The UMD Machine Translation Systems at IWSLT 2015

Amittai Axelrod [2], *Ahmed Elgohary* [1], *Marianna Martindale* [3],
Khánh Nguyễn [1], *Xing Niu* [1], *Yogarshi Vyas* [1], *Marine Carpuat* [1,2]

Dept. of Computer Science [1], UMIACS [2] and iSchool [3]
University of Maryland, College Park
marine@cs.umd.edu

Abstract

We describe the University of Maryland machine translation systems submitted to the IWSLT 2015 French-English and Vietnamese-English tasks. We built standard hierarchical phrase-based models, extended in two ways: (1) we applied novel data selection techniques to select relevant information from the large French-English training corpora, and (2) we experimented with neural language models. Our French-English system compares favorably against the organizers' baseline, while the Vietnamese-English one does not, indicating the difficulty of the translation scenario.

1. Introduction

Our goal at the University of Maryland (UMD) for the 2015 IWSLT evaluation campaign was to test our redesigned machine translation (MT) pipeline for different language pairs and data conditions. We selected the French-English and Vietnamese-English tasks, consisting of translating the transcripts of TED talks.[1] The French-English task is a standard one, with a large amount of available data. On the other end of the spectrum, the Vietnamese-English language pair is a scarce-resource scenario and has not yet received much attention in the Machine Translation community. We translated into English in both tracks, so as to have a larger amount of monolingual data available for training neural language models. Our systems all use a standard hierarchical phrase-based architecture, outlined in Section 2. We describe how we used data selection techniques (Sections 4 and 5) to make the most of the available data (Section 3). We also discuss the impact of neural language models (Section 6) on translation output. Official results on the evaluation test set show that our French-English systems outperformed the organizers' baseline by +0.65 to +1 BLEU, while our Vietnamese-English system were -3 BLEU below the public baseline. We discuss these results in Section 7.

2. Core Machine Translation Architecture

We use the `cdec` [1] machine translation toolkit to build hierarchical phrase-based MT systems [2]. We expected the resulting synchronous context-free grammar (SCFG) phrasal rules to be well suited to modeling both the local reorderings arising from translating French into English, as well as the more complex translation rules needed to map Vietnamese – an analytic head-initial language – into English. Training the MT systems was done by following the baseline `cdec` pipeline.[2] Word alignments were generated using `fast_align` [3], and symmetrized using the *grow-diag-final-and* heuristic. The SCFG rules extracted for each test sentence were scored using a small number of dense features, including rule frequency, maximum lexical alignment within the rule, *etc.* We mostly used 4-gram language models, trained using `kenlm` [4], unless stated otherwise. Model weights were tuned using the MIRA algorithm [5] in order to maximize BLEU [6] on held-out test sets.

3. Data Preparation

The 2015 IWSLT campaign released parallel data from both Wikipedia [7] and TED talks.[3] The remaining corpora were obtained from the 2015 Workshop on Machine Translation (WMT '15) task.[4] We translated into English in both of the evaluation tracks we participated in. The English data was all pre-processed the same way: first tokenized with the Europarl tokenizer[5] and then lowercased with the standard `cdec` tool.

3.1. French–English Data

We processed the French data in the same way as the English data, described above, except the tokenization was done with the Moses tokenizer. Table 1 lists the specific sources contained in the 41M parallel French-English training corpus.

3.2. Vietnamese–English Data

The Vietnamese-English translation task is a scarce-resource scenario, with only 0.5% as much training data as the French-English task. Our training corpus included all of the parallel data made available by the organizers, including the auto-

[1] `http://www.ted.com`

[2] `http://www.cdec-decoder.org/guide/tutorial.html`

[3] https://sites.google.com/site/iwsltevaluation2015/data-provided

[4] http://www.statmt.org/wmt15/translation-task.html

[5] http://www.statmt.org/europarl/v7/tools.tgz

Proceedings of the 12th International Workshop on Spoken Language Translation
Da Nang, Vietnam, December 3-4, 2015

Corpus	Segments	Tokens (Fr)	Tokens (En)
Europarl v7	2.0 M	61.9 M	55.7 M
News Commentary	200 k	6.3 M	5.1 M
Common Crawl	3.2 M	91.2 M	81.1 M
Gigaword Fr-En	22.5 M	810.2 M	667.9 M
UN Corpus	12.9 M	421.7 M	361.9 M
Wikipedia	403 k	9.8 M	11.3 M
TED corpus	207 k	4.5 M	4.2 M
Total	41.5 M	1.406 B	1.187 B

Table 1: French-English Parallel Training Data

matically extracted Wikipedia corpus [7]. This was done to increase vocabulary coverage, despite the domain mismatch of the Wikipedia data with respect to the TED task. The size of each corpus is shown in Table 2.

Corpus	Segments	Tokens (Vi)	Tokens (En)
TED corpus	130.9k	3.2M	2.6M
Wiki	58.1k	662.2k	661k
Total	189k	3.86M	3.29M

Table 2: Vietnamese-English Parallel Training Data

The processing of the Vietnamese side was minimal: we simply tokenized it as if it were English and removed any uppercasing to normalize borrowed foreign words. We experimented with off-the-shelf chunking tools for Vietnamese, but found that they did not help translation quality. The vn-Tokenizer [8] tool takes a hybrid approach that combines finite-state automata, regular expressions, and a maximal-matching strategy. However, it proved too slow to process our training data. We also tried the CRFChunker from the JVnSegmenter software [9], which frames chunking as a supervised sequence labeling problem. This tool comes with a model trained on a small set of 8,000 hand-labeled Vietnamese sentences. Unfortunately, using the CRFChunker to preprocess Vietnamese degrades translation quality by about -0.6 BLEU, possibly due to a domain mismatch.

Choosing not to chunk the Vietnamese text differs from standard practice in related translation tasks. In Chinese-English translation, for example, Chinese word segmentation is a key step of the preprocessing pipeline (with the exception of substring or character-based MT models, as in [10]). However, prior work suggests that defining Chinese word boundaries independently of the translation process is not optimal [11, 12]. Based on this, it seems reasonable to let word alignment patterns define translation-driven Vietnamese phrases.

3.3. Postprocessing

Our translation system used tokenized and un-cased data internally. As such, our MT output required the post-processing steps of re-casing and then de-tokenizing before submission. Recasing aims to restore the capitalization that was lost when normalizing case during preprocessing. We used the `Moses` recaser tool.[6] This tool frames recasing as a monotone translation task from un-cased English into cased English. The tool runs Moses without reordering, using a word-to-word translation model and a cased language model. We trained the recaser language model on the English side of the parallel training corpora in Tables 1 and 2. We detokenized the re-cased output using the rule-based detokenizer tool[7] from Moses [13]. We extended this script to support additional special characters that caused the decoder to crash.

4. Training Data Selection

We faced two problems when building the French-English system. The training process was computationally expensive because of the large amount of parallel training data (41M segments). Additionally, the vast majority of the parallel segments are drawn from various domains and genres that are very different from TED. Table 1 shows that TED talks represent only 0.5% of the parallel segments. We addressed both issues by using data selection to determine the most TED-like subset of the parallel corpus. This pseudo in-domain subset was then used to augment the TED data. This approach yielded a medium-scale training setting, easily handled by our standard MT pipeline on ordinary-sized computers.

4.1. Data Selection Techniques

We compared two data selection techniques in the French-English track. The first was the popular cross-entropy difference or "Moore-Lewis" method from [14], which we refer to as `xediff` for short. The second one was recently proposed [15] and uses a hybrid word/part-of-speech text representation to distinguish between rare and frequent events.

4.1.1. Cross-Entropy Difference

The Moore-Lewis method relies on cross-entropy difference to produce domain-specific systems that are usually as good as or better than systems using all available training data [16]. To implement Moore-Lewis selection, we first trained an in-domain language model (LM) on the in-domain TED data, and another LM on the full pool of general data. The algorithm uses these language models to assign a *cross-entropy difference score* to each data-pool sentence.

Lower scores for cross-entropy difference indicate more relevant sentences, namely those that are *most like* the target domain and *most unlike* the full pool average. After ranking the data pool sentences by this score, the top-n sentences (or sentence pairs) are used to create the desired subset of most-relevant sentences. In this work, we added these sentences to the in-domain corpus and trained MT systems on the combined corpus. A range of values for n is typically considered, selecting the n that performs best on held-out

[6] `http://www.statmt.org/moses/?n=Moses.SupportTools`

[7] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl`

in-domain data. The size of these domain-specific systems scales roughly linearly with the amount of selected data: a system trained on the most domain-relevant 10% of the full out-of-domain dataset will be roughly one-tenth of the size of a system trained using all the available data.

4.1.2. Hybrid Word/POS Representation

The data selection technique from [15] uses a hybrid word/part-of-speech representation for corpora in order to distinguish between rare and frequent events. In some sense, this newer method is a pre-processing step before performing the above-described cross-entropy difference data selection method. This pre-processing step changes the representation of the corpus into something better suited for computing the relevance score for each sentence. After the sentence scoring and corpus re-ranking is done, the original words are put back and the downstream LM or MT system is trained as usual. This method does not have a standard name yet, so in this work we refer to it as `min10` or `new`.

This newer hybrid word/POS data selection aims to improve scaling of the data selection process itself and to improve the vocabulary coverage of the selected data. This is achieved by constructing a hybrid representation of the text that abstracts away words that are infrequent in either of the in-domain and general corpora. The threshold used to determine "infrequent" is a minimum count of 10 in each of the task and pool corpora, but other values could be explored. All words that do not meet this criterion are replaced with their part-of-speech (POS) tags, permitting their n-gram statistics to be robustly aggregated when the task and pool language models are built.

The intuition for abstracting away rare words is that if a domain-relevant sentence includes a rare word in some non-rare context (e.g. "An earthquake in Port-au-Prince"), then another sentence with the same context but a *different* rare word is probably also just as relevant (e.g. "An earthquake in Kodari"). Suppose "Kodari" is an out-of-vocabulary word with respect to the task corpus, and that "Port-au-Prince" appears three times in each corpus. The cross-entropy difference method would reward the first sentence because it knows "Port-au-Prince", but penalize the second sentence because "Kodari" is unknown. The new method would also reward the first sentence, because it has seen "An earthquake in NPP" a few times. The new method would *also* reward the second sentence, for exactly the same reason.

After the corpus has been transformed, the Moore-Lewis data selection algorithm is then used to select parallel segments on the hybrid corpus representation, the data pool is re-sorted by this score, and then the hybrid corpus representation is discarded and the original representations of the selected segments (the regular sentence forms) are then used to train MT systems.

Recent experiments on medium-scale Chinese-English Machine Translation tasks [15] showed that this hybrid method can substantially improve lexical coverage, reduce computational requirements for the data selection model itself, and improve translation quality when compared against the standard approaches of [14] and [16].

4.2. Training Data Selection Results

Each of the two data selection methods tested for the French-English task has three possible instantiations: as a monolingual method on the input side (French), as a monolingual method on the output side (English), or as a bilingual method that combines both the French and English monolingual scores. In each of the six cases, we selected relevant subsets of the data pool and concatenated each of them with the in-domain TED training data when training the downstream MT system. We used `cdec` to train these downstream systems for extrinsic evaluation.

For consistency, we used the KenLM toolkit [4] to build all language models used for the data selection experiments. All of them were 4-gram LMs. To enable fair comparisons, all of the word-based models had vocabularies fixed to: {TED} $\cup$ {Pool minus singletons}. In constructing our hybrid word/POS representations for the new method, we used the Stanford part-of-speech tagger [17] to generate the POS tags for each of the languages.

The amount of data selected for each method was determined empirically by training MT systems on the selected slices and comparing the BLEU scores on the `tst2012` and `tst2013` held-out sets. We tested all three conditions for each of the two methods, though here we present only results from using the monolingual English version of the cross-entropy difference and the new hybrid methods. The monolingual English results are shown in Figure 1. The new method provides significantly better coverage of the words in the in-domain corpus than the Moore-Lewis method, and at least as good MT performance. Though the new method's BLEU scores are slightly better, the difference is not enough to be particularly important.

For this submission, the best performance with the standard cross-entropy difference method was with 3 million selected sentences. With the new hybrid word/POS method, selecting 4 million sentences out of the 41 M in the data pool. More results, graphs, and detailed analysis comparing the two methods can be found in [18]. The results from the monolingual French and bilingual scoring methods followed the same trend as the monolingual English scores, but were overall slightly lower.

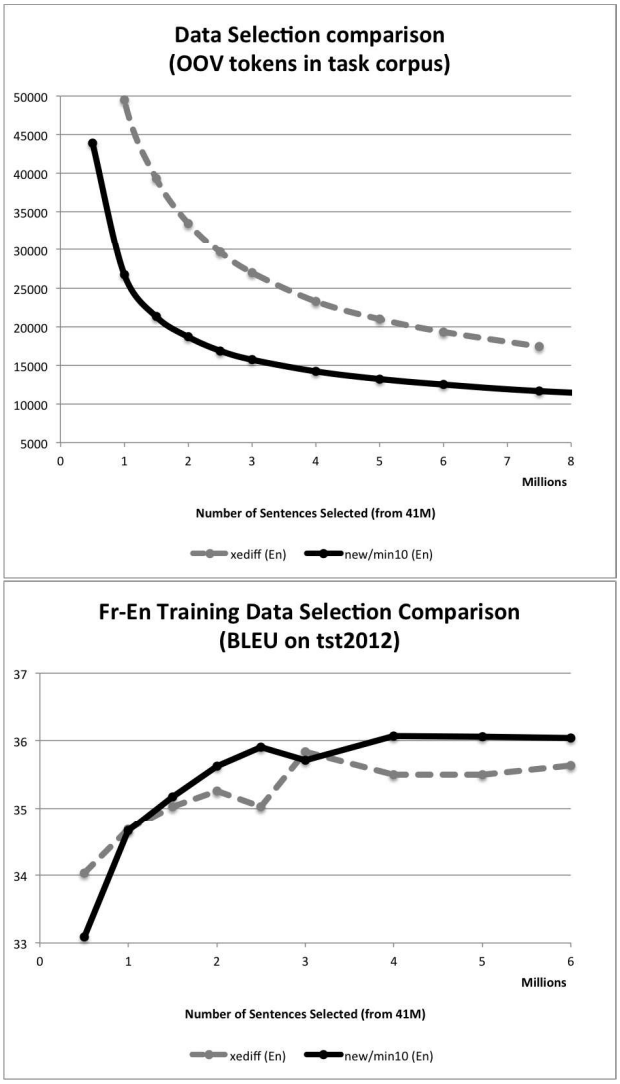

Figure 1: Comparison of the two monolingual English-side data selection methods: Moore-Lewis (grey dashed) and the new hybrid word/POS (solid black): OOV tokens in the TED training set (top), and BLEU scores on `tst2012` (bottom).

Method	BLEU (tst2013)
baseline	37.82
+`xediff` (3 M)	38.29
+`min10` (4 M)	**38.54**

Table 3: Expanding the training set using data selection improves Fr-En translation quality.

After determining the best amount of data to select with each method, we evaluated whether these selected subsets were helpful for translating in-domain test sets. These results are shown in Table 3. The baseline system used the in-domain data in the MT pipeline described in Section 2, and was tuned on the large development set defined below, in Section 5. Table 3 shows that both data selection techniques improve the BLEU score of the translation output. The newer hybrid word/POS method from [15] yielded the largest improvement (+0.7 BLEU), and was therefore used as the training set for our French-English submissions.

5. Tuning Data Selection

Since selecting training data improves translation quality, we hypothesized that similar techniques could also be used to construct better tuning sets. Prior work shows that choosing a good development test set to tune the MT log-linear model parameters is crucial to performance [19, 20]. The IWSLT organizers provided a large number of development test sets for tuning and development purposes (Tables 4 and 5). As a result, we had many options for defining the tuning and tests sets for our experiments.

Corpus	# segments	# fr tokens	# en tokens
dev2010	887	20214	20214
tst2010	1664	33846	31979
tst2011	818	15628	14498
tst2012	1124	23460	21473
tst2013	1026	23293	21706

Table 4: French-English Development Test Sets

Corpus	# segments	# vi tokens	# en tokens
dev2010	769	20750	17410
tst2010	1342	35320	28317
tst2011	1435	32801	26887
tst2012	1553	34292	27983
tst2013	1268	33682	26728

Table 5: Vietnamese-English Development Test Sets

We made the assumption that the most recent test sets would be closest to this year's evaluation data, and therefore used the `tst2013` test set to evaluate translation quality during system development. We proposed two ways to make use of the remaining data at tuning time: First by increasing the number of tuning examples, and secondly by ranking the tuning set and ordering the examples from easiest to hardest.

The development test sets could be used differently: for instance, we could have used several held-out test sets to guide system development.Given our focus on data selection, we decided instead to build a large tuning set by concatenating all development test sets, aside from `tst2013`. As shown in Table 6, this simple strategy yielded a +0.8 BLEU improvement for the Vietnamese-English task, and a +0.75 improvement for the French-English task.

Next, we investigated the impact of ranking the tuning examples. The order in which tuning examples are seen has an impact on learning, because we tune parameters using the *online* MIRA algorithm [21]. Instead of using the natural order of sentences in the original documents, we hypothesized that presenting "easy" examples before "hard" examples might help learning, as in curriculum learning [22].

Task	Tuning set	BLEU
Vi-En	dev2010	23.52
Vi-En	dev2010+tst2010+tst2011+tst2012	**24.30**
Fr-En	dev2010	36.43
Fr-En	dev2010+tst2010+tst2011+tst2012	**37.19**

Table 6: Impact of expanding tuning set on translation quality (train = TED, test set = `tst2013`)

We defined "easier" and "harder" to mean the tuning sentences were more (and less, respectively) similar to the parallel training data. We used the in-domain language model perplexity as a similarity score over sentences. We trained 4-gram models with modified Kneser-Ney smoothing [23] using `kenLM` [4] on the source side of the in-domain TED training data. We then ranked the tuning examples by increasing perplexity. Tables 7 shows that this approach yielded further improvements in translation scores, at least for French-English (+0.6 BLEU), though it had no effect on Vietnamese-English (+0.01 BLEU). This suggests that the order of tuning examples can impact translation quality, but is not guaranteed. However, it is not clear how to best rank examples, and we will investigate alternate ranking criteria (including random order) and re-sampling strategies in future work.

We use the best performing strategy in the final system, and tuned on the concatenation of examples from `dev2010` to `tst2012`, ranked by perplexity.

Task	Tuning set	Order	BLEU
Vi-En	dev2010+tst2010-2012	default	24.30
Vi-En	dev2010+tst2010-2012	ranked	24.31
Fr-En	dev2010+tst2010-2012	default	37.19
Fr-En	dev2010+tst2010-2012	ranked	**37.82**

Table 7: Impact on translation quality of ranking tuning examples by increasing perplexity, for a system trained on the in-domain (TED) data and evaluated on `tst2013`.

6. Neural Language Models

Based on recent promising results[24], neural language models (NLMs) [25, 26] have become standard MT system components. NLMs are typically trained by jointly learning word embeddings and an estimator for the probabilities of words conditioned on their preceding history. We used the Oxford Neural Language Modeling Toolkit (`OxLm`) [27], which implements two useful approximations that can significantly reduce the training and testing time. The first approximation is a class-based factorization to word conditional probabilities where classes are obtained by applying Brown clustering [28] to the vocabulary of the training data. In our experiments, we set the number of clusters to the recommended value of $3\sqrt{|V|}$, where $|V|$ is the vocabulary size. Second, `OxLm` provides an implementation of a noise contrastive estimation (NCE) training algorithm [26] which was shown to dramatically reduce the training time with only a minor reduction to the end-to-end BLEU scores.

We trained two kinds of neural language models on datasets of different scale. The first type (labelled `NlmSmall`) was trained on a small amount of data, with a class-based factorized OxLM using minibatch stochastic gradient descent. The training set consisted of the English side of the in-domain parallel data, described in Section 3. The second set of models (labelled `NlmLarge`) were trained on much larger data sets. These larger corpora were constructed by augmenting the training set from `NlmSmall` wiht subsets of the large pool of permissible monolingual English corpora.[8] We used the `xediff` method described in section 4 to select the 2.5M, 5M, and 7.5M samples from the monolingual pool that were most similar to the training set of `NlmSmall`. We trained three class-based factorized OxLms, one for the concatenation of each selected subset with the `NlmSmall` training corpus. These models are labelled `NlmLarge2.5m`, `NlmLarge5m` and `NlmLarge7.5m` in Table 8. We used the NCE-based algorithm to speed up the training of the three large models.

Neural LM Model	Hyperparameters	Vi-En BLEU
None (baseline)	N/A	24.23
`NlmSmall`	l:100,h:8,f:15,λ:1	25.23
`NlmLarge2.5m`	l:100,h:6,f:20,λ:2	25.43
`NlmLarge5m`	l:100,h:6,f:20,λ:1	25.29
`NlmLarge7.5m`	l:100,h:6,f:20,λ:1	**25.48**

Table 8: The best hyperparameters and the corresponding BLEU scores of the Vietnamese-English pipeline of each of our neural language models.

We fine-tuned the hyperparameters of our language models based on the devset perplexity of each hyperparameter setting. We considered all combinations of the following values of four hyperparameters: (1) dimension of word embeddings $l = \{50, 100, 200, 300\}$, (2) history length (order) that the model conditions on $h = \{4, 5, 6, 7, 8\}$, (3) frequency cutoff (the frequency threshold below which a word is considered unknown) $f = \{5, 10, 15, 20\}$, and (4) training regularization parameter $\lambda = \{0.01, 0.1, 1, 2, 5\}$. We noticed that setting l to 200 or 300 hurt the training and testing times significantly without introducing much benefit to the perplexity scores. Table 8 shows the final hyperparameters learned.

Finally, we evaluated the impact of the neural language models on the output scores of our Vietnamese-English system. All models improved the BLEU score. The largest improvement (+1.2) was obtained with `NlmLarge7.5m`, which we included in our final Vietnamese-English submission. For the French-English system, we used `NlmSmall`.

[8] `https://sites.google.com/site/iwsltevaluation2015/data-provided`

7. Conclusion

We have described the UMD systems submitted to the IWSLT 2015 evaluation campaign. Official results on the evaluation data are provided in Table 9. This table contains scores on the cased, detokenized, output, unlike our internal experimental results in Sections 4, 5, and 6.[9]

System	vi-en	fr-en (2014)	fr-en (2015)
Primary submission	21.57	33.20	32.59
Organizers' baseline	24.61	32.22	31.94

Table 9: Results on evaluation test sets; BLEU scores are computed on cased, untokenized data, using the official IWSLT evaluation server.

The French-English system outperformed the organizers' baseline by approximately +1 BLEU on the 2014 progress test set, and +0.6 on the 2015 test set. This reiterates the benefits of data selection. It is worth noting that these results were obtained using a single n-gram English language model, trained only on the English side of the parallel corpus.

The Vietnamese-English system performed significantly worse than the baseline. This might be due to the lack of preprocessing on the Vietnamese side: as the Vietnamese text was not segmented, the source context captured in SCFG rules was very narrow. In addition, the English n-gram model was trained only on the English side of the parallel data. This can be problematic in a low-resource task such as Vietnamese-English. After the official evaluation period, we augmented our system with 4-gram language models trained on the monolingual English corpus used for neural language modeling. As expected, this approach improved translation quality: we obtained improvements of up to +2 BLEU points on the development test sets.

Overall, our experiments showed that using a standard MT architecture and focusing on parallel data selection for the task at hand is a simple but effective strategy for building MT systems. We will turn our attention to monolingual English data in future work.

8. References

[1] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blumson, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A Decoder, Alignment, and Learning Framework for Finite-State and Context-Free Translation Models," *ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions*, 2010.

[2] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[3] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," *NAACL (North American Association for Computational Linguistics)*, 2013.

[4] K. Heafield, "KenLM : Faster and Smaller Language Model Queries," *WMT (Workshop on Statistical Machine Translation)*, 2011.

[5] D. Chiang, "Hope and Fear for Discriminative Training of Statistical Translation Models," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1159–1187, Apr. 2012.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," *ACL (Association for Computational Linguistics)*, 2002.

[7] K. Wołk and K. Marasek, "Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs," *Procedia Technology*, vol. 18, pp. 126–132, 2014.

[8] N. T. M. Huyên, A. Roussanaly, H. T. Vinh *et al.*, "A hybrid approach to word segmentation of Vietnamese texts," in *Language and Automata Theory and Applications*. Springer, 2008, pp. 240–249.

[9] C.-T. Nguyen and X.-H. Phan, "JVnSegmenter: A Java-based Vietnamese Word Segmentation Tool," 2007. [Online]. Available: http://jvnsegmenter.sourceforge.net

[10] G. Neubig, T. Watanabe, S. Mori, and T. Kawahara, "Machine Translation without Words through Substring Alignment," in *ACL (Association for Computational Linguistics)*, 2012.

[11] D. Wu, "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–404, 1997.

[12] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese Word Segmentation for Machine Translation Performance," in *WMT (Workshop on Statistical Machine Translation)*, 2008.

[13] P. Koehn, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, C. Moran, C. Dyer, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," *ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions*, 2007.

[14] R. C. Moore and W. D. Lewis, "Intelligent Selection of Language Model Training Data," *ACL (Association for Computational Linguistics)*, 2010.

[9] Our internal BLEU scores were all computed using an internal scorer on uncased, tokenized, text.

[15] A. Axelrod, P. Resnik, X. He, and M. Ostendorf, "Data Selection With Fewer Words," *WMT (Workshop on Statistical Machine Translation)*, 2015.

[16] A. Axelrod, X. He, and J. Gao, "Domain Adaptation Via Pseudo In-Domain Data Selection," *EMNLP (Empirical Methods in Natural Language Processing)*, 2011.

[17] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," *NAACL (North American Association for Computational Linguistics)*, 2003.

[18] A. Axelrod, Y. Vyas, M. Martindale, and M. Carpuat, "Class-Based N-gram Language Difference Models for Data Selection," *IWSLT (International Workshop on Spoken Language Translation)*, 2015.

[19] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. F. Zaidan, and C. Callison-Burch, "Machine translation of Arabic dialects," in *NAACL (North American Association for Computational Linguistics*, 2012.

[20] A. Matthews, W. Ammar, A. Bhatia, W. Feely, G. Hanneman, E. Schlinger, S. Swayamdipta, Y. Tsvetkov, A. Lavie, and C. Dyer, "The CMU machine translation systems at WMT 2014," in *WMT (Workshop on Statistical Machine Translation)*, 2014.

[21] D. Chiang, "Hope and fear for discriminative training of statistical translation models," *Journal of Machine Learning Research*, vol. 13, pp. 1159–1187, 2012.

[22] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," *ICML (International Conference on Machine Learning)*, pp. 41–48, 2009.

[23] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, oct 1999.

[24] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," *ACL (Association for Computational Linguistics)*, 2014.

[25] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[26] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *ICML (Internation Conference on Machine Learning)*, 2012.

[27] P. Baltescu, P. Blunsom, and H. Hoang, "OxLM: A neural language modelling framework for machine translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 102, no. 1, pp. 81–92, 2014.

[28] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based N-gram Models of Natural Language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.