Analyzing Optimization for Statistical Machine Translation: MERT Learns Verbosity, PRO Learns Length

Francisco Guzmán **Preslav Nakov** and **Stephan Vogel**
ALT Research Group
Qatar Computing Research Institute, HBKU
{fguzman,pnakov,svogel}@qf.org.qa

Abstract

We study the impact of source length and verbosity of the tuning dataset on the performance of parameter optimizers such as MERT and PRO for statistical machine translation. In particular, we test whether the verbosity of the resulting translations can be modified by varying the length or the verbosity of the tuning sentences. We find that MERT learns the tuning set verbosity very well, while PRO is sensitive to both the verbosity and the length of the source sentences in the tuning set; yet, overall PRO learns best from high-verbosity tuning datasets.

Given these dependencies, and potentially some other such as amount of reordering, number of unknown words, syntactic complexity, and evaluation measure, to mention just a few, we argue for the need of controlled evaluation scenarios, so that the selection of tuning set and optimization strategy does not overshadow scientific advances in modeling or decoding. In the mean time, until we develop such controlled scenarios, we recommend using PRO with a large verbosity tuning set, which, in our experiments, yields highest BLEU across datasets and language pairs.

1 Introduction

Statistical machine translation (SMT) systems nowadays are complex and consist of many components such as a translation model, a reordering model, a language model, etc., each of which could have several sub-components. All components and their elements work together to score full and partial hypotheses proposed by the SMT system's search algorithms.

Thus, putting them together requires assigning them relative weights, e.g., how much weight we should give to the translation model vs. the language model vs. the reordering table. These relative weights are typically learned discriminatively in a log-linear framework, and their values are optimized to maximize some automatic metric, typically BLEU, on a tuning dataset.

Given this setup, it is clear that the choice of a tuning set and its characteristics, can have significant impact on the SMT system's performance: if the experimental framework (training data, tuning set, and test set) is highly consistent, i.e., there is close similarity in terms of genre, domain and verbosity,[1] then translation quality can be improved by careful selection of tuning sentences that exhibit high degree of similarity to the test set (Zheng et al., 2010; Li et al., 2010).

In our recent work (Nakov et al., 2012), we have studied the relationship between optimizers such as MERT, PRO and MIRA, and we have pointed out that PRO tends to generate relatively shorter translations, which could lead to lower BLEU scores on testing. Our solution there was to fix the objective function being optimized: PRO uses sentence-level smoothed BLEU+1, as opposed to the standard dataset-level BLEU.

Here we are interested in a related but different question: the relationship between properties of the tuning dataset and the optimizer's performance. More specifically, we study how the *verbosity*, i.e., the average target/source sentence length ratio, learned by optimizers such as MERT and PRO depends on the nature of the tuning dataset. This could potentially allow us to manipulate the *verbosity* of the translation hypotheses generated at test time by changing some characteristics of the tuning dataset.

[1] Verbosity also depends on the translator; it is often a stylistic choice. and not necessarily related to fluency or adequacy. This aspect is beyond the scope of the present work.

2 Related Work

Tuning the parameters of a log-linear model for statistical machine translation is an active area of research. The standard approach is to use minimum error rate training, or MERT, (Och, 2003), which optimizes BLEU directly.

Recently, there has been a surge in new optimization techniques for SMT. Two parameter optimizers that have recently become popular include the margin-infused relaxed algorithm or MIRA (Watanabe et al., 2007; Chiang et al., 2008; Chiang et al., 2009), which is an on-line sentence-level perceptron-like passive-aggressive optimizer, and pairwise ranking optimization or PRO (Hopkins and May, 2011), which operates in batch mode and sees tuning as ranking.

A number of improved versions thereof have been proposed in the literature including a batch version of MIRA (Cherry and Foster, 2012), with local updates (Liu et al., 2012), a linear regression version of PRO (Bazrafshan et al., 2012), and a non-sampling version of PRO (Dreyer and Dong, 2015); another example is Rampeon (Gimpel and Smith, 2012). We refer the interested reader to three recent overviews on parameter optimization for SMT: (McAllester and Keshet, 2011; Cherry and Foster, 2012; Gimpel and Smith, 2012).

Still, MERT remains the de-facto standard in the statistical machine translation community. Its stability has been of concern, and is widely studied. Suggestions to improve it include using regularization (Cer et al., 2008), random restarts (Moore and Quirk, 2008), multiple replications (Clark et al., 2011), and parameter aggregation (Cettolo et al., 2011).

With the emergence of new optimization techniques there have been also studies that compare stability between MIRA–MERT (Chiang et al., 2008; Chiang et al., 2009; Cherry and Foster, 2012), PRO–MERT (Hopkins and May, 2011), MIRA–PRO–MERT (Cherry and Foster, 2012; Gimpel and Smith, 2012; Nakov et al., 2012). Pathological verbosity was reported when using MERT on recall-oriented metrics such as METEOR (Lavie and Denkowski, 2009; Denkowski and Lavie, 2011), as well as large variance with MIRA (Simianer et al., 2012). However, we are not aware of any previous studies of the impact of sentence length and dataset verbosity across optimizers.

3 Method

For the following analysis, we need to define the following four quantities:

- *source-side length*: the number of words in the source sentence;
- *length ratio*: the ratio of the number of words in the output hypothesis to those in the reference;[2]
- *verbosity*: the ratio of the number of words in the reference to those in the source;[3]
- *hypothesis verbosity*: the ratio of the number of words in the hypothesis to those in the source.

Naturally, the *verbosity* varies across different tuning/testing datasets, e.g., because of style, translator choice, etc. Interestingly, verbosity can also differ across sentences with different source lengths drawn from *the same* dataset. This is illustrated in Figure 1, which plots the average sample source length vs. the average verbosity for 100 samples, each containing 500 randomly selected sentence pairs, drawn from the concatenation of the MT04, MT05, MT06, MT09 datasets for Arabic-English and of newstest2008-2011 for Spanish-English.[4]

We can see that for Arabic-English, the English translations are longer than the Arabic source sentences, i.e., the *verbosity* is greater than one. This relationship is accentuated by length: *verbosity* increases with sentence length: see the slightly positive slope of the regression line. Note that the increasing verbosity can be observed in single-reference sets (we used the first reference), and to a lesser extent in multiple-reference sets (five references for MT04 and MT05, and four for MT06 and MT09). For Spanish-English, the story is different: here the English sentences tend to be shorter than the Spanish ones, and the *verbosity* decreases as the sentence length increases. Overall, in all three cases, the *verbosity* appears to be length-dependent.

[2]For multi-reference sets, we use the length of the reference that is closest to the length of the hypothesis. This is the *best match length* from the original paper on BLEU (Papineni et al., 2002); it is default in the NIST scoring tool v13a, which we use in our experiments.

[3]When dealing with multi-reference sets, we use the average reference length.

[4]The datasets we experiment with are described in more detail in Section 4 below.

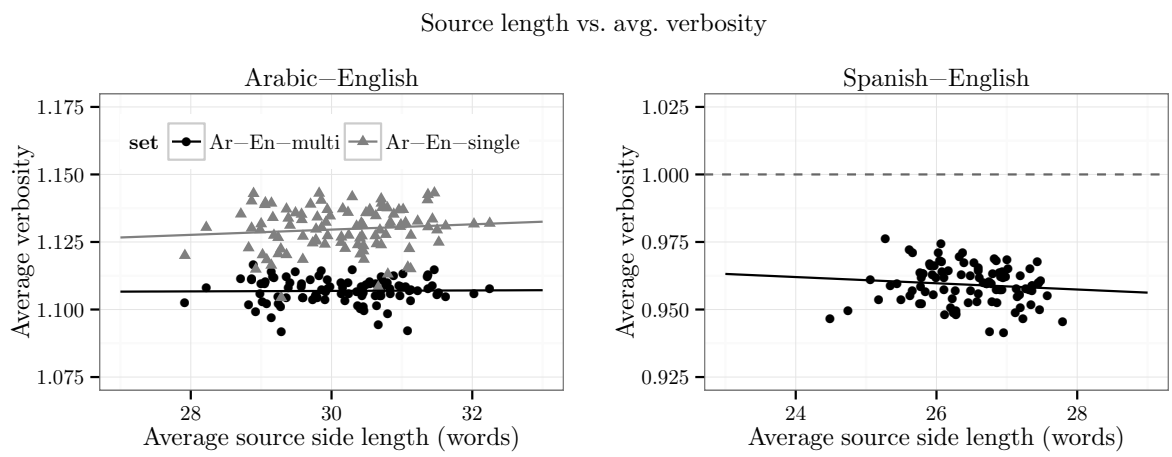

Figure 1: Average source sentence length (x axis) vs. average verbosity (y axis) for 100 random samples, each with 500 sentence pairs extracted from NIST (Left: Arabic-English, multi- and single-reference) and from WMT (Right: Spanish-English, single-reference) data.

The main research question we are interested in, and which we will explore in this paper, is whether the SMT parameter optimizers are able to learn the *verbosity* from the tuning set. We are also interested in the question of how the *hypothesis verbosity* learned by optimizers such as MERT and PRO depends on the nature of the tuning dataset, i.e., its *verbosity*. Understanding this could potentially allow us to manipulate the *hypothesis verbosity* of the translations generated at test time simply by changing the characteristics of the tuning dataset in a systematic and controlled way. While controlling the verbosity of a tuning set might be an appealing idea, this is unrealistic in practice, given that the verbosity of a test set is always unknown. However, the results in Figure 1 suggest that it is possible to manipulate *verbosity* by controlling the average source sentence length of the dataset (and the *source-side* length is always known for any test set). Thus, in our study, we use the source-side sentence length as a data selection criterion; still, we also report results for selection based on verbosity.

In order to shed some light on our initial question (whether the SMT parameter optimizers are able to learn the *verbosity* from the tuning dataset), we contrast the *verbosity* that two different optimizers, MERT and PRO, learn as a function of the average length of the sentences in the tuning dataset.[5]

[5]In this work, we consider both optimizers, MERT and PRO, as *black-boxes*. For a detailed analysis of how their inner workings can affect optimization, see our earlier work (Nakov et al., 2012).

4 Experiments and Evaluation

We experimented with single-reference and multi-reference tuning and testing datasets for two language pairs: Spanish-English and Arabic-English. For Spanish-English, we used the single-reference datasets newstest2008, newstest2009, newstest2010, and newstest2011 from the WMT 2012, Workshop on Machine Translation Evaluation.[6] For Arabic-English, we used the multi-reference datasets MT04, MT05, MT06, and MT09 from the NIST 2012 OpenMT Evaluation;[7] we further experimented with single-reference versions of the MT0x datasets, using the first reference only.

In addition to the above datasets, we constructed tuning sets of different source-side sentence lengths: *short*, *middle* and *long*. Given an original tuning dataset, we selected 50% of its sentence pairs: *shortest* 50%, *middle* 50%, or *longest* 50%. This yielded tuning datasets with the same number of sentence pairs but with different number of words, e.g., for our Arabic-English datasets, *longest* has about twice as many English words as *middle*, and about four times as many words as *shortest*. Constructing tuning datasets with the same number of sentences instead of the same number of tokens is intentional as we wanted to ensure that in each of the conditions, the SMT parameter optimizers learn on the same number of training examples.

[6]www.statmt.org/wmt12/

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

4.1 Experimental Setup

We experimented with the phrase-based SMT model (Koehn et al., 2003) as implemented in Moses (Koehn et al., 2007). For Arabic-English, we trained on all data that was allowed for use in the NIST 2012 except for the UN corpus. For Spanish-English, we used all WMT12 data, again except for the UN data.

We tokenized and truecased the English and the Spanish side of all bi-texts and also the monolingual data for language modeling using the standard tokenizer of Moses. We segmented the words on the Arabic side using the MADA ATB segmentation scheme (Roth et al., 2008). We built our phrase tables using the 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 GigaWord v.5 with Kneser-Ney smoothing using KenLM (Heafield, 2011).

On tuning and testing, we dropped the unknown words for Arabic-English, and we used monotone-at-punctuation decoding for Spanish-English. We tuned using MERT and PRO. We used the standard implementation of MERT from the Moses toolkit, and a fixed version of PRO, as we recommended in (Nakov et al., 2013), which solves instability issues when tuning on the *long* sentences; we will discuss our PRO fix and the reasons it is needed in Section 5 below. In order to ensure convergence, we allowed both MERT and PRO to run for up to 25 iterations (default: 16); we further used 1000-best lists (default: 100).

In our experiments below, we perform three reruns of parameter optimization, tuning on each of the twelve tuning datasets; in the figures, we plot the results of the three reruns, while in the tables, we report BLEU averaged over the three reruns, as suggested by Clark et al. (2011).

4.2 Learning Verbosity

We performed parameter optimization using MERT and PRO on each dataset, and we used the resulting parameters to translate the same dataset. The purpose of this experiment was to study the ability of the optimizers to learn the *verbosity* of the tuning sets. Getting the *hypothesis verbosity* right means that it is highly correlated with the tuning set *verbosity* , which in turn is determined by the dataset source length.

The results are shown in Figure 2. In each graph, there are 36 points (many of them very close and overlapping) since we performed three reruns with our twelve tuning datasets (three length-based subsets for each of the four original tuning datasets). There are several observations that we can make:

(1) MERT is fairly stable with respect to the length of the input tuning sentences. Note how the MERT regression lines imitate those in Figure 1. In fact, the correlation between the *verbosity* and the *hypothesis verbosity* for MERT is r=0.980. PRO, on the other hand, has harder time learning the tuning set verbosity, and the correlation with the *hypothesis verbosity* is only r=0.44. Interestingly, its *length ratio* is more sensitive to the input length (r=0.67): on *short* sentences, it learns to output translations that are slightly shorter than the reference, while on *long* sentences, it yields increasingly longer translations. The dependence of PRO on source length can be explained by the sentence-level smoothing in BLEU+1 and the broken balance between BLEU's precision component and BP (Nakov et al., 2012). The problem is bigger for short sentences since there +1 is added to smaller counts; this results in preference for shorter translations.

(2) Looking at the results for Arabic-English, we observe that having multiple references makes both MERT and PRO appear more stable, allowing them to generate hypotheses that are less spread, and closer to 1. This can be attributed to the *best match reference length*, which naturally dampens the effect of verbosity during optimization by selecting the reference that is closest to the respective hypothesis.

Overall, we can conclude that MERT learns the tuning set's *verbosity* more accurately than PRO. PRO learns *verbosity* that is more dependent on the *source side length* of the sentences in the tuning dataset.

4.3 Performance on the Test Dataset

Next, we study the performance of MERT and PRO when testing on datasets that are different from the one used for tuning. First, we test the robustness of the parameters obtained for specific tuning datasets when testing on various test datasets. Second, we test whether selecting a tuning dataset based on the length of the testing dataset (i.e., *closest*) is a good strategy.

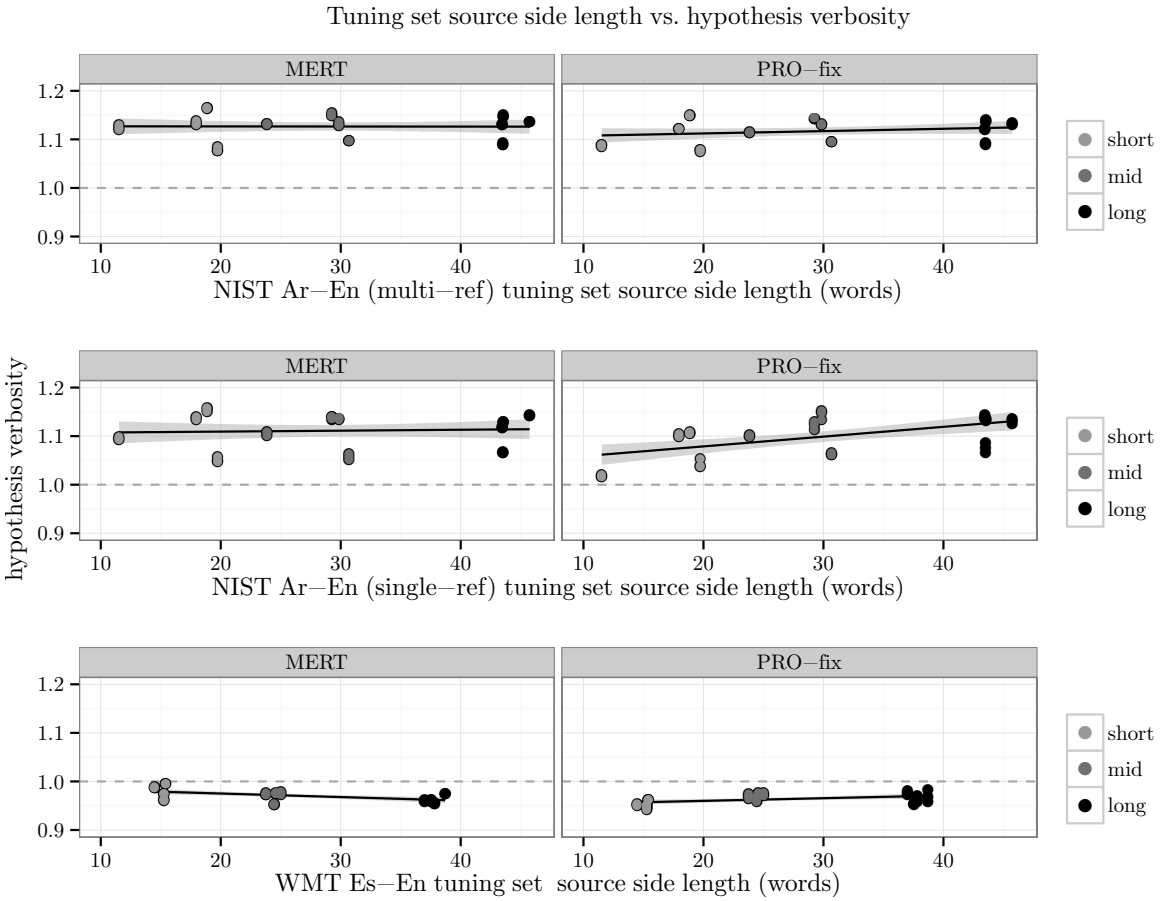

Figure 2: *Source-side length* vs. *hypothesis verbosity* for the tuning dataset. There are 36 points per language pair: four tuning sets, each split into three datasets (*short*, *middle*, and *long*) times three reruns.

For this purpose, we perform a grid comparison of tuning and testing on all our datasets: we tune on each *short*/*middle*/*long* dataset, and we test on the remaining *short*/*middle*/*long* datasets.

The results are shown in Table 1, where each cell is an average over 36 BLEU scores: four tuning sets times three test sets times three reruns. For example, 49.63 in row 1 (tune: short), column 2 (test: middle), corresponds to the average over three reruns of (i) tune on MT04-short and test on MT05-middle, MT06-middle, and MT09-middle, (ii) tune on MT05-short and test on MT04-middle, MT06-middle, and MT09-middle, (iii) tune on MT06-short and test on MT04-middle, MT05-middle, and MT09-middle, and (iv) tune on MT09-short and test on MT04-middle, MT05-middle, and MT06-middle. We further include two statistics: (1) the range of values (max-min), measuring test BLEU variance depending on the tuning set, and (2) the *loss* in BLEU when tuning on *closest* instead of on the best-performing dataset.

There are several interesting observations:

(1) PRO and MERT behave quite differently with respect to the input tuning set. For MERT, tuning on a specific length condition yields the best results when testing on a similar condition, i.e., *zero-loss*. This is a satisfactory result since it confirms the common wisdom that tuning datasets should be as similar as possible to test-time input in terms of *source side length*. In contrast, PRO behaves better when tuning on mid-length tuning sets. However, the average *loss* incurred by applying the *closest* strategy with PRO is rather small, and in practice, choosing a tuning set based on test set's average length is a good strategy.

(2) MERT has higher variance than PRO and fluctuates more depending on the input tuning set. PRO on the contrary, tends to perform more consistently, regardless of the length of the tuning set.

(3) MERT yields the best BLEU across datasets and language pairs. Thus, when several tuning sets are available, we recommend choosing the one closest in length to the test set and using MERT.

	test									
	Arabic-English (multi-ref)			**Arabic-English (1-ref)**			**WMT Spanish-English**			
tuning	**short**	**mid**	**long**	**short**	**mid**	**long**	**short**	**mid**	**long**	avg
MERT										
short	**47.26***	50.71	50.82	**26.69***	28.14	27.49	**25.17***	25.94	27.64	
mid	46.53	**51.11***	51.31	26.22	**28.39***	27.96	24.96	**26.27***	27.97	
long	46.23	50.84	**51.74***	25.80	28.20	**28.27***	24.57	26.08	**28.29***	
max-min	1.04	0.40	0.91	0.89	0.25	0.78	0.59	0.34	0.65	0.65
loss if using *closest*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PRO-fix										
short	**46.74**	50.57	50.97	25.95	27.66	27.28	24.66	**25.83**	**27.89**	
mid	46.59	**50.83**	**51.41**	**25.98**	**28.23**	**28.19**	**24.67**	25.81	27.64	
long	46.08	50.56	51.18	25.87	28.11	28.05	24.58	25.77	27.81	
max-min	0.66	0.27	0.44	0.11	0.58	0.92	0.09	0.06	0.25	0.38
loss if using *closest*	0.00	0.00	0.23	0.02	0.00	0.15	0.01	0.00	0.08	0.06

Table 1: Average test BLEU when tuning on each *short*/*mid*/*long* dataset, and testing on the remaining *short*/*mid*/*long* datasets. Each cell represents the average over 36 scores (see the text). The best score for either MERT or PRO is bold; the best overall score is marked with a *.

4.3.1 Performance vs. Length and Verbosity

The above results give rise to some interesting questions: What if we do not know the source-side length of the test set? What if we can choose a tuning set based on its verbosity? Would it then be better to choose based on length or based on verbosity?

To answer these questions, we analyzed the average results according to two orthogonal views: one based on the tuning set length (using the above 50% length-based subsets of tuning: *short*, *mid*, *long*), and another one based on the tuning set verbosity (using new 50% subsets verbosity-based subsets of tuning: *low-verb*, *mid-verb*, *high-verb*). This time, we translated the full test datasets (e.g., MT06, MT09); the results are shown in Table 2. We can make the following observations:

(1) The best results for PRO are better than the best results for MERT, in all conditions.

(2) Length-based tuning subsets: With a single reference, PRO performs best when tuning on short sentences, but with multiple references, it works best with mid-length sentences. MERT, on the other hand, prefers tuning on long sentences for all testing datasets.

(3) Verbosity-based tuning subsets: PRO yields best results when the tuning sets have high verbosity; in fact, the best verbosity-based results in the table are obtained with this setting. With multiple references, MERT performs best when tuning on high-verbosity datasets; however, with a single reference, it prefers mid-verbosity.

Based on the above results, we recommend that, whenever we have no access to the input side of the testing dataset beforehand, we should tune on datasets with high verbosity.

4.4 Test vs. Tuning Verbosity and Source Length

In the previous subsection, we have seen that MERT and PRO perform differently in terms of BLEU, depending on the characteristics of the tuning dataset. Here, we study a different aspect: i.e. how they behave with respect to *verbosity* and *source side length*.

We have seen that MERT and PRO perform differently in terms of BLEU depending on the characteristics of the tuning dataset. Below we study how other characteristics of the output of PRO and MERT are affected by tuning set *verbosity* and *source side length*.

4.4.1 MERT – Sensitive to Verbosity

Figure 3 shows a scatter plot of tuning *verbosity* vs. test *hypothesis verbosity* when using MERT to tune under different conditions, and testing on each of the unseen full datasets. We test on full datasets to avoid the *verbosity* bias that might occur for specific conditions (see Section 3).

We can see strong positive correlation between the tuning set *verbosity* and the *hypothesis verbosity* on the test datasets. The average correlation for Arabic-English is r=0.95 with multiple references and r=0.98 with a single reference; for Spanish-English, it is r=0.97.

	test					
	Arabic-English (multi-ref)		**Arabic-English (1-ref)**		**WMT Spanish-English**	
tuning	**MERT**	**PRO-fix**	**MERT**	**PRO-fix**	**MERT**	**PRO-fix**
length						
short	48.71	49.12	26.74	**27.35**	26.79	**27.07**
mid	49.27	**49.59**	26.97	27.23	26.99	26.88
long	**49.35**	49.20	**27.23**	27.28	**27.02**	26.84
verbosity						
low-verb	47.90	47.60	25.89	25.88	26.70	26.61
mid-verb	49.16	49.52	**27.69**	27.95	**27.09**	26.81
high-verb	**50.28**	**50.79***	27.36	**28.03***	27.01	**27.38***

Table 2: Average test BLEU scores when tuning on different length- and verbosity-based datasets, and testing on the remaining *full* datasets. Each cell represents the average over 36 scores. The best score for either MERT or PRO is bold; the best overall score is marked with a *.

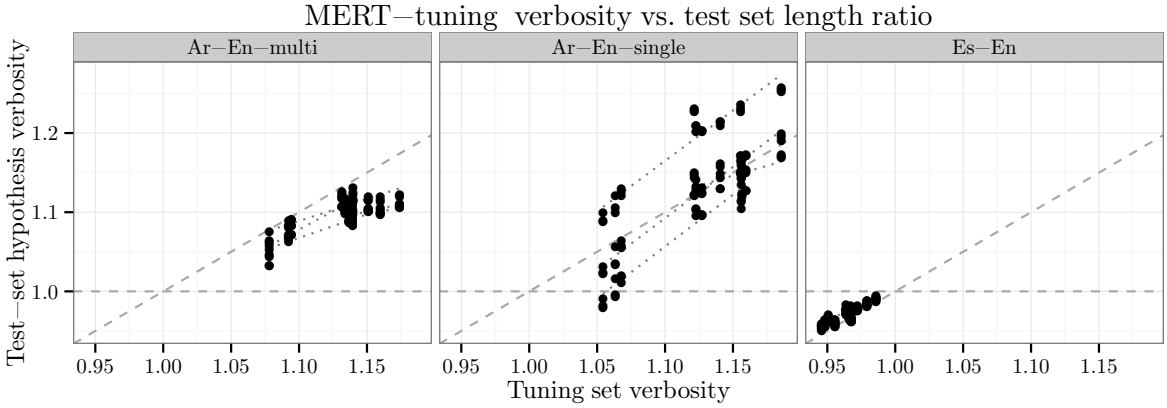

Figure 3: Tuning set *verbosity* vs. test *hypothesis verbosity* when using MERT. Each point represents the result for an unseen testing dataset, given a specific tuning condition. The linear regressions show the tendencies for each of the test datasets (note that they all overlap for Es-En and look like a single line).

These are very strong positive correlations and they show that MERT tends to learn SMT parameters that yield translations preserving the *verbosity*, e.g., lower *verbosity* on the tuning dataset will yield test-time translations that are less verbose, while higher *verbosity* on the tuning dataset will yield test-time translations that are more verbose. In other words, MERT learns to generate a fixed number of words per input word. This can be explained by the fact that MERT optimizes BLEU score directly, and thus learns to output the "right" *verbosity* on the tuning dataset (in contrast, PRO optimizes sentence-level BLEU+1, which is an approximation to BLEU, but it is not the actual BLEU). This explains why MERT performs best when the tuning conditions and the testing conditions are in sync. Yet, this makes it dependent on a parameter that we do not necessarily control or have access to beforehand: the length of the test *references*.

4.4.2 PRO – Sensitive to Source Length

Figure 4 shows the tuning set average *source-side length* vs. the testing hypothesis/reference *length ratio* when using PRO to tune on *short*, *middle*, and *long* and testing on each of the unseen full datasets, as in the previous subsection. We can see that there is positive correlation between the tuning set average *source side length* and the testing hypothesis/reference *length ratio*. For Spanish-English, it is quite strong (r=0.64), and for Arabic-English, it is more clearly expressed with one (r=0.42) than with multiple references (r=0.34). The correlation is significant ($p < 0.001$) when we take into account the contribution of the tuning set *verbosity* in the model. This suggests that for PRO, both *source length* and *verbosity* influence the hypotheses lengths, i.e., PRO learns the tuning set's *verbosity*, much like MERT; yet, the contribution of the length of the source sentences from the tuning dataset is not negligible.

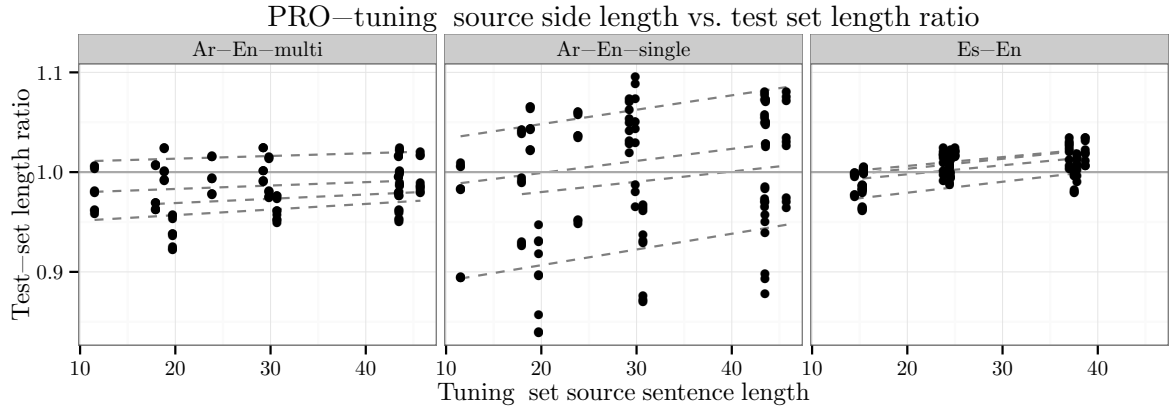

Figure 4: Tuning set average *source length* vs. test hypothesis/reference *length ratio* for PRO. Each point represents the result for an unseen testing dataset, given a specific tuning condition. The linear regressions shows the tendencies across each of the testing datasets.

Finally, note the "stratification" effect for the Arabic-English single-reference data. We attribute it to the differences across test datasets. These differences are attenuated with multiple references due to the *closest-match reference length*.

5 Discussion

We have observed that high-verbosity tuning sets yield better results with PRO. We have further seen that we can manipulate verbosity by adjusting the average length of the tuning dataset. This leads to the natural question: can this yield better BLEU? It turns out that the answer is "yes". Below, we present an example that makes this evident.

First, recall that for Arabic-English longer tuning datasets have higher verbosity. Moreover, our previous findings suggest that for PRO, higher-verbosity tuning datasets will perform better in this situation. Therefore, we should expect that longer tuning datasets could yield better BLEU. Table 3 presents the results for PRO with Arabic-English when tuning on MT06, or subsets thereof, and testing on MT09. The table shows the results for both multi- and single-reference experiments; naturally, manipulating the tuning set has stronger effect with a single reference. Lines 1-3 show that as the average length of the tuning dataset increases, so does the length ratio, which means better brevity penalty for BLEU and thus higher BLEU score. Line 4 shows that selecting a random-50% subset (included here to show the effect of using mixed-length sentences) yields results that are very close to those for *middle*.

Comparing line 3 to lines 4 and 5, we can see that tuning on *long* yields longer translations and also higher BLEU, compared to tuning on the full dataset or on *random*.

Next, lines 6 and 7 show the results when applying our smoothing fix for sentence-level BLEU+1 (Nakov et al., 2012), which prevents translations from becoming too short; we can see that *long* yields very comparable results. Yet, manipulating the tuning dataset might be preferable since it allows (*i*) faster tuning, by using part of the tuning dataset, (*ii*) flexibility in the selection of the desired verbosity, and (*iii*) applicability to other MT evaluation measures. Point (*ii*) is illustrated on Figure 5, which shows that there is direct positive correlation between verbosity, length ratio, and BLEU; note that the tuning set size does not matter much: in fact, better results are obtained when using less tuning data.

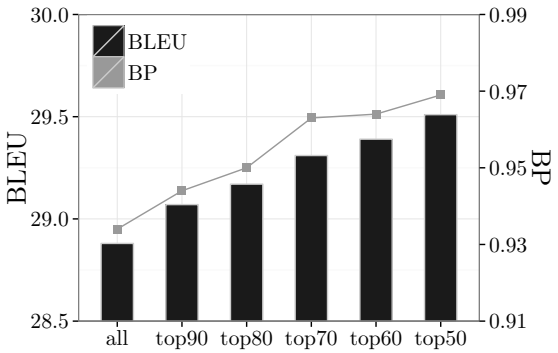

Figure 5: PRO, Arabic-English, 1-ref: tune on $N\%$ longest sentences from MT06, test on MT09.

	Tuning	multi-ref BLEU	multi-ref len. ratio	1-ref BLEU	1-ref len. ratio
1	tune-short	46.38	0.961	27.44	0.894
2	tune-mid	47.44	0.977	29.11	0.950
3	tune-long	**47.47**	0.980	**29.51**	0.969
4	tune-random	47.43	0.978	28.96	0.941
5	*tune-full*	*47.18*	*0.972*	*28.88*	*0.934*
6	tune-full, BP-smooth=1	**47.52**	0.984	**29.43**	0.962
7	tune-full, BP-smooth=1, grounded	**47.61**	0.991	**29.68**	0.979

Table 3: PRO, Arabic-English: tuning on MT06, or subsets thereof, and testing on MT09. Statistically significant improvements over *tune-full* are in bold: using the sign test (Collins et al., 2005), $p < 0.05$.

6 Conclusion and Future Work

Machine translation has, and continues to, benefit immensely from automatic evaluation measures. However, we frequently observe delicate dependencies between the evaluation metric, the system optimization strategy, and the pairing of tuning and test datasets. This leaves us with the situation that *getting lucky* in the selection of tuning datasets and optimization strategy overshadows scientific advances in modeling or decoding. Understanding these dependencies in detail puts us in a better position to construct tuning sets that match the test datasets in such a way that improvements in models, training, and decoding algorithms can be measured more reliably.

To this end, we have studied the impact that source-side length and verbosity of tuning sets have on the performance of the translation system when tuning the system with different optimizers such as MERT and PRO. We observed that MERT learns the verbosity of the tuning dataset very well, but this can be a disadvantage because we do not know the verbosity of unseen test sentences. In contrast, PRO is affected by both the verbosity and the source-side length of the tuning dataset.

There may be other characteristics of test datasets, e.g., amount of reordering, number of unknown words, complexity of the sentences in terms of syntactic structure, etc. that could have similar effects of creating good or bad luck when deciding how to tune an SMT system. Until we have such controlled evaluation scenarios, our short-term recommendations are as follows:

- Know your tuning datasets: Different language pairs and translation directions may have different *source-side length – verbosity* dependencies.
- When optimizing with PRO: select or construct a high-verbosity dataset as this could potentially compensate for PROs tendency to yield too short translations. Note that for Arabic-English, higher verbosity means longer tuning sentences, while for Spanish-English, it means shorter ones; translation direction might matter too.
- When optimizing with MERT: If you know beforehand the test set, select the *closest* tuning set. Otherwise, tune on longer sentences.

We plan to extend this study in a number of directions. First, we would like to include other parameter optimizers such as Rampeon (Gimpel and Smith, 2012) and MIRA. Second, we want to experiment with other metrics, such as TER (Snover et al., 2006), which typically yields short translations, and METEOR (Lavie and Denkowski, 2009), which yields too long translations. Third, we would like to explore other SMT models such as hierarchical (Chiang, 2005) and syntax-based (Galley et al., 2004; Quirk et al., 2005), and other decoders such as cdec (Dyer et al., 2010), Joshua (Li et al., 2009), and Jane (Vilar et al., 2010).

A long-term objective would be to design a metric that measures the closeness between tuning and test datasets, which includes the different characteristics, such as length distribution, verbosity distribution, syntactic complexity, etc., to guarantee a more stable evaluation situations, but which would also allow to systematically test the robustness of translation systems, when deviating from the matching conditions.

Text

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments, which have helped us to improve the paper.

References

Marzieh Bazrafshan, Tagyoung Chung, and Daniel Gildea. 2012. Tuning as linear regression. In *Proceedings of the 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '12, pages 543–547, Montréal, Canada.

Daniel Cer, Daniel Jurafsky, and Christopher D Manning. 2008. Regularization and search for minimum error rate training. In *Proceedings of the Third Workshop on Statistical Machine Translation*, WMT '08, pages 26–34, Columbus, Ohio, USA.

Mauro Cettolo, Nicola Bertoldi, and Marcello Federico. 2011. Methods for smoothing the optimizer instability in SMT. In *Proceedings of the Thirteenth Machine Translation Summit*, MT Summit XIII, pages 32–39, Xiamen, China.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '12, pages 427–436, Montréal, Canada.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 224–233, Honolulu, Hawaii, USA.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '09, pages 218–226, Boulder, Colorado, USA.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, ACL '05, pages 263–270, Ann Arbor, Michigan, USA.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL-HLT '11, pages 176–181, Portland, Oregon, USA.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 531–540, Ann Arbor, Michigan.

Michael Denkowski and Alon Lavie. 2011. Meteor-tuned phrase-based SMT: CMU French-English and Haitian-English systems for WMT 2011. Technical report, Technical Report CMU-LTI-11-011, Language Technologies Institute, Carnegie Mellon University.

Markus Dreyer and Yuanzhe Dong. 2015. APRO: All-pairs ranking optimization for MT tuning. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '15, pages 1018–1023, Denver, Colorado, USA.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, ACL '10, pages 7–12, Uppsala, Sweden.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '04, pages 273–280, Boston, Massachusetts, USA.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '12, pages 221–231, Montréal, Canada.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, Edinburgh, United Kingdom.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1352–1362, Edinburgh, Scotland, United Kingdom.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (Volume 1)*, 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 the International Workshop on Spoken Language Translation*, IWSLT '05, Pittsburgh, Pennsylvania, USA.

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 the 45th Annual Meeting of the Association for Computational Linguistics (Demonstration session).*, ACL '07, pages 177–180, Prague, Czech Republic.

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

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, WMT '09, pages 135–139, Athens, Greece.

Mu Li, Yinggong Zhao, Dongdong Zhang, and Ming Zhou. 2010. Adaptive development data selection for log-linear model in statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 662–670, Beijing, China.

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 the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 402–411, Jeju Island, Korea.

David McAllester and Joseph Keshet. 2011. Generalization bounds and consistency for latent structural probit and ramp loss. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, NIPS '11, pages 2205–2212, Granada, Spain.

Robert C Moore and Chris Quirk. 2008. Random restarts in minimum error rate training for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, COLING '08, pages 585–592, Manchester, United Kingdom.

Preslav Nakov, Francisco Guzmán, and Stephan Vogel. 2012. Optimizing for sentence-level BLEU+1 yields short translations. In *Proceedings of the 24th International Conference on Computational Linguistics*, COLING '12, pages 1979–1994, Mumbai, India.

Preslav Nakov, Francisco Guzmán, and Stephan Vogel. 2013. A tale about PRO and monsters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '13, pages 12–17, Sofia, Bulgaria.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL '03, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania, USA.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 271–279, Ann Arbor, Michigan, USA.

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 the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL '08, pages 117–120, Columbus, Ohio, USA.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 11–21, Jeju Island, Korea.

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 the 7th Biennial Conference of the Association for Machine Translation in the Americas*, AMTA '06, pages 223–231, Cambridge, Massachusetts, USA.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proceedings of the 5th Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 262–270, Uppsala, Sweden.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 764–773, Prague, Czech Republic.

Zhongguang Zheng, Zhongjun He, Yao Meng, and Hao Yu. 2010. Domain adaptation for statistical machine translation in development corpus selection. In *Proceedings of the 4th International Universal Communication Symposium*, IUCS '10, pages 2–7, Beijing, China.