LIMSI English-French Speech Translation System

Natalia Segal[1], *Hélène Bonneau-Maynard*[1,2], *Quoc Khanh Do*[1,2],
Alexandre Allauzen[1,2], *Jean-Luc Gauvain*[1], *Lori Lamel*[1], *François Yvon*[1]

LIMSI-CNRS[1] and University Paris-Sud[2], rue John von Neumann, F 91403 Orsay

firstname.lastname@limsi.fr

Abstract

This paper documents the systems developed by LIMSI for the IWSLT 2014 speech translation task (English→French). The main objective of this participation was twofold: adapting different components of the ASR baseline system to the peculiarities of TED talks and improving the machine translation quality on the automatic speech recognition output data. For the latter task, various techniques have been considered: punctuation and number normalization, adaptation to ASR errors, as well as the use of structured output layer neural network models for speech data.

1. Introduction

LIMSI participated in the IWSLT 2014 Evaluation Campaign in the spoken language translation (SLT) task for English→French language pair. Although LIMSI hosts both automatic speech recognition (ASR) and machine translation (MT) research activities, this was our first contribution to the SLT task and the effort was thus focused on one single translation direction. This year's SLT task consists in automatic transcription and translation of a test set composed of several recordings of TED online conferences[1]. The automatic speech transcriptions that have been used in our experiments were produced by the in-house ASR system adapted to TED data, rather than using the transcripts provided by the organizers (hypotheses from several automatic speech recognizers combined using the ROVER approach). As far as the automatic translation step is concerned, we addressed various typical challenges of SLT: to bring automatic transcriptions closer to the expectations of the MT system (mainly trained on written text), to adapt MT models to erroneous ASR output, and to improve the general translation quality.

This paper is structured as follows. We first present the ASR system and the adaptation steps taken to improve the automatic transcriptions of the TED data. We then describe various approaches used to bring the ASR output data and the expected MT input data format into accordance with each other, as well as our attempts to adapt standard MT systems to ASR output. Finally, the impact of re-scoring n-best translation hypotheses using SOUL models is presented in the closing section.

[1] https://www.ted.com/

2. ASR systems: adaptation to TED talks data

The LIMSI automatic speech recognition system for broadcast data [1] was adapted to the task of transcribing TED talks. The adaptations concern the acoustic and language models and the pronunciation dictionary.

Prior to transcription, the audio documents are partitioned identifying the portions containing speech to be transcribed [2] and associating segment cluster labels, where each segment cluster ideally represents one speaker.

Two types of acoustic features are used. The first are PLP-like [3], with cepstral normalization carried out on a segment-cluster basis [1]. A 3-dimensional pitch feature vector (pitch, Δ and $\Delta\Delta$ pitch) is added to the original PLP one, resulting in a 42-dimension feature vector. The second type are probabilistic features produced by a Multi-Layer Perceptron (MLP) from raw TRAP-DCT features [4], which have been shown to improve system performance when concatenated with cepstral features [5]. The MLP networks were trained using the simplified training scheme proposed in [6] using phone state targets. The feature vector formed by concatenating the MLP, PLP and pitch features has 81 elements.

The acoustic models are gender-independent, tied-state, left-to-right 3-state HMMs with Gaussian mixture observation densities (typically 32 components). The triphone-based phone models are word-independent, but position-dependent. The states are tied by means of a decision tree to reduce model size and increase triphone coverage. The acoustic models are speaker-adaptive (SAT) and Maximum Mutual Information (MMIE) trained.

N-gram language models are obtained by interpolating multiple unpruned component LMs trained on subsets of the training texts and used for both decoding and lattice rescoring. Language model training is performed with LIMSI STK toolkit which allows efficient handling of huge language models without any pruning or cutoff.

Word decoding is carried out in two passes. Each decoding pass produces a word lattice with cross-word, word-position dependent acoustic models, followed by consensus decoding with a 4-gram language model and pronunciation probabilities. The system vocabulary contains 95k words. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR and MLLR [7], and the lattices produced are rescored with a 4-gram back-off

dataset	WER (del., ins.)
dev2010	15.0 (4.0, 3.5)
tst2010	12.7 (3.3, 2.7)

Table 1: Case-insensitive recognition results on the 2010 dev and tst data, scored using sclite.

LM. The first decoding pass is carried out with a modified version of our 2011 Quaero system for broadcast data in English [8, 9] in which a language model trained on the provided ASR texts including the IWSLT14 TED LM transcriptions (3.2M words) was interpolated with the baseline 78k-word language model. The first decoding pass is done in 1xRT. The acoustic models in the first pass were trained on the data distributed in Quaero as well as on data from other sources from previous European or national projects and from the LDC. All acoustic and other language model training data predate December 31, 2010. The Euronews data provided by the organizers was not used. The second pass decoding used the same interpolated language model with acoustic models trained only on 180 hours of transcribed TED talks predating December 31, 2010 to better target the TED data.

The case-insensitive recognition results on the 2010 dev and tst data are given in Table 1 scoring with the NIST sclite scoring using the provided stm and no glm.

3. MT systems: adaptation to speech data

3.1. Machine Translation with N-code

NCODE implements the bilingual n-gram approach to SMT [10, 11, 12] that is closely related to the standard phrase-based approach [13]. In this framework, the translation is divided into two steps. To translate a source sentence $\mathbf{f}$ into a target sentence $\mathbf{e}$, the source sentence is first reordered according to a set of rewriting rules so as to reproduce the target word order. This generates a word lattice containing the most promising source permutations, which is then translated. Since the translation step is monotonic, the peculiarity of this approach is to rely on the n-gram assumption to decompose the joint probability of a sentence pair in a sequence of *bilingual* units called *tuples*.

The best translation is selected by maximizing a linear combination of feature functions using the following inference rule:

$$\mathbf{e}^* = \underset{\mathbf{e},\mathbf{a}}{\operatorname{argmax}} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{f}, \mathbf{e}, \mathbf{a}), \qquad (1)$$

where K feature functions (f_k) are weighted by a set of coefficients (λ_k) and where $\mathbf{a}$ denotes the set of hidden variables corresponding to the reordering and segmentation of the source sentence. Along with the n-gram translation models and target n-gram language models, 13 conventional features are combined: 4 *lexicon models* similar to the ones used in standard phrase-based systems; 6 *lexicalized reordering models* [14, 15] aimed at predicting the orientation of the next translation unit; a "weak" distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model* which compensate for the system preference for short translations. Features are estimated during the training phase. Training source sentences are first reordered so as to match the target word order by unfolding the word alignments [12]. Tuples are then extracted in such a way that a unique segmentation of the bilingual corpus is achieved [11] and n-gram translation models are then estimated over the training corpus composed of tuple sequences made of surface forms or POS tags. Reordering rules are automatically learned during the unfolding procedure and are built using part-of-speech (POS), rather than surface word forms, to increase their generalization power [12].

3.2. MT baseline

This section describes the MT systems trained on written material that served as a benchmark for the succeeding experiments aiming at improving the translation quality for speech transcriptions.

All the parallel corpora used in our translation systems have been preprocessed to remove excessively long sentences as well as sentences with an important length difference between the source and the target. The common preprocessing also included tokenization using the in-house tool described in [16] and word alignments using MGIZA++ [17] and Moses's grow-diag-final-and heuristic for alignment symmetrization.

All the MT systems developed in this study make use of the N-code system described above for translation model training and for decoding. Since the N-code system uses factored models, the training corpora have been tagged with part-of-speech (POS) labels using TreeTagger [18]. The target language model used discriminative log-linear interpolation approach to combine the model trained on TED monolingual data provided by the organizers and the bigger LM trained on WMT data (SRILM [19] toolkit was used for both models).

Our baseline system only uses the training data provided by the IWSLT campaign organizers, composed exclusively of TED talks recordings: we were thus subsequently able to quickly experiment with various adaptation techniques as well as to measure the impact of including large, out-of-domain, corpora.

We performed some additional cleaning on TED corpus, mostly related to extra textual information not present in the audio signal: removing speaker names or initials at the beginning of some lines, removing comments between square brackets and between parentheses, etc. Those notes are added by transcribers in order to facilitate the understanding of the text by human readers, but are useless and even confusing in the context of automatic speech translation.

3.2.1. Impact of the out-of-domain corpora

We tried to improve the performance of the baseline system trained on in-domain data only, by adding various bilingual corpora from the WMT Evaluation Campaign [20]: News-Commentary (NC), Europarl (EPPS) and Gigaword filtered as in [21] (GIGA). All those models were tuned on the same manually transcribed development set (dev2010). As can be seen in Table 2, only the filtered Gigaword corpus actually helped improve the performance of the baseline system. In accordance with these results, we used only this corpus as the additional out-of-domain corpus for our final system.

Table 2: *Baseline MT experiments with written corpora.*

training corpora	BLEU	
	dev2010	test2010
TED	28.8	33.2
TED + NC + EPPS	29.5	33.0
TED + NC + EPPS + GIGA	29.6	34.0
TED + GIGA	29.7	**34.4**

For the sake of speeding up the experiments with the adaptation of the MT systems to the characteristics of the speech data, only the TED corpus was used for training those intermediate systems. Our final system, however, to which the SOUL re-scoring was applied, made use of both TED and the Gigaword data.

3.3. Narrowing the gap between ASR and MT

An important source of MT quality deterioration on ASR output consists in various formatting differences between this output and the written corpora used for the training of the MT engine. One of the promising axes of improving the speech translation quality is therefore to reduce the gap between the ASR output and the source part of the parallel corpora. This goal can be achieved both by post-processing the speech recognition output before translation and by modifying the source part of the corpora used in MT training to make them more alike. In this work, we have experimented with two types of such processing: normalization of numbers and punctuation insertion. Other types of normalization might of cause be considered, such as the normalization of units of measurement, dates, acronyms etc.

3.3.1. Normalization of numbers

One inconsistency between the output of the ASR system and the expected input of the MT system is the fact that the speech recognition system produces the numbers spelled out, whereas MT systems are trained on written texts where numbers are usually written in digits. In both cases, the choice of the approach to number processing is optimal for the corresponding system: a fully spelled representation is closest to the pronunciation (big numbers may correspond to several pronounced words) and is thus convenient for ASR; digital representation is best suited for MT since it is much easier to translate to the equivalent digital representation on the target side. For speech translation, however, the inconsistency in number representations is one obvious source of the translation quality's deterioration. To transform fully spelled numbers in the ASR output into digits, we used a rule-based algorithm provided by LIMSI's ASR system as part of the post-processing to the main recognition system. It must be noted, however, that the numbers in written texts and the numbers produced via the above processing are not always the same. On the one hand, the automatically produced digital forms may contain errors, and on the other hand, human transcriptions are not always consistent and can choose either to spell out or not some of the numbers (*e.g. 1/3rd vs. one-third*). To bring ASR output as close as possible to the expectations of MT, we applied the number transformation to the source side of the TED corpus. In order to do this, we first converted all the digital numbers to text and then re-converted them to digits using the same algorithm as for the post-processed ASR output. A new MT system was then trained based on this corpus (norm).

To evaluate the impact of the number normalization on speech translation, we used the test set provided by the organizers (tst2010), for which we compared the translation performance on manual transcriptions to the performance on the automatic transcriptions produced by our baseline ASR system (WER=17%). Table 3 compares the performance of the baseline system to the performance of the system trained and tuned on normalized corpora. As expected, on the ASR output better results were obtained with normalization. However, the results on the manual transcriptions suffered a small degradation which is most probably due to to the errors produced by the normalization processing.

Table 3: *Experiments with number normalization.*

training corpora	normalization	BLEU (tst2010)	
		auto	manual
TED	no norm	20.5	33.2
	norm	**21.0**	33.0

3.3.2. Punctuation

Speech speech recognition systems do not generally produce punctuation as part of their output. The LIMSI ASR system makes it possible to add punctuation in a post-processing step, but it only includes very basic punctuation marks, such as commas and stop signs. The MT system, on the other hand, is expected to produce fully punctuated text as its output and is typically trained on punctuated sources. The performance on the manually transcribed test data, that does not contain any recognition errors, is nevertheless degraded dramatically if the punctuation is removed from the source side of the test (BLEU=25.5, as compared to BLEU=33.0 for the punctuated test, see Table 3).

Possible solutions to this problem have been explored, for example, in [22]. One solution is to build a new MT system based on the training corpora with unpunctuated source side: the system is thus trained to implicitly insert punctuation as part of the general translation process (implicit punctuation). Another solution is to produce automatic punctuation for the source language and to insert some punctuation marks to speech recognition output before translation (explicit punctuation in source): this approach has the advantage of allowing to keep the MT system unchanged. Our experiments with both approaches are shown in Table 4. We trained a new MT system unpunctuated in source (implicit punct), where we removed all the punctuation marks from the source side of both training corpus (TED) and tuning corpus (dev2010). This unpunctuated system was applied to the normalized ASR output without punctuation in test. The punctuated version of the TED MT system was applied to the same test punctuated by one of our two punctuation systems. Both of these punctuation systems were based on MT techniques and were trained on unpunctuated TED corpus as source and the same corpus with punctuation in target. One system used all the possible punctuations (all), whereas the other only used simple unpaired punctuation: commas, stops, colons, semi-colons, question and exclamation marks (main). The implicit punctuation and as well as the explicit punctuation with main marks achieve equivalent performance on test corpus. The fact that main punctuation insertion yields in better performance than all punctuation insertion can be explained by the fact that the paired punctuation marks (such as quotes or parentheses) are often separated by several words and are therefore much harder to predict correctly in the MT framework. The data sparsity also contributes to the fact that the insertion of all the types of the punctuation may add more errors than correct predictions.

Table 4: *Experiments with punctuation.*

training corpora	punct test	BLEU (tst2010 auto)
TED (implicit punct)	none	**24.4**
TED (man punct)	none	21.0
	auto all	24.0
	auto main	**24.4**

3.4. Adaptation of MT systems to ASR output

In addition to various surface differences between ASR output and MT training corpora such as described above, the most important source of difficulties for speech translation are the errors and the irregularities present in speech recognition output: if the source is degraded, the quality of translation is likely to suffer subsequently. It is to be expected, however, that for some types of errors the translation quality could be improved if the training data for MT included the errors produced by the recognizer, thus allowing for the MT system adapt to the variation in the output of this specific recognizer. This is why we experimented with an extra training corpus (TED auto) obtained by automatic transcription of the speech signal of the talks present in TED training corpus by our baseline ASR system. The corpus thus produced was normalized as described above. Since both punctuated and unpunctuated versions of the manual TED training corpus produced similar results and for the sake of time, we used only the unpunctuated version for these experiments so as to quickly determine the impact of the ASR output in training.

Table 5 compares different configurations for training corpora:

- TED manual transcription only
- TED auto transcription only
- TED manual and TED auto used separately (two translation tables)
- TED manual and TED auto used together (one translation table)

The source side of the development corpus (dev2010) was composed of manual transcriptions for the first model, of automatic transcriptions for the second model and of both automatic and manual transcriptions for the last two models.

Using both corpora produces the best results probably since it allows for the MT system to learn on both correct and erroneous examples. The best performance is achieved with one translation table.

Table 5: *Adaptation to ASR output in MT training.*

training corpora	BLEU (test2010 auto, no punct)
TED man only	24.4
TED auto only	24.2
TED man+auto (2 tables)	24.6
TED man+auto (1 table)	**24.8**

3.5. Final MT system configuration

Based on the results of all the experiments with speech translation described above, for the final systems we used two corpora in training:

- TED man+auto (in one corpus)
- Gigaword (filtered)

Table 6 presents the results for these systems both with and without punctuation in source. The performance of the punctuated system (with ASR data re-punctuated by *punct main*) proved to be slightly better, so this system was used for the final step of the processing: SOUL NNLM and NNTM n-best re-scoring. This table also reports the performance of the final punctuated MT system on the test set transcribed with the final ASR system adapted to TED data (WER=12.8%),

as compared to the same test set transcribed with the baseline ASR system (WER=17%). This shows the impact of the ASR quality on the translation performance. We subsequently used this test set for the experiments with SOUL.

Table 6: *Final MT system performance and the impact of the ASR adaptation to TED data on the MT performance.*

training corpora	punctuation	BLEU (test2010 auto)	
		ASR baseline	ASR final run
TED man+auto (1 table)	no punct	24.8	-
+ GIGA	no punct	25.0	-
	punct main	25.5	**27.7**

3.6. SOUL models

Neural networks, working on top of conventional n-gram back-off language models, have been introduced in [23, 24] as a potential means to improve discrete language models. As in previous submissions in the WMT evaluation (see [25] for instance), we took advantage of the recent proposal of [26]. Using a specific neural network architecture, the *Structured OUtput Layer* (SOUL), it becomes possible to estimate n-gram models that use large vocabulary, thereby making the training of large neural network models feasible both for target language models and for translation models [27]. Moreover, the peculiar parameterization of continuous models allows us to consider longer dependencies than the one used by conventional n-gram models, for instance $n = 10$ instead of $n = 4$.

3.6.1. Description of model structure

SOUL language model is a feed-forward multilayer neural networks estimating word's probability given its context made of the $n-1$ previous words (typically $n = 10$). While this model is similar to neural probabilistic language models introduced in [23], the output layer that predicts the word is organized into a tree structure. This structured output layer allows the model to predict words for large vocabulary applications.

SOUL translation models rely on a specific decomposition of the joint probability $P(\mathbf{f}, \mathbf{e}, \mathbf{a})$ of a sentence pair, where $\mathbf{f}$ is a sequence of I *reordered* source words $(f_1, ..., f_I)$, and $\mathbf{e}$ contains J target words $(e_1, ..., e_J)$, and $\mathbf{a}$ is an alignment between $\mathbf{f}$ and $\mathbf{e}$. In the n-gram approach to SMT [10, 11, 12] this segmentation is a by-product of source reordering, and ultimately derives from initial words and phrase alignments. In this framework, the basic translation units are tuples, which are analogous to phrase pairs, and represent a matching $u = (\overline{f}, \overline{e})$ between a source phrase $\overline{f}$ and a target phrase $\overline{e}$.

The n-gram assumption decomposes the joint probability into the products of tuples' probabilities as follow:

$$P(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \prod_{i=1}^{L} P(u_i | u_{i-1}, ..., u_{i-n+1}) \quad (2)$$

However, as mentioned in [27], this decomposition implies a large vocabulary of bilingual tuples, hence its generalisation capability is limited due to data sparsity issues. As a remedy, the n-gram probabilities in the right-hand side of (2) are factored by first decomposing tuples into source and target parts (or phrases), and then considering each part as a word stream. The decomposition process results in 4 word-factored bilingual models as described in [27], each of which produces a feature score that is added to the final system before SOUL (Section 3.4).

3.6.2. Integration of SOUL *models*

Given the computational cost of computing n-gram probabilities with neural network models, we resorted to a two-pass approach: the first pass uses a conventional system to produce an N-best list (the N most likely hypotheses); in the second pass, probabilities are computed by SOUL models for each hypothesis and added as new features. Then the N-best list is reordered according to a combination of all features including these new features. In our experiments, 10-gram SOUL models were used to rescore 300-best lists. Overall system's log-linear coefficients were optimised using k-best Batch Margin Infused Relaxed Algorithm (KBMIRA) [28] on the automatically transcribed development set.

3.6.3. Training

SOUL models are trained to maximise the likelihood. This optimization is carried out using a mini-batch version of Stochastic Back-propagation (see [24, 26] for more details). However, given the computational cost of each training batch, training corpora are usually resampled at each epoch: instead of performing several epochs over the whole training data, a different small random subset is used at each epoch.

To mitigate the impact of in-domain and out-of-domain corpora, the target language model was trained using for each epoch a set of n-grams of which 75% were sampled from TED data, and the remaining 25% from Gigaword.

SOUL translation models were trained on bilingual tuples constructed from the word alignments of training corpora's sentence pairs. The mixing of training corpora was more complicated as TED corpus contains both manual and automatic transcriptions. In an attempt to narrow the gap between ASR and MT as mentioned in Section 3.3, we used TED auto corpus along with TED manual to train our translation models. To separately evaluate the impact of each corpus, three configurations were tested. The first two consisted in training models on TED manual and TED auto separately. In the third configuration, a mix of TED data (manual and auto concatenated) and Gigaword was used, where 75% of n-grams

Systems	dev	test
Before SOUL	23.7	27.7
Adding all 4 SOUL TMs		
+ TMs TED manual	24.1	27.9
+ TMs TED auto	24.2	28.0
+ TMs mixing TED-GIGA	24.4	27.9
Adding all 4 SOUL TMs and SOUL target LM		
+ TMs TED manual + LM	24.3	27.9
+ TMs TED auto + LM	24.3	27.6
+ TMs mixing TED-GIGA + LM	**24.4**	**28.3**

Table 7: Results of the reranking process with various added feature functions. The first line indicates the result for the best MT system before SOUL. The upper and lower parts of the table show results of adding SOUL TMs and target LM into this system.

used at each epoch were sampled from the former, and 25% from the latter.

Table 7 presents results of adding SOUL features into the best MT system. The performance is evaluated in terms of BLEU scores on the automatically transcribed development and test sets. As shown in the upper part of the table, the models trained on TED auto yield slightly better results than those trained on TED manual. It might be due to the fact that hypotheses in the development and test sets were generated using source sentences automatically transcribed as described previously, and hence are closer to TED auto's bilingual tuples. However, the use of SOUL target language model gave gain only on the configuration trained on the mixed corpora of TED and Gigaword; the best result shown in the last line corresponds to the final system submitted for the evaluation as our primary system.

4. Conclusions

In this paper, we described our submissions for the IWSLT 2014 speech translation task. Our contribution is twofold: first, we investigated different approaches to adapt a standard speech recognition system to TED talks; then the different components of the MT system were improved for a better interaction with ASR output. The MT systems were trained using our in-house translation system (NCODE). We experimented with various techniques for bringing the ASR output data and the expected MT input data format as close as possible. In particular, number normalization and punctuation insertion both allowed to improve the translation quality over the baseline system on ASR data. We also exprimented with various configurations for including the ASR data as part of the MT system so as to adapt this system to the errors and other specific features of the speech recognition output.

Our best submission used both manual and ASR data pooled together for building one translation table. This system was augmented with the integration of continuous space models in a n-best rescoring step. Surprisingly, the gains on the ASR output test data were rather small as compared to the improvement obtained on very similar task for text translation (see [29, 25]). Further analyses are required to better explain these results.

5. Acknowledgements

The authors would like to expresses their gratitude to Jan Niehues for his help and advice in the preparation of this submission.

6. References

[1] J.-L. Gauvain, L. Lamel, and G. Adda, "The LIMSI Broadcast News Transcription System," *SPCOM*, vol. 37, no. 1-2, pp. 89–108, 2002.

[2] ——, "Partitioning and transcription of broadcast news data," *ICSLP*, vol. 98, no. 5, pp. 1335–1338, 1998.

[3] H. Hermansky, "Perceptual linear prediction (PLP) analysis for speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, April 1990.

[4] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, P. Sojka, I. Kopeček, and K. Pala, Eds. Springer Berlin Heidelberg, 2004, vol. 3206, pp. 465–472.

[5] P. Fousek, L. Lamel, and J.-L. Gauvain, "On the use of mlp features for broadcast news transcription," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, no. 5246/2008. Springer Verlag, Berlin/Heidelberg, 2008, pp. 303–310.

[6] Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan, "Using mlp features in SRI's conversational speech recognition system." in *Interspeech*, 2005, pp. 2141–2144.

[7] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[8] L. Lamel, S. Courcinous, J. Despres, J.-L. Gauvain, Y. Josse, K. Kilgour, F. Kraft, V. B. Le, H. Ney, M. Nußbaum-Thom, I. Oparin, T. Schlippe, R. Schlüter, T. Schultz, T. F. da Silva, S. Stüker, M. Sundermeyer, B. Vieru, N. T. Vu, A. Waibel, and C. Woehrling, "Speech Recognition for Machine Translation in Quaero," in *IWSLT*, San Francisco, CA, USA, 2011.

[9] L. Lamel, " Multilingual Speech Processing Activities in Quaero: Application to Multimedia Search in Unstructured Data," in *The Fifth International Conference: Human Language Technologies - The Baltic Perspective*, Tartu, Estonia, October 4-5 2012, pp. 1–8.

[10] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 3, pp. 205–225, 2004.

[11] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà, "N-gram-based Machine Translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[12] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, no. 3, pp. 199–215, 2006.

[13] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *KI-2002: Advances in artificial intelligence*, ser. LNAI, M. Jarke, J. Koehler, and G. Lakemeyer, Eds., vol. 2479. Springer Verlag, 2002, pp. 18–32.

[14] C. Tillmann, "A unigram orientation model for statistical machine translation," in *Proceedings of HLT-NAACL*, 2004, pp. 101–104.

[15] J. M. Crego, F. Yvon, and J. B. Mariño, "N-code: an open-source bilingual N-gram SMT toolkit," *Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 49–58, 2011.

[16] D. Déchelotte, G. Adda, A. Allauzen, H. Bonneau-Maynard, O. Galibert, J.-L. Gauvain, P. Langlais, and F. Yvon, "LIMSI's statistical translation systems for WMT'08," in *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, 2008, pp. 107–110.

[17] 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.

[18] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994, pp. 44–49.

[19] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 2, Denver, USA, 2002, pp. 901–904.

[20] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, MA, June 2014, pp. 12–58.

[21] A. Allauzen, G. Adda, H. Bonneau-Maynard, J. M. Crego, H.-S. Le, A. Max, A. Lardilleux, T. Lavergne, A. Sokolov, G. Wisniewski, and F. Yvon, "LIMSI @ WMT11," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011, pp. 309–315.

[22] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation (IWSLT 2011)*, 2011, pp. 238–245.

[23] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.

[24] H. Schwenk, D. Déchelotte, and J.-L. Gauvain, "Continuous space language models for statistical machine translation," in *Proceedings of the COLING/ACL on Main conference poster sessions*, Morristown, NJ, USA, 2006, pp. 723–730.

[25] A. Allauzen, N. Pécheux, Q. K. Do, M. Dinarelli, T. Lavergne, A. Max, H.-s. Le, and F. Yvon, "LIMSI @ WMT13," in *Proceedings of the Workshkop on Statistical Machine Translation*, Sofia, Bulgaria, 2013, pp. 62–69.

[26] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5524–5527.

[27] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, Montréal, Canada, June 2012, pp. 39–48.

[28] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012, pp. 427–436.

[29] T. Lavergne, A. Allauzen, H. S. Le, and F. Yvon, "LIMSI's experiments in domain adaptation for IWSLT 11," in *International Workshop on Spoken Language Translation, IWSLT*, 2011, pp. 62–67.