be sharp, giving a high probability to a small number of outputs and a low probability to the rest. This can lead to worse hypotheses having higher results in final. This is an undesirable behavior, which a deeper analysis of the correlation between BLEU score and NN outputs probabilities could explain.

We used the trained NNMT model to rescore the 1000-best list produced by the previously trained SMT model.

3.2.4. Submitted systems

A total of six systems were submitted for evaluation. Several rescoring process have been performed. For the sake of comparison, our best single SMT system has been submitted as *contrastive2* as well as our best NNMT system based on Groundhog (*contrastive5*). This SMT system has been rescored with the two CSLM presented in previous section. *contrastive3* and *contrastive4* correspond to the rescoring with CSLM11 and CSLM19, respectively. Those two systems have also been rescored with the NNMT model obtained with Groundhog. The *primary* system corresponds to the *contrastive3* rescored with Groundhog deep neural network and the *contrastive1* corresponds to the *contrastive4* rescored with the same deep neural translation model.

The results and discussion are presented in the next section.

3.3. Results and discussion

The results obtained on the development and test data are presented in Table 7.

The main observation that we can make is that all the results are coherent. Improvement obtained by rescoring with the CSLM and the NN model on the development corpus are reflected on the internal test (*liumtst15*) and the official evaluation test corpus (*test2015*). The gains observed by rescoring the 1000-best list of hypotheses with a high order CSLM are along previous results in the literature (around +1 BLEU point on development and test data). One can notice that the two different CSLM provide very similar results (in terms of perplexity during training and in terms of BLEU after rescoring).

During system development, we were surprised by the gains observed when rescoring with the NNMT system compared to the lower results obtained (on *liumdev15* and *liumtst15*). An interesting result is that the rescoring with the NNMT model provides similar results on the official test set.

A key point when applying a rescoring process is the optimization of the feature functions weights. The weights for the CSLM and the NNMT model have been optimized with CONDOR [14], a numerical optimizer, with -BLEU as the objective function to minimize. The initial weights are set to those obtained with MERT during the SMT system tuning phase. The initial weights for the CSLM and NNMT features are set to the backoff LM weight (*e.g.* 0.0357). This is motivated by the fact that the LM and CSLM features have a similar distribution. After optimization, the LM had its weights decreased to 0.0314, the CSLM weight increased to 0.0391 while the NNTM feature function saw its weight highly increased (0.0486).

4. Conclusion

We presented the LIUM's ASR and SMT systems which participated in the ASR and SLT tracks of the IWSLT'15 evaluation campaign.

For ASR, we participated to the German transcription task, which is a new challenge to us since we built our first German systems for the campaign. We achieved an official WER of 17.8 of the 2015 test set which seems consistent with our experiments on previous development and test sets.

By rescoring with a continuous space language model, we obtained a gain of about 0.6% BLEU on the SLT test data. On top of that, an additional gain of almost %1 BLEU point is obtained by rescoring with a neural network translation model. The latter result is more surprising since the translation score of the NNMT system is significantly lower than the SMT systems.

5. Acknowledgements

This work was partially funded by the European Commission through the EUMSSI project, under the contract number 611057, in the framework of the FP7-ICT-2013-10 call. This

Name	CSLM	NNMT	*liumdev15*	*liumtst15*	*test2015*			
	rescoring	rescoring			Case		No-Case	
			%BLEU	%BLEU	%BLEU	%TER	%BLEU	%TER
Primary	CSLM11	yes	33.81	39.61	18.51	79.06	20.02	76.41
Contrast1	CSLM19	yes	33.82	39.65	18.53	78.96	20.10	76.29
Contrast2	-	no	31.81	37.35	16.95	80.61	18.36	78.01
Contrast3	CSLM11	no	32.81	38.36	17.54	80.04	19.02	77.31
Contrast4	CSLM19	no	32.70	38.28	17.56	80.07	19.03	77.45
Contrast5	-	-	31.62	36.79	14.88	84.69	16.98	80.38

Table 7: Results obtained with the submitted systems on internal dev and test corpora and the official evaluation test corpus.

work was also partially funded by the French National Research Agency (ANR) through the TRIAGE project, under the contract number ANR-12-SECU-0008-01.

6. References

[1] R. Gretter, "Euronews: a multilingual speech corpus for ASR," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014.

[2] A. Rousseau, "XenC: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 73–82, 2013.

[3] D. Povey, A. Ghoshal, G. Boulianne, L. Burge, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, december 2011, iEEE Catalog No.: CFP11SRW-USB.

[4] H. Schwenk, "CSLM - a modular open-source continuous space language modeling toolkit," in *Interspeech*, august 2013, pp. 1198–1202.

[5] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[6] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, Lyon, France, 2013.

[7] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of Interspeech*, September 2002, pp. 901–904.

[8] S. Meignier and T. Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, Dallas (Texas, USA), mars 2010.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 2007, pp. 177–180.

[10] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1045–1054.

[11] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08, 2008, pp. 49–57. [Online]. Available: http://portal.acm.org/citation.cfm?id=1622110.1622119

[12] H. Schwenk, "Continuous Space Language Models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR'15*, 2015.

[14] F. V. Berghen and H. Bersini, "CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm," *Journal of Computational and Applied Mathematics*, vol. 181, pp. 157–175, September 2005.