

## LIUM's systems for the IWSLT 2011 Speech Translation Tasks

*Anthony Rousseau, Fethi Bougares, Paul Deléglise,  
Holger Schwenk, Yannick Estève,*

Laboratoire Informatique de l'Université du Maine (LIUM)  
University of Le Mans, France

`firstname.lastname@lium.univ-lemans.fr`

### Abstract

This paper describes the three systems developed by the LIUM for the IWSLT 2011 evaluation campaign. We participated in three of the proposed tasks, namely the Automatic Speech Recognition task (ASR), the ASR system combination task (ASR\_SC) and the Spoken Language Translation task (SLT), since these tasks are all related to speech translation. We present the approaches and specificities we developed on each task.

### 1. Introduction

This paper describes the three systems developed by the LIUM for the IWSLT 2011 evaluation campaign [1]. This year, new interesting tasks were proposed compared to last year evaluation campaign. As a matter of fact, the three tasks we participated in are all linked together in the same pipeline: speech recognition, ASR system combination and speech translation. Like the last year campaign, all of the considered tasks were related to the TED talks, requiring speech recognition of English, and speech translation from English to French.

The remainder of this paper is structured as follows: in section 2, we describe our system setup for the ASR task and the specific corpus we created for it. In section 3, we discuss our participation to the ASR system combination task along with the different approaches we considered. Then in section 4, we expose our contributions to the SLT task, and particularly our data selection techniques. Lastly, the paper concludes on a discussion on encountered issues and future work perspectives in section 5.

### 2. Automatic Speech Recognition Task

In this section, we describe how our ASR system has been developed. First, we explain how we made our own training corpus based on TED talks, then we describe the system itself and finally we discuss the results we obtained on the IWSLT 2011 ASR task.

#### 2.1. The LIUM's TED corpus

For our system training, we aimed at using audio and transcripts from the TED talks. In order to get the desired data, we developed a specific tool intended to grab it from the TED website. This led us to dispose of 818 audio files (talks), along with their corresponding transcripts, for a total of 216 hours of audio (192 hours of speech) distributed among 698 unique speakers. Among these speakers, we have 129 hours of male speech and 63 hours of female speech.

Unfortunately, the TED transcripts are not exact transcripts (verbatim) of what the speakers pronounce in their talks. For instance, they lack speech disfluencies like repetitions or hesitations and some expressions or contractions are either missing or transcribed differently. Moreover, there is no proper segmentation and the timings we extracted aren't precise at all, thus they can't be used in order to train an ASR system.

##### 2.1.1. Creating and refining the training corpus

In order to dispose of the collected data as a real ASR training corpus, the first step we needed to achieve was to get proper alignments from the downloaded transcripts. We started by generating an automatic segmentation of the audio data using our in-house speaker segmentation and clustering tool (*LIUM\_SpkDiarization*), presented in [2]. The idea was to initiate the process by first decoding all the available audio using the default acoustic model provided in the CMU Sphinx 3 package and a 4-gram language model trained on all the text contained in the transcripts. Then, using the NIST Scoring Toolkit *sclite* tool compiled with the *diff* algorithm option enabled, we were able to map the unaligned text to our output, thus creating rough reference STM files for our audio data, based on our CTM timings.

This way, by scoring our decoder outputs against the newly-created reference files, we were able to get an idea of the quality of our alignments, even if the WER score is not a good metric to measure this. This helped us to remove the worst-aligned talks, and left us with 794 talks representing

135 hours of speech: 91 hours of male and 44 hours of female.

This first iteration (the bootstrap) led us to train a new acoustic model based on our TED audio. Then, by performing a forced alignment and decoding our audio again, we were able to generate a more accurate set of reference STM files. In order to get a sufficient accuracy, we only kept part of the segments where the decoding output and the unaligned text agreed at least on the first and last word of the segment, thus removing words on each extremity. This second iteration left us with 779 talks, for an amount of speech of 152 hours, 106 hours of male and 46 hours of female. Starting from this data, and for a third time, we trained a new acoustic model and decoded all our speech. We then kept only segments which were consistent enough, *i.e.* the segments which were perfectly aligned (word by word) with the original text. This way, we were able to circumvent the fact that our transcript text was approximative.

In the end, our TED corpus is composed of a total of 773 talks, representing 118 hours of speech: 82 hours of male and 36 hours of female. The table 1 resumes the statistics of our corpus for each iteration.

Iteration	#Talks	Speech (hours)	Gender speech	
			Male	Female
0 (orig.)	818	192 h	129 h	63 h
1	794	135 h	91 h	44 h
2	779	152 h	106 h	46 h
3 (final)	773	118 h	82 h	36 h

Table 1: TED audio training corpus statistics by iteration.

In order to enhance our corpus with some diversity, we added parts of the 1997 English Broadcast News Speech corpus (HUB4) which represent a total of 65 hours of speech (41 hours of man speech and 23 hours of woman speech).

### 2.1.2. Creating the development corpus

When training an ASR system, it is mandatory to have at our disposal a development corpus well related to the training data, with precise transcriptions. This helps achieving fine tuning of all weights used in the system. In order to get such data, we took the talks used for the IWSLT 2010 dev and test corpora, and transcribed them manually to get references as precise as possible (since the TED transcriptions are not verbatim transcriptions).

In terms of size, this development corpus, composed of 19 talks, represents a total of 4 hours and 13 minutes of speech. Among these, male speech counts for 3 hours and 14 minutes, while female speech represents 59 minutes.

## 2.2. Architecture of the LIUM’s ASR system

### 2.2.1. Vocabulary and language modeling

In order to select the optimal vocabulary for our system training, we trained unigram language models on the textual data from TED, HUB4 and all the monolingual corpora proposed for the IWSLT 2011 task. In a second time, we interpolate them to get a global unigram model. That model is then sorted according to the word probabilities in reverse order, which allows us to select the most likely words appearing in the corpora, as described in [3]. For our system, we selected the 150k most likely words, plus all of the TED and HUB4 words to ensure that our system training would be consistent. This left us with a vocabulary size of 157,6k words.

To train our language models (LM), we used the SRILM toolkit [4]. The selected vocabulary is exactly the same as the one described above to keep the system’s consistency. We trained several 4-gram LMs, one for each monolingual corpus, which were then interpolated to create the final LM, a 4-gram back-off model with Kneser-Ney discounting. The interpolation weights are computed with an EM procedure, using the textual data from the unmodified development corpus mentioned in section 2.1.2. Given the vocabulary limited size of our system, we didn’t apply any cut-offs on the final language model.

### 2.2.2. Description

Our in-house ASR system is a five-pass system based on the open-source CMU Sphinx system (version 3 and 4), similar to the LIUM’08 french ASR system described in [5]. The acoustic models were trained in the same manner, except that we added a multi-layer perceptron (MLP) using the Bottle-Neck feature extraction as described in [6].

The input speech representation of our MLP is a concatenation of nine frames of thirty-nine MFCC coefficients (twelve MFCC features, energies,  $\Delta$  and  $\Delta^2$  derivatives). The topology of the MLP is the following: the first hidden layer is composed of 4000 neurons, the second one, used as the decoding output, of 40 and the third one, used for training, of 123 (41 phonemes, 3 states per phoneme). For the decoding, we first perform a Principal Components Analysis (PCA) transformation on the 40 parameters. Then two streams are decoded: the first one is composed by the 40 parameters of the PCA transformation while the second one is made of 39 standard PLP features. The streams likelihoods are weighted in order to obtain a resulting likelihood dynamic similar to one single PLP stream. Training of the MLP features is performed using the ICSI QuickNet libraries (see [7]).

Here is a summary of the five passes performed by the system for decoding:

- #1 The first pass uses generic acoustic models and a 3-gram language model.
- #2 The best hypotheses generated by pass #1 are used to compute a CMLLR transformation for each speaker. Decoding #2, using SAT and Minimum Phone Error (MPE) acoustic models and CMLLR transformations, generates word-graphs.
- #3 In the third pass, the computed MLP features are used to rescore the word-graphs obtained during the second pass.
- #4 The fourth pass consists in expanding with a 4-gram language model the linguistic scores of the updated word-graphs of the third pass.
- #5 The last pass generates a confusion network from the fourth pass expanded word-graphs and applies the consensus method to extract the final one-best hypothesis.

### 2.3. Results

The official results from the IWSLT 2011 organizers showed that our system performed well, be it on development and test data from IWSLT 2010, where it was the second best system, or on test data from this year’s campaign, where it was the third best system.

Unfortunately, no information is given by the organizers regarding the size of other systems training data.

Data set	LIUM system (WER)	Best system (WER)
Dev 2010	19.2%	17.8%
Test 2010	18.2%	15.8%
Test 2011	17.4%	15.3%

Table 2: Official results for LIUM ASR system, in WER.

## 3. ASR System Combination Task

Improvements from ASR system combination are usually best when systems are sufficiently different. Since there are five different ASR system submitted to the IWSLT 2011 ASR task, many system combination schemes can be applied.

In this section we describe different combination methods used during IWSLT ASR combination task. We also present the performance of each combination on the development corpus.

### 3.1. Combination methods

Many combination methods have been proposed in the literature. These techniques perform at different level and most

of them operate at the ASR output level. ROVER [8], CNC [9] and lattice combination [10] are examples of such kind of combination. At the same time, other techniques are developed either to combine models (acoustic and linguistic) or to operate on decoding process.

#### 3.1.1. ROVER

ROVER is a simple voting mechanism using the 1-best output from each component system. This combination is divided in two steps: alignment and voting. Previous experiments have shown that best results are obtained when combined systems are ordered by increasing WER. Initial performance of combined systems and ROVER combination using NIST tools are reported in table 3.

Systems	dev2010 (WER)	tst2010 (WER)
System 0	21.2%	19.7%
System 1	23.7%	22.3%
System 2 (LIUM)	19.2%	18.2%
System 3	28.7%	28.0%
System 4	17.8%	15.8%
Rover-4-2-0-1	16.2%	14.6%

Table 3: Performance of the ASR submitted systems for IWSLT 2011 on the development corpus.

#### 3.1.2. Bag Of NGram driven decoding

Bag Of NGram driven decoding (BONG) is a combination method operating at the decoding process level [11]. This combination method takes into consideration hypotheses coming from auxiliary systems by merging all the n-grams observed in each segment into the same bag of n-grams (with  $n = 3$ ). These bags of n-grams are used during the decoding process of the primary system in order to reevaluate the linguistic scores. This combination method proposes new hypotheses which are neither proposed by the auxiliary systems nor by the primary one. The final output can then be integrated in the initial ROVER combination scheme. Table 4 presents results when BONG combination is applied using the system 4 and the system 0 as auxiliary (BONG<sub>4-0</sub>).

As shown in table 4, BONG combination improve primary system (LIUM) accuracy by 0.9 and 0.8 absolute points respectively on dev and test set. In addition, output of BONG combination generate also complementary system which can be added to the final ROVER combination (Rover-(4-BONG-2-0-1)) to get an accuracy improvement of 0.5 absolute points on development and test set, compared to initial ROVER.

System	dev2010 (WER)	tst2010 (WER)
LIUM	19.2%	18.2%
BONG <sub>4-0</sub>	18.3%	17.0%
Rover-4-2-0-1	16.2%	14.6%
Rover-(4-BONG-2-0-1)	15.7%	14.1%

Table 4: Combinations Word Error Rate.

## 4. Spoken Language Translation Task

In this section, we explain how our Statistical Machine Translation (SMT) system for SLT was built. First, we describe the SLT system itself. Then, we detail how the resources provided or allowed for the task were selected to train the translation and language models of the system. Finally, we emphasize on input type selection, weighing of the corpora and results obtained on the IWSLT 2011 SLT Task.

All initial experiments (including data selection) have been done on internal versions of the development and test sets, which have been built by merging our ASR lattice output with the first external ASR lattice output available (released by the organizers after the *dev2010* and *tst2010* run submissions) in a confusion network and extracting the most likely solution from it. We also changed the repartition of the talks between the original development and test set, increasing the size of the dev set by reducing the size of the test set, in order to make the tuning process more robust with more data. We call these sets *LIUM dev2010* and *LIUM tst2010*. We then introduced different input types, after the baseline system had been fixed.

Moreover, all of our data was processed by a newer version of our in-house script first described in [12] and based on previous work by [13]. The goal of this script is to make training, development and test data resemble to ASR outputs.

### 4.1. Architecture of the LIUM’s SLT System

The goal of statistical machine translation (SMT) is to produce a target sentence  $e$  from a source sentence  $f$ . Our system is a phrase-based system [14, 15] which uses a log linear framework in order to introduce several models explaining the translation process:

$$\begin{aligned}
 e^* &= \arg \max_e p(e|f) \\
 &= \arg \max_e \left\{ \exp \left( \sum_i \lambda_i h_i(e, f) \right) \right\} \quad (1)
 \end{aligned}$$

The feature functions  $h_i$  are the system models and the  $\lambda_i$  weights are typically optimized to maximize a scoring function on a development set [16].

The phrase-based system uses fourteen features functions, namely phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, a word and a phrase penalty and a target language model (LM).

#### 4.1.1. Description

Our system is based on the Moses SMT toolkit [17] and is constructed as follows. First, word alignments in both directions are calculated. We used a multi-threaded version of the GIZA++ tool [18].<sup>1</sup> This speeds up the process and corrects an error of GIZA++ that can appear with rare words.

Phrases and lexical reorderings are extracted using the default settings of the Moses toolkit. The parameters of Moses were tuned on *LIUM dev2010*, using the MERT tool.

#### 4.1.2. Language modeling

The French language models were trained on all the French parts of the allowed parallel corpora, in addition to the proposed News monolingual corpus. 4-gram back-off LMs were used. The word list contains all the French words of our phrase table filtered on the 150k words from the ASR decoding vocabulary. Separate LMs were build on each data source with the SRI LM toolkit [4] and then linearly interpolated, optimizing the coefficients with an EM procedure. The perplexity of this LMs was 96.0.

In addition, we build a 5-gram continuous space language model for French [19]. This model was trained on the same data than the back-off model, using a resampling technique. The continuous space language model is interpolated with the 4-gram back-off model and used to rescore n-best lists. This reduces the perplexity to 84.9.

#### 4.1.3. Recasing

Since our SLT system does not contain any punctuation or case, it was necessary to recover them for the final output. We used the same technique as in last year’s evaluation campaign, namely recasing using a separate SMT system dedicated to this task [12]. This technique is summarized by figure 1.

The main differences with last year are:

- less but more appropriate training data using perplexity data selection based on a French in-domain LM;
- suppression of the lexical reordering (instead of limiting it);
- a better development set for the tuning of the system, coming from a real ASR output.

<sup>1</sup>The source is available at <http://www.cs.cmu.edu/~qing/>

Corpus	#En tokens (millions)		#Fr tokens (millions)	
	Orig.	ASR	Orig.	ASR
TED	2.0M	1.8M	2.2M	2.0M
News-Comm.	2.8M	2.6M	3.3M	3.1M
Europarl v6	50.6M	46.6M	56.2M	51.2M
ccb2	232.5M	220.0M	272.6M	258.4M
<b>TOTAL</b>	<b>287.9M</b>	<b>271.0M</b>	<b>334.3M</b>	<b>314.7M</b>
LIUM dev2010	N/A	39k	N/A	39k
LIUM tst2010	N/A	9k	N/A	9k

Table 5: Characteristics of the considered parallel corpora. Orig is the original data while ASR is the processed data.

## 4.2. Bilingual data selection

For this task, we considered the following corpora among those available: the latest versions of News-Commentary and Europarl, the TED corpus provided by the organizers and a subset of the French-English 10<sup>9</sup> Gigaword. Like the last year’s evaluation campaign, we didn’t took into account the un200x corpus due to our experiments, showing its inappropriate style regarding the TED in-domain data. The Gigaword corpus was filtered with the same techniques used in our WMT 11 systems, as described in [20]. We call this internal subset ccb2. Table 5 summarizes the characteristics of those different corpora.

In order to select the best set of corpora for the system, we built several baseline systems with different sets of training data, to evaluate the impact on performance induced by each one of them. Table 6 presents the results we obtained for each of our experiments.

What we found is the following: surprisingly, the very small nc6-TED set is globally better than the biggest eparl6-ccb2-nc6-TED set, although it uses sixty times less data. These experiments also shown that introducing the nc6 corpus helps a lot, while adding the eparl6 corpus

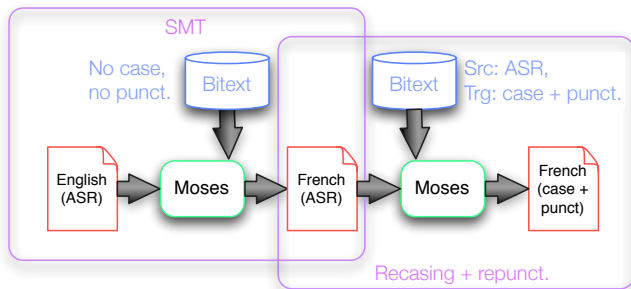


Figure 1: Global architecture of our SLT system, with recasing.

Data set	#En filtered tokens	LIUM dev2010	LIUM tst2010
TED	1.8 M	23.69	25.09
<b>nc6-TED</b>	<b>4.4 M</b>	<b>24.30</b>	<b>25.68</b>
eparl6-nc6-TED	51.0 M	23.92	25.38
eparl6-ccb2-nc6-TED	271.0 M	24.34	25.32

Table 6: BLEU results in function of the used bitexts (no case, no punct).

deteriorates the performance of the system. Looking at these results, it also appeared that adding our ccb2 subset of Gigaword should help to achieve better performance, but with the help of a second level of filtering to discard the out-of-domain data.

In order to filter our ccb2 corpus, we tried a filtering approach based on LM perplexities, inspired by previous work described in [21]. We first built a 4-gram LM on the English data from the TED corpus. Using this LM, we computed the perplexity of each sentence from the ccb2 English part and sorted them in an ascending order. We then applied different thresholds on the sorted corpus and the resulting sets were integrated in our training data, in order to study the impact of the selection on our system’s performance as shown in table 7.

We can observe that adding our ccb2 subset has no impact on the development data, while it improves significantly the performance on test data (when the threshold is inferior or equal to 70). The best compromise between performance, training data size and need in computing resources resides in a threshold equal to 70.

## 4.3. Speech translation and corpus weighing

Starting from our baseline ccb2.px70-nc6-TED system, we had to determine which type of input would be the best for translation. We considered three different types of input: our ASR system 1-best output, the output extracted from the lattice fusion we made using the confusion networks and

Data set	#En tokens	LIUM dev2010	LIUM tst2010
nc6-TED-ccb2.px50	4.9 M	24.22	25.98
nc6-TED-ccb2.px60	5.2 M	24.20	25.87
<b>nc6-TED-ccb2.px70</b>	<b>5.7 M</b>	<b>24.17</b>	<b>26.04</b>
nc6-TED-ccb2.px80	6.2 M	24.29	25.29
nc6-TED-ccb2.px100	7.4 M	24.28	25.45
nc6-TED-ccb2.px150	11.9 M	24.31	25.39

Table 7: Performance of the system given the filtering threshold for the ccb2 corpus (no case, no punct).

Weighting	LIUM 1-best output		Lattice fusion		BONG	
	dev2010	test2010	dev2010	test2010	dev2010	test2010
ccb2.px70-nc6-TED	23.63	24.62	24.17	<b>26.04</b>	24.65	26.34
ccb2.px70-2xnc6-7xTED			24.18	25.72	24.82	26.50
ccb2.px70-2xnc6-8xTED	23.97	25.01	24.16	25.86	24.67	<b>26.78</b>

Table 8: Results of our SLT system given the input type and the weighting applied (no case, no punct).

Enhancements	LIUM 1-best		BONG	
	dev2010	tst2010	dev2010	tst2010
Baseline	23.97	25.01	24.67	<b>26.78</b>
+ CN	24.17	25.44	N/A	N/A
+ CSLM	24.30	25.67	24.97	<b>27.05</b>

Table 9: Enhancements applied to our baseline system (no case, no punct).

finally the BONG output described in section 3.1.2.

We also considered a basic corpus weighting technique based on the interpolation coefficients calculated during the French LM building. By duplicating the in-domain TED corpus several times in the data set, we enhance its weight thus its relative importance.

The tables 8 and 9 shows the BLEU scores obtained given the input type used, the weights applied on the TED and nc6 corpora and the enhancements integrated in the system. We selected the BONG input for our primary submission. We also submitted a contrastive run based on a 100% LIUM ASR output, using the improved word lattice technique with confusion networks presented in last year’s campaign, where the moses 1-best input is replaced with the confusion networks from our decoder [12].

Table 10 presents the final results for these two submitted runs on the official *tst2011*. Two text conditions were considered: with casing and punctuation, then without casing and punctuation.

## 5. Conclusion

We presented the development of our systems for three of the IWSLT 2011 evaluation campaign tasks: automatic speech recognition (ASR), ASR system combination (ASR\_SC) and spoken language translation (SLT).

Submission	tst2011	
	case+punct	no case+nopunct
Primary	28.23	29.40
Contrastive	26.96	28.16

Table 10: Official results for LIUM SLT system, in BLEU.

In the official evaluation, the English ASR system ranked third, the English ASR\_SC system ranked second and the English–French SLT system ranked first according to the BLEU score for both text conditions. The contrastive run also ranked second for case+punct. condition and third for nocase+nopunct condition.

## 6. References

- [1] M. Federico, L. Bentivogli, M. Paul, and S. Stueker, “Overview of the IWSLT 2011 evaluation campaign,” in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, (San Francisco (CA)), December 2011.
- [2] S. Meignier and T. Merlin, “LIUM SpkDiarization: an open source toolkit for diarization,” in *CMU SPUD Workshop*, (Dallas (Texas, USA)), mars 2010.
- [3] A. Allauzen and J.-L. Gauvain, “Construction automatique du vocabulaire d’un système de transcription,” in *Journées d’Étude sur la Parole*, 2004.
- [4] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *International Conference on Spoken Language Processing*, pp. 257–286, November 2002.
- [5] P. Deléglise, Y. Estève, S. Meignier, and T. Merlin, “Improvements to the LIUM french ASR system based on CMU Sphinx: what helps to significantly reduce the word error rate?,” in *Interspeech 2009*, (Brighton (United Kingdom)), 6-10 september 2009.
- [6] F. Grézl and P. Fousek, “Optimizing bottle-neck features for LVCSR,” in *2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4729–4732, IEEE Signal Processing Society, 2008.
- [7] ICSI, *QuickNet*, 2011 (accessed October 10, 2011). <http://www.icsi.berkeley.edu/Speech/qn.html>.
- [8] J. Fiscus, “A post-processing system to yield reduced word error rates: recogniser output voting error reduction (ROVER),” in *ASRU*, pp. 347–354, 1997.

- [9] G. Evermann and P. Woodland, “Posterior probability decoding, confidence estimation and system combination,” in *Proceedings of the Speech Transcription Workshop*, (College Park, MD), 2000.
- [10] M. Chengyuan, J.-K. Hong-Kwang, and S. Hagen, “A comparative study on system combination schemes for LVCSR,” in *ICASSP*, pp. 4394–4397, 2010.
- [11] F. Bougares, Y. Estève, P. Deléglise, and G. Linarès, “Bag of n-gram driven decoding for LVCSR system harnessing,” in *ASRU*, 2011.
- [12] A. Rousseau, L. Barrault, P. Deléglise, and Y. Estève, “LIUM’s statistical machine translation system for IWSLT 2010,” in *International Workshop on Spoken Language Translation (IWSLT) 2010*, (Paris (France)), 2-3 déc. 2010.
- [13] D. Déchelotte, H. Schwenk, G. Adda, and J.-L. Gauvain, “Improved machine translation of speech-to-text outputs,” in *Proc. of Interspeech*, (Antwerp, Belgium), pp. 2441–2444, August 2007.
- [14] P. Koehn, F. J. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*, (Edmonton, Canada), pp. 48–54, 2003.
- [15] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*, vol. 29, pp. 19–51, March 2003.
- [16] F. Och and H. Ney, “Discriminative training and maximum entropy models for statistical machine translation,” in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, PA), pp. 295–302, July 2002.
- [17] 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), pp. 177–180, Association for Computational Linguistics, June 2007.
- [18] Q. Gao and S. Vogel, “Parallel implementations of word alignment tool,” in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing, SETQA-NLP ’08*, (Stroudsburg, PA, USA), pp. 49–57, Association for Computational Linguistics, 2008.
- [19] H. Schwenk, “Continuous space language models,” *Computer Speech and Language*, vol. 21, pp. 492–518, 2007.
- [20] H. Schwenk, P. Lambert, L. Barrault, C. Servan, S. Abdul-Rauf, H. Afli, and K. Shah, “LIUM’s SMT machine translation systems for WMT 2011,” in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, (Edinburgh, Scotland), pp. 464–469, Association for Computational Linguistics, July 2011.
- [21] J. Gao, J. Goodman, M. Li, and K.-F. Lee, “Toward a unified approach to statistical language modeling for chinese,” in *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 1, (New York, NY, USA), pp. 3–33, ACM, March 2002.