Parameter Optimization for Statistical Machine Translation: It Pays to Learn from Hard Examples

Preslav Nakov, Fahad Al Obaidli, Francisco Guzmán and Stephan Vogel
Qatar Computing Research Institute, Qatar Foundation
Tornado Tower, floor 10, PO box 5825
Doha, Qatar
{pnakov,faalobaidli,fherrera,svogel}@qf.org.qa

Abstract

Research on statistical machine translation has focused on particular translation directions, typically with English as the target language, e.g., from Arabic to English. When we reverse the translation direction, the multiple reference translations turn into multiple possible inputs, which offers both challenges and opportunities. We propose and evaluate several strategies for making use of these multiple inputs: (a) select one of the datasets, (b) select the best input for each sentence, and (c) synthesize an input for each sentence by fusing the available inputs. Surprisingly, we find out that it is best to tune on the hardest available input, not on the one that yields the highest BLEU score. This finding has implications on how to pick good translators and how to select useful data for parameter optimization in SMT.

1 Introduction

Nowadays, statistical machine translation (SMT) systems are data-driven, and thus critically depend on the available resources for training, tuning and evaluation. These resources are hard to obtain, which has limited research to a small number of language pairs for which biligual sentence-aligned parallel corpora, called *bitexts*, are available.

What is often not realized is that SMT research has further been restricted to only some translation directions, e.g., those of interest to evaluation campaigns such as NIST and IWSLT or to funding agencies such as DARPA. This is because stable SMT evaluation requires multiple reference translations for the target language. Such multiple references are often available for the English (target) side of the tuning and the evaluation dataset, but not for the source language, e.g., Arabic, Chinese.

Reversing the translation direction yields (i) a single reference translation and (ii) multiple versions for each tuning/testing input sentence. There is little we can do about (i),[1] but (ii) offers interesting opportunities for tuning and evaluation.

Below we focus on the question of how to make best use of the multiple available inputs at *tuning* time. We propose and evaluate several strategies for making use of these multiple inputs: (a) select one of the datasets, (b) select the best input for each sentence, and (c) synthesize an input for each sentence by fusing the available inputs.

2 Related Work

One relevant line of research is on *multi-source translation*, which generates a single translation given multiple versions of the input. This line was started by Och and Ney (2001), who translated the different inputs in isolation and then selected one of them. It has been further extended with various strategies for generating a consensus translation by combining either the inputs (Schroeder et al., 2009) or the outputs (Matusov et al., 2006) of the SMT system. In contrast, we assume having multiple sources at tuning but not at testing time.

A related line focused on *data selection*. For *training* data, this includes filtering (Moore and Lewis, 2010; Foster et al., 2010), instance-weighting (Axelrod et al., 2011; Matsoukas et al., 2009) and model adaptation (Hildebrand et al., 2005). For *tuning* data, Liu et al. (2012) built a separate tuning dataset for each test sentence, which is too costly for real-world translation.

To the best of our knowledge, ours is the first attempt to make best use at *tuning* time of multiple input versions of the same tuning sentence and a single reference translation for it. Previous English–Arabic SMT has used the first input (Al-Haj and Lavie, 2012; Kholy and Habash, 2012).

[1] One could hire translators, but this would be costly.

3 Method

3.1 Choosing a dataset

We can select one of the input datasets.

Select-first. One possible baseline is to select randomly, e.g., the dataset that is listed first.

Concat-all. Another baseline is to concatenate all tuning datasets: using each of the available English versions of a given sentence as input, each paired with the only available Arabic reference.

Then, there are a number of strategies that select the dataset yielding the highest BLEU score:

Backtranslate. We can backtranslate the single target-language reference to English, then evaluate this translation with respect to each of the English inputs, and select the one yielding the highest BLEU score. We can do this using our own system, trained in the opposite, X-English direction; this makes the results potentially more relevant to a system trained and tuned on the same dataset, but in the English-X direction. Another option is to use Google Translate, which would avoid the bias to our datasets. One could argue in favor of either option, and we experiment with both.

X-vs-all-but-X. Here we pretend that one of the English inputs is in fact a translation, and we evaluate this "translation" with respect to the remaining English datasets. We calculate the BLEU score for each of the English datasets using the remaining English datasets as references, and we select the one with the highest BLEU. This minimizes the risk of selecting an outlier dataset for tuning.

Best-on-tuning. Given an English input, we use it to tune the parameters of our SMT system, then we use these learned parameters to translate each of the English inputs, and we evaluate them using BLEU. Then, we average the BLEU scores, where the averaging is over (a) the translations of all English inputs or (b) all but the one used for tuning. The rationale behind (a) is to make all BLEU scores comparable, while that for (b) is to clearly separate tuning from testing, i.e., not to test on the particular dataset that was used for tuning. In either case, we select the dataset that achieved the highest such average.

3.2 Synthesizing a dataset from full sentences

Instead of selecting an *entire* input dataset, we can *synthesize* a new dataset by fusing the available inputs. The easiest way is to do selection at the sentence-level: for each tuning reference sentence, we can select one of the available English inputs.

We will do the selection with respect to some English reference, e.g., backtranslation of the Arabic reference generated by our own system or by Google translate. Below, we present the similarity measures that we use for the selection.

BLEU+1 (B1). BLEU+1 (Lin and Och, 2004) is a smoothed version of BLEU (Papineni et al., 2002) used to address sparseness problems with n-gram matches when comparing sentences.

BLEU+1 BP smooth (B1-BP). The BLEU+1 approximation of BLEU smooths the n-gram counts but not the brevity penalty, thus destroying the balance between the two; it also assigns a non-zero precision to cases with zero matches. Thus, we experiment with a version of BLEU+1 from (Nakov et al., 2012) that smooths the brevity penalty and also uses a "grounding" factor.

BLEU+1 Sigmoid LP (B1-SG). Note that the brevity penalty of BLEU/BLEU+1 penalizes shorter but not longer sentences. Thus, we also experiment with a version of BLEU+1 with a symmetric length penalty, which penalizes the squared differences in length using a sigmoid function:

$$LP(s_i, r) = 3 - 4 * \texttt{sig}\left(\left[\frac{l(s_i) - l(r)}{\alpha}\right]^2\right)$$

where $l(s_i)$ and $l(r)$ are the length of the i-th input and of the reference, respectively, and α is a tolerance factor (set to 5 in our experiments).

Length Difference (DL). We also try to minimize the difference in length.

Minimum BLEU+1 (MIN-B1). Next, instead of maximizing BLEU+1, we can minimize it, i.e., pick the hardest input sentence, and tune the SMT system to perform well on such hard input.

Minimum Length (MIN-L). Finally, we can just pick the shortest sentence.

3.3 Synthesizing a dataset by fusing sentences

MEMT. Instead of selecting one of the possible inputs, we can synthesize a new input by mixing different inputs at the *sub-sentence* level. Here, we use the Multi-Engine Machine Translation system, or MEMT, (Heafield and Lavie, 2010) to merge different input sentences. It merges all input sentences into a lattice and then extracts a new candidate from that lattice using features such as length, language model, and n-gram matches; it tries to maximize BLEU with respect to a given reference: again, a backtranslation of the reference to English using own SMT system or Google Translate.

TEST ⇒	MT050		MT051		MT052		MT053		MT054		AVERAGE	
TUNE ⇓	BLEU	len	BLEU	len	BLEU	len	BLEU	len	BLEU	len	BLEU	len
MT040	**34.63**	0.984	**30.96**	0.984	**29.73**	0.973	~~40.40~~	1.014	**35.46**	0.988	**34.24**	**0.989**
MT041	34.37	0.969	30.59	0.966	29.44	0.954	**40.91**	0.999	35.31	0.972	34.12	0.972
MT042	34.34	0.967	30.57	0.964	29.08	0.952	40.64	0.998	35.12	0.970	33.95	0.970
MT043	33.99	0.957	30.23	0.952	29.06	0.943	40.62	0.988	~~34.81~~	0.960	33.74	0.960
MT044	~~33.87~~	0.961	~~30.18~~	0.957	~~28.96~~	0.947	40.51	0.992	34.82	0.965	~~33.67~~	0.964
MT04ALL	34.37	0.970	30.49	0.967	29.42	0.957	40.72	1.001	35.15	0.973	34.03	0.974
best−worst	0.76		0.78		0.77		0.51		0.65		0.57	

Table 1: **Tuning on MT04 and testing on MT05.** Shown are BLEU scores and hypothesis/reference length ratios. The best and the worst BLEU scores for each test MT05 dataset are in **bold** and ~~stroke out~~, respectively; the last row shows the absolute difference between them.

4 Experiments and Evaluation

4.1 Experimental Setup

We used the phrase-based SMT model (Koehn et al., 2003), as implemented in the Moses toolkit (Koehn et al., 2007), to train an SMT system translating from English to Arabic.

For tuning and evaluation, we used two multi-reference datasets, MT04 and MT05, from the NIST 2012 OpenMT Evaluation,[2] each with a single Arabic input and five English reference translations, which we inverted, ending up with five English inputs and one Arabic reference for each one.

We trained the English-Arabic system (translation, reordering, and language models) on all training data from NIST 2012 except for UN data. Following Kholy and Habash (2012), we normalized the Arabic training, development and test data using MADA (Roth et al., 2008), fixing automatically all wrong instances of *alef*, *ta marbuta* and *alef maqsura*. We segmented the Arabic words by splitting out conjunctions (MADA scheme D1). For English, we converted all words to lowercase.

We built our phrase tables using the standard Moses pipeline with max-phrase-length 7 and Kneser-Ney smoothing. We also built a lexicalized reordering model (Koehn et al., 2005): *msd-bidirectional-fe*. We used a 5-gram language model trained on the GigaWord v.5 with Kneser-Ney smoothing using KenLM (Heafield, 2011). For optimization, we used MERT. For evaluation, we used NIST's BLEU scoring tool v13a, which we ran on a desegmented Arabic output, where conjunctions are attached to the following word.

In order to ensure stability, we performed three reruns of MERT for each experiment, and we report evaluation results averaged over the three reruns, as suggested by Foster and Kuhn (2009).

[2] www.nist.gov/itl/iad/mig/openmt12.cfm

4.2 Tuning on MT04, testing on MT05

TEST ⇒	AVERAGE		AVG, no self	
TUNE ⇓	BLEU	len	BLEU	len
MT040	29.41	1.014	**30.30**	1.020
MT041	30.13	**0.993**	30.18	**0.993**
MT042	30.07	0.991	30.14	0.990
MT043	30.03	0.983	29.36	0.981
MT044	**30.14**	0.986	29.32	0.982

Table 2: **Tuning and testing on MT04.** We tune on the English input in the first column, then we translate all MT04x inputs. We report BLEU and hyp/ref length ratios averaged over (a) all MT04 datasets, and (b) all but the one used for tuning.

Table 1 shows the results when tuning on MT04 and testing on MT05. There are several interesting observations we can make. First, the choice of *test* dataset has a huge impact on the BLEU score: in some cases, more than 11 BLEU points, e.g., compare MT052 to MT053. Second, from the *tuning* dataset perspective, we can see 0.51-0.78 absolute difference in BLEU between the best (mostly MT040) and the worst choice (mostly MT044). These differences are large enough to justify our interest in tuning input selection.

Table 1 also allows us to assess the performance of the two baselines: *select-first* is optimal, achieving an overall BLEU score of 34.24, while *concat-all* is in the middle (would be third best if ranked with the rest) with a BLEU score of 34.03.

Table 2 shows the results when tuning on one MT04 dataset, and testing on all MT04 datasets. The results are averaged (a) over all MT04 datasets and (b) over all but the one used for tuning. In case (a) (see columns 2 and 3), MT044 is selected, which is the worst possible choice. However, in case (b) (see columns 4 and 5), the best score is achieved for MT040, which is the optimal choice, i.e., *best-on-tuning* yields optimal results when averaging over all but the tuning dataset.

Moreover, note that the BLEU scores in column 4 of Table 2 go in strictly decreasing order for MT040, MT041, MT042, MT043, MT044, and they do so also in Table 1. This suggests that the *best-on-tuning* strategy is very reliable here.

TEST	REF: all-but-X BLEU	len
MT040	52.81	0.976
MT041	57.16	**1.005**
MT042	58.55	1.007
MT043	**63.28**	1.008
MT044	62.56	1.013

Table 3: **X vs. all-but-X for MT04.** BLEU scores and hyp/ref length ratios when testing on each English input, using all the rest as references.

Table 3 implements *X-vs-all-but-X*. It shows the results when tuning on each English input, using all other inputs as references. The highest BLEU score is achieved by MT043, which is the second worst choice. Thus, this is a very poor strategy here; however, below we will see that it is quite reliable if we make a choice based on length ratio.

TEST	Our System BLEU	len	Google BLEU	len
MT040	26.04	1.036	26.29	**0.992**
MT041	29.46	**0.979**	28.11	0.937
MT042	29.99	0.977	28.00	0.935
MT043	32.21	0.974	**30.36**	0.933
MT044	**32.27**	0.962	29.94	0.921

Table 4: **Backtranslate MT04.** BLEU scores and hyp/ref length ratios when backtranslating the Arabic reference to English, and then evaluating it with respect to each of the English inputs.

Table 4 shows the results when backtranslating the Arabic reference to English, and then scoring it with respect to each of the English inputs. The backtranslation uses (a) our own system trained to translate in the reverse direction, and (b) Google Translate. We can see that *backtranslate* performs poor: with (a), it selects MT044, the worst choice, and with (b), it selects MT043, the second worst; however, it works better if we use length ratios.

Table 5 shows the results when tuning on datasets synthesized from full sentences (all but the last line) or by fusing sentences (the last line), where we optimize some function with respect to a backtranslation obtained from (a) our own system or (b) Google Translate. We can see that no combination could improve over the best individual system, but the best synthesized dataset yielded a score matching that of the best individual system.

TUNE	Our System BLEU	len	Google BLEU	len
B1	34.05	0.971	33.92	0.981
B1-BP	34.11	0.967	33.94	0.977
B1-SG	34.03	0.982	34.19	0.989
DL	34.21	0.982	34.07	0.990
MIN-L	33.53	1.020	**34.24**	1.005
MIN-B1	**34.23**	0.978	34.05	0.966
MEMT	33.71	**0.998**	33.47	**1.000**

Table 5: **Tuning on synthesized MT04 datasets, testing on MT05.** BLEU scores and hyp/ref length ratios averaged over all MT05 test datasets.

We believe that these results are due to our inability to choose a reliable reference translation: backtranslation generates an automatic translation, which most of the time is arguably worse in quality than the English inputs, which are *human*, after all. In future work, we plan to try other ways to generate a good reference translation.

4.3 Tuning on MT05, testing on MT04

Table 6 shows the results when tuning on MT05 and testing on MT04. Once again, the choice of *test* dataset has a huge impact on the BLEU score: this time up to 7 BLEU points, e.g., compare MT040 to MT044. We further see 0.5-1.5 absolute difference in BLEU between the best (mostly MT051) and the worst choice (mostly MT050).

This time, *select-first* does not work at all: it selects MT050, which is the worst possible choice (while it was best in the reverse, MT04-MT05, translation direction). However, the *concat-all* strategy performs reasonably well: it would be second best if ranked together with the individual inputs (it was third best in the reverse direction).

Table 7 shows that the *best-on-tuning* strategy once again works quite well, selecting MT051, which is the optimal choice. Note that this time the optimal choice is made regardless of whether the averaging is done over all datasets or over all but the tuning dataset (in the reverse direction, averaging over all made the worst possible choice, while averaging over all but the one used for tuning made an optimal choice).

Next, Table 8 shows that *X-vs-all-but-X* would select MT054, which is in the middle of the possible choices: not the worst, but also not the best (it was second worst in the reverse direction).

Table 9 shows that *backtranslate* does not work well: for both our SMT system and Google Translate, it selects MT053, the second worst choice (it was also second worst in the reverse direction).

TEST ⇒	MT040		MT041		MT042		MT043		MT044		AVERAGE	
TUNE ⇓	BLEU	len	BLEU	len	BLEU	len	BLEU	len	BLEU	len	BLEU	len
MT050	25.23	0.989	~~28.41~~	1.018	~~28.28~~	1.022	~~30.98~~	1.026	~~31.08~~	1.031	~~28.80~~	1.017
MT051	25.49	0.963	**29.38**	0.987	**29.23**	0.990	**32.22**	0.996	**32.61**	1.001	**29.79**	0.987
MT052	25.27	0.971	28.67	0.994	28.87	0.996	31.58	1.003	31.85	1.008	29.25	0.994
MT053	~~24.98~~	0.921	28.72	0.944	28.85	0.945	31.90	0.953	32.30	0.957	29.35	0.944
MT054	25.42	0.973	29.27	0.986	28.66	1.000	31.90	1.005	32.19	1.009	29.49	0.994
MT05ALL	**25.53**	0.964	29.17	0.986	29.03	0.989	32.06	0.996	32.37	1.002	29.63	0.987
best−worst	0.55		0.97		0.95		1.24		1.53		0.99	

Table 6: **Tuning on MT05 and testing on MT04.** Shown are BLEU scores and hypothesis/reference length ratios. The best and the worst BLEU scores for each test MT04 dataset are in **bold** and ~~stroke out~~, respectively; the last row shows the absolute difference between them.

TEST ⇒	AVERAGE		AVG, no self	
TUNE ⇓	BLEU	len	BLEU	len
MT050	33.98	**0.995**	33.78	**0.996**
MT051	**34.28**	0.969	**35.11**	0.971
MT052	33.98	0.975	**35.11**	0.979
MT053	33.37	0.930	31.68	0.922
MT054	34.25	0.971	33.96	0.971

Table 7: **Tuning and testing on MT05.** We tune on the English input in the first column, then we translate all MT05x inputs. We report BLEU and hyp/ref length ratios averaged over (a) all MT05 datasets, and (b) all but the one used for tuning.

	REF: all-but-X	
TEST	BLEU	len
MT050	63.38	**0.998**
MT051	58.20	0.992
MT052	62.73	0.994
MT053	66.88	1.026
MT054	**70.53**	1.005

Table 8: **X vs. all-but-X for MT05.** BLEU scores and hyp/ref length ratios when testing on each English input, using all the rest as references.

Table 10 shows the results when tuning on synthesized datasets. As before, this does not improve over the best individual system. Again, we can blame this on the bad selection of reference, but there could be also something else: selection strategies that synthesize input datasets based on what is *easiest* to translate might not be as useful as we have assumed. In the following section, we give some insight on why this might be the case.

5 Discussion

So far, we have explored input selection alternatives that make use of BLEU as a central criterion (while we have also experimented with some sentence selection strategies based on length, this was peripheral), and, in many cases, these strategies were very successful.

	Our System		Google	
TEST	BLEU	len	BLEU	len
MT050	34.56	1.010	33.79	1.024
MT051	30.54	1.014	30.74	1.027
MT052	30.52	1.020	30.76	1.033
MT053	**38.66**	0.944	**37.66**	0.956
MT054	36.17	**0.992**	36.08	**1.005**

Table 9: **Backtranslate MT05.** BLEU scores and hyp/ref length ratios when backtranslating the Arabic reference to English, and then evaluating it with respect to each of the English inputs.

	Our System		Google	
TUNE	BLEU	len	BLEU	len
B1	**29.64**	**1.011**	29.33	**1.017**
B1-BP	29.36	1.014	**29.43**	**1.017**
B1-SG	28.93	1.023	29.38	1.020
DL	28.76	1.032	29.08	1.022
MIN-L	27.07	1.068	28.18	1.055
MIN-B1	28.57	1.020	28.82	1.030
MEMT	28.68	1.031	28.69	1.036

Table 10: **Tuning on synthesized MT05 datasets, testing on MT04.** BLEU scores and hyp/ref length ratios averaged over all MT04 test datasets.

Below we explore two alternative strategies for best input dataset selection for tuning: (a) looking for the dataset that yields a tuning length ratio that is closest to 1, and (b) choosing the hardest input. We further explore the potential of using perplexity for tuning input selection.

5.1 Choosing length closest to 1

Above, we have considered the BLEU/BLEU+1 score as the main criterion for input dataset selection. This makes sense since this is the standard evaluation measure, which we are optimizing at test time. However, there are other reasonable criteria that could be considered. For example, recent work has suggested that length is an important factor for parameter optimization in statistical machine translation (Nakov et al., 2012).

Thus, we considered how the above strategies would work when selecting not the dataset yielding the highest BLEU, but that for which the source/reference length ratio is closest to 1. This turned out to work in some but not all cases.

When tuning on MT04: Table 2 shows that if looking for the best length instead of the best BLEU, the *best-on-tuning* strategy would select MT041, which is the second best choice.

The same choice would make *X-vs-all-but-X* (see Table 3) and *backtranslate* when using our system (see Table 4). With Google Translate, however, it would make the best choice: MT040.

When tuning on MT05: Table 7 shows that *best-on-tuning* would select MT050, which is the worst choice. The same choice would make *X-vs-all-but-X* (see Table 8). Both strategies made the second best choice for MT04. The *backtranslate* strategy, however, selects MT054, both with our SMT system and with Google Translate; this is the second best choice (see Table 9). On MT04, this strategy made an optimal choice.

Overall, the length ratio works great for *backtranslate* (best or second best choice), but for *best-on-tuning* and *X-vs-all-but-X* results are mixed.

5.2 Choosing the hardest dataset

A closer look at the strategies for *backtranslate* and *X-vs-all-but-X* reveals something unexpected: Tables 3, 4, 8, and 9 show that selecting the input dataset with the lowest BLEU would yield an optimal choice in all these cases.

We had assumed that the input that yields the highest BLEU score should be of highest quality, and thus the best to learn from. Instead, a closer inspection has found that the high-BLEU datasets were more literal translations, which were less fluent in English and thus ultimately of lower quality. So, we should really train on the hardest dataset.

In fact, this is not very surprising: a student would learn more from hard lessons than from easy ones. Thus, the best strategy to prepare for an exam is to learn hard rather than easy lessons.

It is reasonable to expect that hard inputs would have lower perplexity with respect to our language model, i.e., that they would be more similar to the training data, and thus that they should be also closer to the expected test time input. We tested this hypothesis by calculating the perplexity for all input MT04 datasets, and we found for MT040 the perplexity is indeed lower than for MT044.

The results are shown in Table 11, where we show the logarithm of the probability instead of the perplexity because the perplexity was too low.

These numbers offer yet another possible explanation about why combining inputs could not improve: it looks like MT040 is much better than the rest, and thus maybe there are simply no enough good translations in the remaining datasets.

INPUT	log P
MT040	**-98,862**
MT041	-106,022
MT042	-103,542
MT043	-104,780
MT044	-106,341

Table 11: **Log-probability of the different inputs** calculated with respect to the language model.

6 Conclusion and Future Work

We have studied the question of how to select/synthesize a good *tuning* dataset for SMT in the special case, when we have multiple possible input (English) versions of the same sentence and a single reference (Arabic) translation.

We have experimented with a number of strategies, and we have found that it is best to tune on the hardest available input, not on the one that yields the highest BLEU score (i.e., the easiest). We believe that this finding has implications on how we should pick good translators and how we should select useful data for parameter optimization. On the other hand, it might also indicate a problem with BLEU as an evaluation measure.

In future work, we plan to test our methods on other Arabic-English datasets that have multiple English references. We further plan experiments with other language pairs, e.g., Chinese-English, which are available from NIST and IWSLT. We also want to study the effect of the tuning dataset selection on evaluation measures other than BLEU, e.g., TER (Snover et al., 2006) and METEOR (Lavie and Denkowski, 2009). Looking at tuning dataset selection that takes the test data into account is another promising direction for future work. Features from quality estimation (Specia et al., 2010) might be also helpful to determine the best input to tune on.

Another related, but different, research direction is about how to best *evaluate* (as opposed to *tune*, which we have explored above) an SMT system in case multiple possible versions of the input sentences are available.

References

Hassan Al-Haj and Alon Lavie. 2012. The impact of Arabic morphological segmentation on broad-coverage English-to-Arabic statistical machine translation. *Machine Translation*, 26(1-2):3–24.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP'11*, pages 355–362, Edinburgh, United Kingdom.

George Foster and Roland Kuhn. 2009. Stabilizing minimum error rate training. In *Proceedings of WMT'09*, pages 242–249, Athens, Greece.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of EMNLP'10*, pages 451–459, Cambridge, MA.

Kenneth Heafield and Alon Lavie. 2010. Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93(1):27–36.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of WMT'11*, pages 187–197, Edinburgh, Scotland.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT'05*, pages 133–142, Budapest, Hungary.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for English-Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL'03*, pages 48–54, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of IWSLT'05*, Pittsburgh, PA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL'07*, pages 177–180, Prague, Czech Republic.

Alon Lavie and Michael Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23:105–115.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of COLING'04*, pages 501–507, Geneva, Switzerland.

Lemao Liu, Hailong Cao, Taro Watanabe, Tiejun Zhao, Mo Yu, and Conghui Zhu. 2012. Locally training the log-linear model for SMT. In *Proceedings of EMNLP-CoNLL'12*, pages 402–411, Jeju Island, Korea.

Spyros Matsoukas, Antti-Veikko Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of EMNLP'09*, pages 708–717, Singapore.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proceedings of EACL'06*, pages 33–40, Trento, Italy.

Robert Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of ACL'10*, pages 220–224, Uppsala, Sweden.

Preslav Nakov, Francisco Guzmán, and Stephan Vogel. 2012. Optimizing for sentence-level BLEU+1 yields short translations. In *Proceedings of COLING'12*, pages 1979–1994, Mumbai, India.

Franz Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proceedings of MT Summit*, volume 8, pages 253–258, Santiago de Compostela, Spain.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, PA.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL'08*, pages 117–120, Columbus, OH.

Josh Schroeder, Trevor Cohn, and Philipp Koehn. 2009. Word lattices for multi-source translation. In *Proceedings of EACL'09*, pages 719–727, Athens, Greece.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA'06*, pages 223–231, Cambridge, MA.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine translation*, 24(1):39–50.