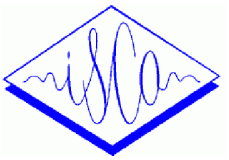
ISCA Archive
http://www.isca-speech.org/archive



International Workshop on
Spoken Language Translation 2011
San Francisco, CA, USA
December 8-9, 2011

LIMSI's experiments in domain adaptation for IWSLT11

Thomas Lavergne, Alexandre Allauzen, Hai-Son Le, François Yvon

LIMSI-CNRS and Université Paris-Sud
BP 133, 91403 Orsay
{lavergne,allauzen,haison,yvon}@limsi.fr

Abstract

LIMSI took part in the IWSLT 2011 TED task in the MT track for English to French using the in-house n-code system, which implements the n-gram based approach to Machine Translation. This framework not only allows to achieve state-of-the-art results for this language pair, but is also appealing due to its conceptual simplicity and its use of well understood statistical language models. Using this approach, we compare several ways to adapt our existing systems and resources to the TED task with mixture of language models and try to provide an analysis of the modest gains obtained by training a log linear combination of in- and out-of-domain models.

1. Introduction

The performance of current Statistical Machine Translation (SMT) systems depends heavily on the data that is used to estimate the various statistical models. As has often been pointed out, good performance can only be obtained if a sufficiently large amount of *in-domain* training data is available, which is not often the case, except for a rather restricter number of domains.

Therefore, improved methods for adapting statistical models using both in-domain and out-of-domain data are actively sought and several proposals have been studied in the literature (see below). The IWSLT'11 "TED" task offers a nice test case for adaptation techniques, since the volume of talk data is, by far, outnumbered by the other sources of data, be they parallel or monolingual.

LIMSI took part in the IWSLT 2011 TED task in the MT track for English to French with the intent to improve our understanding of adaptation techniques for SMT. Our submission is based on the n-gram based approach to Machine Translation [1, 2], a framework in which it is relatively simple to re-implement and compare various adaptation strategies.

Several proposal have been put forward to adapt SMT systems: in the typical situation where a small amount of in-domain data is backed up by larger out-of-domain corpora, various ways to combine the two source of informations can be entertained. The most simple-minded approach is to pool all the available data into one single mixed-domain training corpus; carefully selecting the out-of-domain data based on their similarity with the in-domain texts, at the level of sentences [3], or at the level of phrases however proves to be more effective. Pooling can also be performed directly at the level of models using various mixture modeling strategies [4, 5, 6]. Depending on the available resources, this approach can be applied to the sole language or translation model, or to both models. In the less favorable case where only monolingual data is available, self-training techniques using an out-of-domain SMT system to build an artificial in-domain parallel corpus have also delivered improved performance in several studies [7].

Following the study of [4], we have considered various ways to build mixture models. If all adaptation strategies were indeed useful, a rather paradoxical finding, that was already mentioned in the Foster et al's study, and that we reproduced in various past experiments [8], is that performing an ad-hoc linear combination of models seems to be more effective than tuning the weights of a log-linear model combination with MERT [9]. This finding seems to contradict the findings of [5]. We have found again the same effect, and try to provide some analysis for this unexpected behavior. Another contribution of the paper is an empirical study of adaptation for Neural Network Language models, which was found here to improve the performance of the non-adapted models.

The rest of the paper is organized as follows. In Sections 2 and 3, we describe our decoder, then the various sources of data that have been used to train our baseline systems. Section 4 presents the experimental results achieved during the development period where we contrasted several adaptation policies. We conclude and give further prospects in Section 5.

2. An overview of n-code

Our in-house n-code SMT[1] system implements the bilingual n-gram approach to SMT [1]. Given a source sentence s_1^J, a translation hypothesis $\hat{t}_1^I$ is defined as the sentence which

[1]The latest version of the system used for this evaluation can be downloaded at http://www.limsi.fr/Individu/jmcrego/bincoder/

maximizes a linear combination of feature functions:

$$\hat{t}_1^I = \arg\max_{t_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^J, t_1^I) \right\} \quad (1)$$

where s_1^J and t_1^I respectively denote the source and the target sentences, and λ_m is the weight associated with the feature function h_m. The translation feature is the log-score of the translation model based on bilingual units called *tuples*. The probability assigned to a sentence pair by the translation model is estimated by using the n-gram assumption:

$$p(s_1^J, t_1^I) = \prod_{k=1}^{K} p((s,t)_k | (s,t)_{k-1} \ldots (s,t)_{k-n+1}),$$

where s refers to a source symbol (resp. t for target) and $(s,t)_k$ to the k^{th} tuple of the given bilingual sentence pair. It is worth noticing that, since both languages are linked up in tuples, the context information provided by this translation model is bilingual. In addition to the translation model, *eleven* feature functions are combined: a *target-language model* (see Section 3.2 for details); four *lexicon models*; two *lexicalized reordering models* [10] aiming 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. The four *lexicon models* are similar to the ones used in a standard phrase-based system: two scores correspond to the relative frequencies of the tuples and two lexical weights are estimated from the automatically generated word alignments. The weights associated to feature functions are optimally combined using a discriminative training framework [9] (Minimum Error Rate Training (MERT) using the provided *tst2010* data as development set and *dev2010* as test set.

An interesting feature of the current version of n-code is its ability to consider an arbitrary number of translation and target language models. By default, these models are just added in the log-linear combination, and their weight is adjusted with MERT to the development (and hopefully test) domain.

2.1. Training

In n-code, a translation model is estimated over a training corpus composed of tuple sequences using classical smoothing techniques. Tuples are extracted from a word-aligned corpus (using MGIZA++[2] with default settings) in such a way that a unique segmentation of the bilingual corpus is achieved, allowing to estimate the n-gram model. Figure 1 presents a simple example illustrating the unique tuple segmentation for a given word-aligned pair of sentences (top).

The resulting sequence of tuples *(1)* is further refined to avoid *NULL* words in the source side of the tuples *(2)*. Once the whole bilingual training data is segmented into tuples,

[2] http://geek.kyloo.net/software

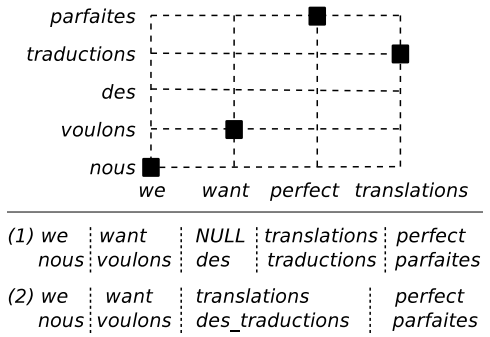


Figure 1: Tuple extraction from a sentence pair.

n-gram language model probabilities can be estimated. In this example, note that the English source words *perfect* and *translations* have been reordered in the final tuple segmentation, while the French target words are kept in their original order.

2.2. Inference

During decoding, source sentences are encoded in the form of word lattices containing the most promising reordering hypotheses, so as to reproduce the word order modifications introduced during the tuple extraction process. Hence, at decoding time, only those encoded reordering hypotheses are translated. Reordering hypotheses are introduced using a set of reordering rules automatically learned from the word alignments.

In the previous example, the rule [*perfect translations* $\rightsquigarrow$ *translations perfect*] produces the swap of the English words that is observed for the French and English pair. Typically, part-of-speech (POS) information is used to increase the generalization power of such rules. Hence, rewriting rules are built using POS rather than surface word forms. Refer to [11] for details on tuple extraction and reordering rules.

3. Baselines

Our baseline system is the one developped for the WMT'2011 evaluation. This system is fully described in [8]. In the rest of this paper, we will refer to the WMT training corpora, except explicit mention to the TED data provided by the IWSLT evaluation campaign.

3.1. Data Pre-processing and Selection

The word alignment model was estimated on all the parallel data provided by WMT. The resulting model was used to carry on a forced alignment of the TED bilingual data. However, the United Nation corpus was discarded during the training of the translation models To train the target language models, we also used all provided data and monolingual corpora released by the LDC for French. Moreover, all parallel corpora were POS-tagged with the TreeTagger [12].

3.1.1. Tokenization

We took advantage of our in-house text processing tools for the tokenization and detokenization steps [13]. Previous experiments have demonstrated that better normalization tools provide better BLEU scores [14]. Thus all systems are built in "true-case."

3.1.2. Filtering the GigaWord Corpus

The available parallel data for English-French includes a large Web corpus, referred to as the *GigaWord* parallel corpus. This corpus is very noisy, and contains large portions that are not useful for translating news text. The first filter aimed at detecting foreign languages based on perplexity and lexical coverage. Then, to select a subset of parallel sentences, trigram LMs were trained for both French and English languages on a subset of the available News data: the French (resp. English) LM was used to rank the French (resp. English) side of the corpus, and only those sentences with perplexity above a given threshold were selected. Finally, the two selected sets were intersected. In the following experiments, the threshold was set to the median or upper quartile value of the perplexity. Therefore, half (or 75%) of this corpus was discarded.

3.2. Target Language Modeling

Neural networks, working on top of conventional n-gram models, have been introduced in [15, 16] as a potential means to improve conventional n-gram language models (LMs). However, probably the major bottleneck with standard NNLMs is the computation of posterior probabilities in the output layer. This layer must contain one unit for each vocabulary word. Such a design makes handling of large vocabularies, consisting of hundreds thousand words, infeasible due to a prohibitive growth in computation time. While recent work proposed to estimate the n-gram distributions only for the most frequent words (short-list) [16], we explored the use of the SOUL (Structured OUtput Layer Neural Network) language model for SMT [17], which allowed us to handle output vocabularies of arbitrary sizes.

Moreover, in our setting, increasing the order of standard n-gram LM did not show any significant improvement. This is mainly due to the data sparsity issue and to the drastic increase in the number of parameters that need to be estimated. With NNLM however, the increase in context length at the input layer results in only a linear growth in complexity in the worst case [16]. Thus, training longer-context neural network models is still feasible, and was found to be very effective in our system.

3.3. Baseline n-gram Back-off Language Models

The baseline French language model was developped for the WMT shared task, and we assumed that the test set consisted off a selection of news texts dating from the end of 2010 to the beginning of 2011. This assumption was based on what was done for the WMT 2010 evaluation. Thus, we built a development corpus in order to optimize the vocabulary and the target language model.

In order to cover different periods, two development sets were used. The first one is *newstest2008*. This corpus is two years older than the targeted time period; therefore, a second development corpus named *dev2010-2011* was collected by randomly sampling bunches of 5 consecutive sentences from the provided news data of 2010 and 2011.

To estimate such large LMs, a vocabulary was first defined for each language by including all tokens observed in the Europarl and News-Commentary corpora. This vocabulary was then expanded with all words that occur more than 5 times in the French-English *GigaWord* corpus, and with the most frequent proper names taken from the monolingual news data of 2010 and 2011. This procedure resulted in a vocabulary containing around 500k words.

All the training data allowed in the constrained task were divided into 7 sets based on dates or genres. On each set, a standard 4-gram LM was estimated from the 500k words vocabulary using absolute discounting interpolated with lower order models [18, 19].

All LMs except the one trained on the news corpora from 2010-2011 were first linearly interpolated. The associated coefficients were estimated so as to minimize the perplexity evaluated on *dev2010-2011*. The resulting LM and the 2010-2011 LM were finaly interpolated with *newstest2008* as development data. This procedure aims to avoid overestimating the weight associated to the 2010-2011 LM.

3.4. The SOUL Model

We give here a brief overview of the SOUL LM; refer to [17] for the complete training procedure. Following the classical work on distributed word representation [20], we assume that the output vocabulary is structured by a clustering tree, where each word belongs to only one class and its associated sub-classes. If w_i denotes the i-th word in a sentence, the sequence $c_{1:D}(w_i) = c_1, \ldots, c_D$ encodes the path for the word w_i in the clustering tree, with D the depth of the tree, $c_d(w_i)$ a class or sub-class assigned to w_i, and $c_D(w_i)$ the leaf associated with w_i (the word itself). The n-gram probability of w_i given its history h can then be estimated as follows using the chain rule:

$$P(w_i|h) = P(c_1(w_i)|h) \prod_{d=2}^{D} P(c_d(w_i)|h, c_{1:d-1})$$

Figure 2 represents the architecture of the NNLM to estimate this distribution, for a tree of depth $D = 3$. The SOUL architecture is the same as for the standard model up to the output layer. The main difference lies in the output structure which involves several layers with a softmax activation function. The first softmax layer *(class layer)* estimates the class probability $P(c_1(w_i)|h)$, while other output *sub-class layers* estimate the sub-class probabilities $P(c_d(w_i)|h, c_{1:d-1})$.

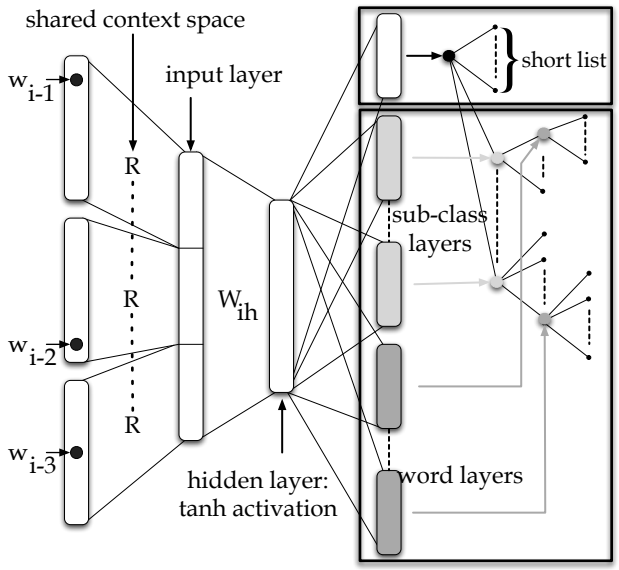


Figure 2: Architecture of the Structured Output Layer Neural Network language model.

Finally, the *word layers* estimate the word probabilities $P(c_D(w_i)|h, c_{1:D-1})$. Words in the short-list are a special case since each of them represents its own class without any sub-classes ($D = 1$ in this case).

4. Developing TED systems

All our systems are two-pass systems, where a first decoding with n-code provides us with lists of n-best hypotheses; these n-best are lists are then reevaluated using continuous space language models. All the language models involved in these systems were subject to domain adaptation. We first discuss experiments with the adaptation of the core models, before presenting adaptation of the SOUL LMs.

4.1. Domain adaptation in n-code

As described in the section 2.1, n-code makes its decisions based on a set of probabilistic scores; three of them play a major role in shaping good translation hypotheses: the source reordering model, the bilingual translation model, and the target language model. Given that our initial attempts at adapting the reordering component have not shown any improvement, we therefore decided to focused on the other two models. For the bilngual and monolingual language models, we compared the three following approaches:

(i) use only small in-domain models built with the sole TED data provided for the IWSLT translation task;

(ii) use the out-of-domain models built for the WMT 2011 evaluation; these models where initially built for News data and the only adaptation was to retune the system on TED development data;

(iii) combine in- and out-of-domain data and/or models.

The first and third lines of Table 1 illustrate the results of the two baselines (i) and (ii): the system built only with the in-domain data and the system built for WMT after retuning. Even if the WMT system was designed for a different task, the joint effect of a large amount of training data and the newly optimized weights seems sufficient to obtain a system that clearly outperforms the small in-domain system.

The strategy (iii) can be implemented in many different ways. For the bilingual model (**BiLM**), we consider two adaptation schemes: simply pool together all the available parallel data and trained a new bilingual model from scratch (ALL), or keep two distinct models and use the ability of n-code to take information from several bilingual models (WMT+TED). This second approach is attractive as it let the discriminative optimization adjust the weight of the in- and out-of-domains models, so as to get the best BLEU score on the development data set. It is also easier to implement, as it does not need to rebuild a new LM, but rather use the existing ones as they are.

For the target language model (**TrgLM**), we also compared two approaches. The first is the linear interpolation approach, where the interpolation coefficients are estimated in order to optimize the perplexity of the development data (BIG). The alternative (discriminative log-linear interpolation) is similar to the method used for the bilingual model and is also denoted (WMT+TED).

The results of all the various tested combinations are in Table 1. All these numbers are obtained with several (at least 4) runs of MERT, so as to minimize the chances of an accidentally good (or poor) run.

BiLM	TrgLM	dev	test
TED	TED	31.4	26.0
TED	BIG	33.8	28.0
WMT	WMT	32.9	26.9
WMT	WMT + TED	33.5	27.3
WMT	BIG	33.2	27.5
WMT + TED	WMT	32.6	27.2
WMT + TED	WMT + TED	33.1	27.4
WMT + TED	BIG	33.3	27.7
ALL	WMT	33.3	28.1
ALL	WMT + TED	33.5	27.4
ALL	BIG	**33.9**	**28.2**

Table 1: Translation results for the different combinations of monolingual and bilingual models. (BLEU)

Both strategies for adapting the monolingual model have succeeded in providing us with modest gains: compare, for instance, the results of the systems with TrgLM=WMT with the ones where TrgLM=WMT+TED: in all situations, an BLEU improvement is observed on the development data. Surprisingly, the linear-combination, which optimizes the perplexity, works slightly better (lines with TrgLM=BIG)

than the log-linear combination, and these improvements carry over on our internal test set.

The adaptation of the bilingual model was also quite successful: both adaptation strategies improve performance over the corresponding baseline system. Small improvements are obtained even with adapted LMs (compare the lines with BiLM=WMT and BiLM=WMT+TED with the adapted target LM BIG). Again, it seems that retraining the model with all the in- and out-domain data is here slightly better (BiLM=BIG) than combining the models directly through MERT. The latter strategy is however much less costly, as training the small TED BiLM is done at almost no cost, compared to the time needed to recompute the ALL bilingual LM from scratch.

The overall conclusion of these experiments are somewhat disappointing, since pooling all the available parallel training data was found to be better than performing a combination at the model level. We do not have a good explanation for this observation: one hypothesis is that our tuning procedure resulted in models that were too close to the development data, and did not generalize on our internal tests; another hypothesis is that the TED model is too small to provide MERT with consistent assessments of translation hypotheses. In any case, complementary analyses are needed to better understand these results.

4.2. Adaptation in SOUL

For these evaluations, our baseline SOUL models are also the ones developed on News data for the WMT'11 evaluation. Details regarding these models are given in [8]. It suffices here to say that these baseline models use a history of 10 words, and that their development involves a combination of four different neural networks, each of which is trained on a randomly selected subpart of the whole French monolingual data. The only adaptation that is performed with these models consists in retuning their weight in the linear combination of weights use to reevaluate the n-best lists.

These models were adapted by run a small number of additional iterations using the entirety of the TED monolingual data. Results are given in Table 2.

BiLM	TrgLM	NN-model	dev	test
WMT+TED	WMT+TED	SOUL	33.24	28.21
WMT+TED	WMT+TED	SOUL+TED	33.74	28.50
ALL	BIG	SOUL	34.51	29.31
ALL	BIG	SOUL+TED	34.79	29.50

Table 2: Reranking results using the SOUL language model, without and with domain adaptation. (BLEU)

As was already observed in several experiments, using the large continuous space language models provides a clear improvement over the baseline: rescoring with SOUL improves the score of approximately one BLEU point. Interestingly, these improvements are attained with translation and target language models that have already been adapted, when the SOUL model has only seen News data. Adapting the SOUL model with in-domain data does even slightly better: compared to the initial WMT baseline, the total accumulated improvement of adaptation is approximately +2.5 bleu points.

Most of the results presented above have been obtained as the result of post-evaluation analyses. Our primary submission for the official TED task uses two separate bilingual models, as well as two separated target language models, and a non-adapted SOUL LM; the corresponding results are reported in [21].

5. Conclusion

In this paper, we presented LIMSI's submission for IWSLT'2011 text translation task. These results were obtained using our in-house n-code system, which implements th n-gram based approach to SMT. One convenient feature of n-code is its ability to handle a arbitrary number of bilingual and target side language models, a facility which makes domain adaptation straightforward: it suffices to incorporate all the available in- and out-of-domain models in the log-linear combination and let the tuning procedure determine the best mixture weights. In particular, models computed for other purposes can be reused as they are, and do not need to be retrained. In this evaluation, this strategy was somewhat sub-optimal, as better results on the internal test data were obtained by pooling the available data and/or models prior to MERT training. These unexpected results may be due to a small mismatch between our development and test conditions. We were much more successful in our use of n-best rescoring with continuous space language models, a strategy that again provided us with clear gains with respect to the baseline; we also showed that these gains are even higher when the continuous space models are also adapted.

6. Acknowledgments

This work was partly funded by OSEO under the Quaero program. The authors wish to acknowledge the help of Josep Maria Crego in the development of the baseline systems.

7. References

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

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

[3] G. Foster, C. Goutte, and R. Kuhn, "Discriminative instance weighting for domain adaptation in statistical machine translation," in *Proc. of the 2010 Conference*

on Empirical Methods in Natural Language Processing, Cambridge, MA, 2010, pp. 451–459.

[4] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007, pp. 128–135.

[5] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*, Morristown, NJ, USA, 2007, pp. 224–227.

[6] G. Sanchis-Trilles and M. Cettolo, "Online language model adaptation via n-gram mixtures for statistical machine translation," in *Proceedings of the Conference of the European Association for Machine Translation*, Saint Raphaël, France, 2010.

[7] H. Schwenk, "Investigations on large-scale lightly-supervised training for statistical machine translation," in *Proceedings of the International Workshop on Spoken Language Translation*, Hawaii, USA, 2008, pp. 182–189.

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

[9] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL '03: Proc. of the 41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, 2003, pp. 160–167.

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

[11] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, no. 3, pp. 199–215, 2007.

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

[13] D. Déchelotte, G. Adda, A. Allauzen, O. Galibert, J.-L. Gauvain, H. Maynard, and F. Yvon, "LIMSI's statistical translation systems for WMT'08," in *Proc. of the NAACL-HTL Statistical Machine Translation Workshop*, Columbus, Ohio, 2008.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL '02: Proc. of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *JMLR*, vol. 3, pp. 1137–1155, 2003.

[16] H. Schwenk, "Continuous space language models," *Computer, Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[17] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, Prague (Czech Republic), 22-27 May 2011.

[18] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP'95*, Detroit, MI, 1995, pp. 181–184.

[19] S. F. Chen and J. T. Goodman, "An empirical study of smoothing techniques for language modeling," Computer Science Group, Harvard University, Tech. Rep. TR-10-98, 1998.

[20] P. Brown, P. de Souza, R. Mercer, V. Della Pietra, and J. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[21] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2011 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.