The LIUM English-to-French Spoken Language Translation System and the Vecsys/LIUM Automatic Speech Recognition System for Italian Language for IWSLT 2014

Anthony Rousseau[1], Loïc Barrault[1], Paul Deléglise[1], Yannick Estève[1], Holger Schwenk[1]
Samir Bennacef[2], Armando Muscariello[2], Stephan Vanni[2]

LIUM[1], University of Le Mans, France
Vecsys[2], Montigny-le-Bretonneux, France
`firstname.lastname@lium.univ-lemans.fr`
`{sbennacef, amuscariello, svanni}@vecsys.fr`

Abstract

This paper describes the Spoken Language Translation system developed by the LIUM for the IWSLT 2014 evaluation campaign. We participated in two of the proposed tasks: (i) the Automatic Speech Recognition task (ASR) in two languages, Italian with the Vecsys company, and English alone, (ii) the English to French Spoken Language Translation task (SLT). We present the approaches and specificities found in our systems, as well as the results from the evaluation campaign.

1. Introduction

This paper describes the ASR and SLT systems developed by the LIUM for the IWSLT 2014 evaluation campaign. This year, the campaign has the particularity to bring new recognition languages and translation directions, while still proposing TED Talks recognition and translation tasks. Consequently, we participated in the two tasks mentioned above, with English and Italian languages for the ASR task; and English to French for the SLT task. Since we last participated in IWSLT three years ago in 2011, new approaches and specificities were developed by the LIUM, both in the ASR and in the SLT tasks, which will be detailed here. For ASR in Italian, this work was made in collaboration with the Vecsys company.

The remainder of this paper is structured as follows: in section 2, we describe the data used for both tasks and how the selection was performed. In section 3, we present the architecture of our ASR system and the results obtained on the various corpora used during the campaign. Then in section 4, we expose the architecture of our SLT system, along with its results during the campaign. Lastly, the section 6 concludes this system description paper.

2. Data Selection for the Tasks

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. Selection for the ASR task

2.1.1. Acoustic models training data selection

For our acoustic modeling we used as a primary source the TED-LIUM corpus release 2 [1], removing from it all talks recorded after December 31st, 2010. In order to strengthen this base, we first added data from the Euronews corpora [2] distributed by the campaign organizers and from the 1997 English Broadcast News Speech (HUB4) [3]. Then, from the MediaEval 2014 evaluation campaign Search and Hyperlinking Task data transcripts (BBC recordings from 2008 which were decoded by the LIUM) [4], we applied a threshold on our confidence measures to select the best possible segments for our system within a limit of 50 hours of speech. Table 1 summarizes the characteristics of the data included in our ASR system acoustic models.

Corpus	Duration	Segments	Words
TED-LIUM	130.1h	61 796	1 447 022
Euronews	68.2h	33 686	817 649
1997 HUB4	75.0h	20 652	852 517
MedialEval 14	50.0h	46 713	368 118
Total	323.3h	162 847	3 485 306

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

2.1.2. Language models training data selection

Since language models training data is constrained for the ASR task, we applied our data selection tool XenC [5] on each allowed corpus at our disposal: basically all of publicly available WMT14 data, a provided TED Talks closed-captions corpus and the LDC Gigaword. 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. Table 2 summarizes the characteristics of the monolingual text data used to estimate our system language models.

Corpus	Original # of words	Selected # of words	% of Orig.
IWSLT14	0.1M	0.1M	100.00
Common Crawl	195.4M	13.6M	6.98
Europarl v7	56.4M	1.8M	3.22
Gigaword LDC	4 985.3M	168.2M	3.37
10^9 FR-EN	649.4M	11.9M	1.83
News Crawl	1 503.1M	44.8M	2.98
News-Comm. v7	4.7M	0.7M	14.04
UN 200x	360.1M	1.8M	0.50
Yandex 1M	24.1M	4.6M	19.01
Total (w/o IWSLT14)	7 778.5M	247.4M	3.18

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

2.2. Data processing and selection for the SLT task

All available corpora have been used to train the different component of the SMT system. The source side of the bi-texts 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, put capital letters in lowercase and rewrite numbers in letters.

Once the processing performed, monolingual and bilingual data selection has been applied using XenC [5]. For this purpose, the TED corpus has been used as in-domain corpus (to compute in-domain cross-entropy) and the provided development data (dev2010 and tst2010) was used to determine the quantity of data by perplexity minimization.

3. Automatic Speech Recognition Task in English

In this section, we will describe the Automatic Speech Recognition system developed by the LIUM for the IWSLT 2014 campaign, as well as present the results (both in-house and official) obtained on various corpora.

3.1. Architecture of the LIUM ASR system

Our system architecture is mainly based on the Kaldi open-source speech recognition toolkit [6] which uses finite state transducers (FSTs) for decoding. A first step is performed with the Kaldi decoder by using a bigram language model and standard GMM/HMM models to compute a fMLLR matrix transformation. A second decoding step is performed by using the same 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 5-gram Continuous Space Language Model [7].

3.1.1. Speaker segmentation

To segment the audio recordings and to cluster speech segments by speaker, we used the *LIUM_SpkDiarization* speaker diarization toolkit [8]. This speaker diarization system is composed of an acoustic Bayesian Information Criterion (BIC)-based segmentation followed by a BIC-based hierarchical clustering. Each cluster represents a speaker and is modeled with a full covariance Gaussian. A Viterbi decoding re-segments the signal using GMMs with 8 diagonal components learned by EM-ML, for each cluster. Segmentation, clustering and decoding are performed with 12 MFCC+E, computed with a 10ms frame rate. Gender and bandwidth are detected before transcribing the signal. This speaker segmentation was used by all the LIUM and Vecsys ASR systems.

3.1.2. 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 323.3 hours set, with 9 500 tied triphone states and 200 000 gaussians.

On top of these models, we train a deep neural network (DNN) 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 323.3 hours set as the training material. The resulting network is composed of 7 layers for a total of 36.8 millions parameters and each of the 6 hidden layers has 2 048 neurons. The output dimension is 7 296 units and the input dimension is 440, which corresponds to an 11 frames window with 40 LDA parameters each. Weights for the network are initialized using 6 restricted Boltzmann machines (RBMs) stacked

as a deep belief network (DBN). The first RBM (Gaussian-Bernoulli) is trained with a learning rate of 0.01 and the 5 following RBMs (Bernoulli-Bernoulli) are trained with a rate of 0.4. The learning rate for the DNN training is 0.00001. The segments and frames are processed randomly during the network training with stochastic gradient descent (SGD) in order to minimize cross-entropy between the training data and network output. When these training steps are done, the last step of training is processed, by applying the minimum Bayes risk criterion, as indicated above. To speed up the learning process, we use a general-purpose graphics processing unit (GPGPU) and the CUDA toolkit for computations.

3.1.3. Language modeling

For language modeling, we rely on the SRILM language modeling toolkit [9] as well as the Continuous Space Language Model toolkit. The vocabulary used in the LIUM ASR system is composed of 165 371 entries. The bigram language model (2G LM) used during the Kaldi decoding part is trained on the data presented in section 2.1.2.

With the SRILM toolkit, one 2G LM is estimated for each corpus source, with no cut-offs and the modified Kneser-Ney discounting method. These 2G LM are then linearly interpolated to produce the final 2G LM which will be used in the final system, using the IWSLT 2011 development and test corpora. To rescore the word-lattices produced by Kaldi, a trigram and a quadrigram language models (3G and 4G LM) 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. The table 3 details the interpolation coefficients for the 2G, 3G and 4G language models as well as the final perplexity for each final model.

Corpus	Coefficients			
	2G	3G	4G	CSLM
IWSLT14	.36353	.23963	.19110	N/A
Common Crawl	.14404	.26584	.34979	N/A
Europarl v7	.00272	.00244	.00277	N/A
Gigaword LDC	.30076	.27450	.24411	N/A
10^9 FR-EN	.02709	.02882	.02701	N/A
News Crawl	.13535	.14751	.14241	N/A
News-Comm. v7	.00173	.00254	.00220	N/A
UN 200x	.00300	.00411	.00391	N/A
Yandex 1M	.02179	.03461	.03670	N/A
Perplexity	209.31	134.38	107.72	123.03

Table 3: Interpolation coefficients and perplexities for the bigram, trigram, quadrigram and CSLM language models used in the LIUM ASR system.

3.2. Results

We submitted three runs (one primary, two constrastives) for the ASR task. The first contrastive is the one described in section 3.1. The second constrastive is basically the same system as the first, with a DNN similar the the CRIM one described in [10]. The primary is the fusion of the two systems described above at the word-lattices level. The table 4 presents the official results from the campaign organizers. Rankings are not known at the time of this paper publication.

System	tst2014	tst2013
Primary	12.3 %	16.0 %
Contrastive 1	13.4 %	17.3 %
Contrastive 2	13.8 %	17.4 %

Table 4: Official results (Word Error Rate) for the LIUM at the IWSLT 2014 Automatic Speech Recognition track.

4. Spoken Language Translation Task

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

4.1. Architecture of the LIUM SLT system

The SMT system is based on the Moses toolkit [11]. 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, an Operation Sequence Model (OSM) [12] have been trained and included in the system.

4.1.1. Translation model

The translation models have been trained with the standard procedure. First, the bitexts are word aligned in both directions with GIZA++ [13]. Then the phrase pairs are extracted and the lexical and phrase probabilities are computed. The weights have been optimized with MERT using two versions of the development data. For some systems, the provided transcriptions were used, and for others, the outputs of our ASR system was used. This was performed for the sake of comparing the impact of ASR systems improvement (observed during the last few years).

4.1.2. Language modeling

The language model is an interpolated 4-gram back-off LM trained with SRILM [9] 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 was 69.37.

In addition, several large context CSLM [14] have been trained, each with a different architecture. Those models are used (alone or in combination) to rescore the n-best list of SMT hypotheses. The weights for the CSLM have been optimized with CONDOR [15], a numerical optimizer, with $-$BLEU as objective function to minimize.

Name	Order	Projection Layer	Perplexity
CSLM1	12	384	40.72
CSLM2	12	448	40.19
CSLM3	16	384	40.58

Table 5: Architecture of the various CSLM trained for rescoring the n-best list of SMT hypotheses.

4.1.3. Submitted systems

A total of six systems were submitted for evaluation. One of the differences lies in the development data used for tuning. The provided development data corresponds to ROVER outputs of several years old ASR systems. Considering that ASR systems have greatly evolved during the last few years, we thought that comparing an SMT system tuned with outputs of an old combination of ASR systems with a state-of-the-art ASR system would be interesting. The other difference concerned the use of a rescoring step. As mentioned in the previous section, several CSLM have been trained. Some systems did not include any rescoring process at all, some use only one CSLM and some combined the three CSLM probabilities to determine the best hypothesis. When using only one CSLM, the best performing model on the development data has been chosen. The results and discussion are presented in the next section.

4.2. Results and discussion

The results obtained on the development and test data are presented in Table 6.

We translated two version of the test data. $test2014 - iwslt$ is the provided test data, which corresponds to a ROVER combination of the outputs of the systems participating in the IWSLT'14 ASR task. $test2014 - lium$ corresponds to the 1-best output of the LIUM ASR system.

The first comment is that the results that we observed on the development data are not reflected in the test data. Tuning with two versions of the development data, providing difference of more than 2 BLEU points results in similar scores on the test data. This is well understood when there is a mismatch between tuning and testing conditions (i.e. tuning with $-lium$ corpus and testing on $-iwslt$). As the ASR results have not been provided yet, we can't make the link between the WER and the SMT results. Also, a deeper analysis of the outputs have to be performed in order to explain this behavior.

The main improvements are obtained by rescoring the 1000-best list of hypotheses with one or more CSLM. By comparing Contrast2 and Contrast4 systems on one hand, and Contrast4 and Contrast6 systems on the other hand, we can observe that CSLM rescoring provide a gain of up to 1.2 and 1.78 BLEU respectively on the development and test data.

However, combining the three different CSLM does not provide anymore gain. This was already observed on the development data, but the result was never worse than using only one language model. This tends to prove that CSLM with different architectures (context and projection layer size in this case) does not have a great impact on the final score.

5. Automatic Speech Recognition Task in Italian

The ASR system used to process Italian data is a combination of the Vecsys ASR system and the LIUM ASR system. Both systems share the same speaker segmentation and the same training data, very restricted in the ASR task for Italian. The speaker diarization system is the same as the one used to process English data.

5.0.1. Training data

To train language models for Italian, the number of authorized sources of training data was very low. We used the data provided by the organizers to train language and acoustic models, in addition to the Italian Google n-grams, listed in the permissive data (LDC2009T25). For acoustic models, in addition to the Euronews corpora [2] distributed by the organizers, we used about 100 hours of manually annotated data owned by the Vecsys company, and recorded before June 30th 2011. Notice that we extracted about 75h from the Euronews automatically annoted data: about 175 hour of recordings were used to train the acoustic models of the Vecsys and LIUM systems.

5.1. Vecsys system

Vecsys speech recognition system is based on a multi-pass GMM/HMM decoding of the input speech, mostly derived from the CMU Sphinx toolkit. A first pass aims at providing a transcription which, in accordance with the speaker segmentation, is employed to estimate speaker-specific fMLLR matrices for feature transformation. The transformed features are used in a second decoding that produces word lattices, using the same trigram back-off language model as for the first pass, and then acoustically rescored to improve inter-word senone scores. The final transcription is obtained by joint linguistic rescoring of the word lattices from a 4 gram back-off and a 4-gram continuous space language model, followed by a confusion network decoding.

Name	CSLM	Dev		test2014-iwslt				test2014-lium			
				Case		No-Case		Case		No-Case	
		Name	%BLEU	%BLEU	%TER	%BLEU	%TER	%BLEU	%TER	%BLEU	%TER
Primary	Comb	d10t10lium	25.79	26.82	59.40	27.85	57.69	24.90	60.93	25.92	59.55
Contrast1	Comb	d10t10iwslt	23.30	26.78	59.40	27.82	57.99	25.06	61.10	26.04	59.73
Contrast2	CSLM1	d10t10lium	25.70	26.76	58.82	27.81	57.46	24.98	60.72	25.99	59.33
Contrast3	CSLM3	d10t10iwslt	23.24	26.89	59.36	27.94	57.94	24.95	61.43	25.96	59.97
Contrast4	-	d10t10lium	24.49	25.17	59.83	26.17	58.47	23.59	61.66	24.52	60.26
Contrast5	-	d10t10iwslt	22.26	25.14	60.61	26.16	59.21	23.65	62.24	24.64	60.82

Table 6: Results obtained with the submitted systems. Corpora d10t10lium and d10t10iwslt correspond to respectively the transcription obtained with the LIUM ASR system and the provided development data.

5.1.1. Acoustic modeling

Italian phonetic lexicon is describes by a set of 27 phonemes, and, for all consonants, gemination is modeled by doubling the consonant symbol in a word pronunciation, rather than defining a special symbol for the geminate consonant. The GMM-HMM acoustic models are computed from 13-dimensional PLP features (including energy) to which first and second order derivates are appended. In all decoding steps, PLPs are multiplied by an LDA and a MLLT matrix, both estimated on the same training data, to obtain a 29-dimensional vector. The first pass uses a light-weight set of models which comprises 6000 senones, each modeled by a 16-component GMM, estimated by ML modeling, and adapted by MAP according to gender. The second pass uses 8000 senones, each modeled by a 32-component GMM, estimated by MPE modeling from an initial set of m/f SAT models.

5.1.2. Language modeling

Back-off language models are obtained by interpolation of two back-off models, one estimated by Witten-Bell discounting on the Google N-gram corpus, the other from Kneser-Ney discounting and no cut-off on the TED transcriptions provided by the organizers. On this same corpus, a 4 gram continuous space language model is trained: scores are computed for 4-grams of words included in a short list of 16384 words out of 109500 words. Such scores are linearly interpolated with those read from the back-off model for the corresponding 4 grams.

5.2. LIUM ASR system for Italian

The architecture of the LIUM ASR system for Italian is the same as the one described in this paper for English language. The phonetic lexicon was built from the lexicon provided in the Festival tool for speech synthesis [16], by using the statistical grapheme-to-phoneme (g2p) tool described in [17] in order to compute the pronunciation of words not included in the Festival Italian lexicon. This Festival lexicon contains about 400,000 words.

5.3. Merging Vecsys and LIUM ASR systems

Vecsys and LIUM used the same audio segmentation, provided by the *LIUM_SpkDiarization* speaker diarization system. 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, as described in [18].

6. Conclusion

We presented the LIUM's and Vecsys' ASR and SMT systems which participated in the ASR and SLT tracks of the IWSLT'14 evaluation campaign.

By integrating some of the latest LIUM developments in Automatic Speech Recognition, we were able to achieve a Word Error Rate score of 12.3 % on the ASR evaluation track. While we currently can't compare it to other results for the tst2014 corpus, we can compare the 16.0 % tst2013 score to the last year results, which would have been ranked 4th.

By rescoring with a continuous space language model, we obtained a gain of about 1.7% BLEU on the SLT test data. However, the combination of several CSLM rescoring did not produced anymore gain.

7. Acknowledgements

8. References

[1] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014.

[2] 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.

[3] J. Fiscus, J. Garofolo, M. Przybocki, W. Fisher, and D. Pallett, "1997 English broadcast news speech (HUB4) LDC98S71," Web Download, 1998.

[4] M. Eskevich, R. Aly, D. N. Racca, R. Ordelman, S. Chen, and G. J. Jones, "The search and hyperlinking task at MediaEval 2014," in *Working Notes Proceedings of the MediaEval 2014 Workshop*, Barcelona, Spain, october 2014.

[5] 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.

[6] 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.

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

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

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

[10] A. Rousseau, G. Boulianne, P. Deléglise, Y. Estève, V. Gupta, and S. Meignier, "LIUM and CRIM ASR system combination for the REPERE evaluation campaign," in *Text, Speech and Dialogue*. Springer International Publishing, 2014, pp. 441–448.

[11] 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 *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (Demo and Poster Sessions)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 177–180. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-2045

[12] 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*. Association for Computational Linguistics, 2011, pp. 1045–1054.

[13] 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. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 49–57. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622110.1622119

[14] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, pp. 492–518, 2007.

[15] F. Vanden Berghen and H. Bersini, "Condor, a new parallel, constrained extension of powell's uobyqa algorithm: Experimental results and comparison with the dfo algorithm," *J. Comput. Appl. Math.*, vol. 181, no. 1, pp. 157–175, Sept. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.cam.2004.11.029

[16] P. Cosi, F. Tesser, R. Gretter, C. Avesani, and M. Macon, "Festival speaks Italian!" in *INTERSPEECH*, 2001, pp. 509–512.

[17] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[18] A. Rousseau, G. Boulianne, P. Deléglise, Y. Estève, V. Gupta, and S. Meignier, "LIUM and CRIM ASR System Combination for the REPERE Evaluation Campaign," in *Text, Speech and Dialogue*. Springer, 2014, pp. 441–448.