Results of the WMT15 Tuning Shared Task

Miloš Stanojević and **Amir Kamran**
University of Amsterdam
ILLC
{m.stanojevic,a.kamran}@uva.nl

Ondřej Bojar
Charles University in Prague
MFF ÚFAL
bojar@ufal.mff.cuni.cz

Abstract

This paper presents the results of the WMT15 Tuning Shared Task. We provided the participants of this task with a complete machine translation system and asked them to tune its internal parameters (feature weights). The tuned systems were used to translate the test set and the outputs were manually ranked for translation quality. We received 4 submissions in the English-Czech and 6 in the Czech-English translation direction. In addition, we ran 3 baseline setups, tuning the parameters with standard optimizers for BLEU score.

1 Introduction

Almost all modern statistical machine translation (SMT) systems internally consider translation candidates from several aspects. Some of these aspects can be very simple and one parameter is sufficient to capture them, such as the word penalty incurred for every word produced or the phrase penalty controlling whether the sentence should be translated in fewer or more independent phrases, leading to more or less word-for-word translation. Other aspects try to assess e.g. the fidelity of the translation, the fluency of the output or the amount of reordering. These are far more complex and formally captured in a model such as the translation model or language model.

Both the simple penalties as well as the scores from the more complex models are called *features* and need to be combined to a single score to allow for ranking of translation candidates. This is usually done using a linear combination of the scores:

$$\text{score}(e) = \sum_{m=1}^{M} \lambda_m h_m(e, f) \qquad (1)$$

where e and f are the candidate translation and the source, respectively, and $h_m(\cdot, \cdot)$ is one of the M penalties or models. The tuned parameters are $\lambda_m \in \mathbb{R}$, called *feature weights*.

Feature weights have a tremendous effect on the final translation quality. For instance the system can produce extremely long outputs, fabulating words just in order to satisfy a negatively-weighted word penalty, i.e. a bonus for each word produced. An inherent part of the preparation of MT systems is thus some optimization of the weight settings.

If we had to set the weights manually, we would have to try a few configurations and pick one that leads to reasonable outputs. The common practice is to use an optimization algorithm that examines many settings, evaluating the produced translations automatically against reference translations using some evaluation measure (traditionally called "metric" in the MT field). In short, the optimizer tunes model weights so that the final combined model score correlates with the metric score.

The metric score, in turn, is designed to correlate well with human judgements of translation quality, see Stanojević et al. (2015) and the previous papers summarizing WMT metrics tasks. However, a metric that correlates well with humans on final output quality may not be usable in weight optimization for various technical reasons. BLEU (Papineni et al., 2002) was shown to be very hard to surpass (Cer et al., 2010) and this is also confirmed by the results of the invitation-only WMT11 Tunable Metrics Task (Callison-Burch et al., 2010)[1]. Note however, that some metrics have been successfully used for system tuning (Liu et al., 2011; Beloucif et al., 2014).

The aim of the WMT15 Tuning Task[2] is to attract attention to the exploration of all the three

[1] http://www.statmt.org/wmt11/tunable-metrics-task.html

[2] http://www.statmt.org/wmt15/tuning-task/

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 274–281,
Lisboa, Portugal, 17-18 September 2015.

	Source	Sentences cs	Sentences en	Tokens cs	Tokens en	Types cs	Types en
LM corpora	News Commentary v8	162309	247966	3.6M	6.2M	162K	81K
TM corpora	Europarl v7, CCrawl and News Comm. v9	911952		17.7M	20.8M	652K	361K
Dev set	newstest2014	3003		51K	60K	19K	13K
Test set	newstest2015	2656		39K	47K	16K	11K

Table 1: Data used in the WMT15 tuning task.

	Dev		Test	
Direction	Token	Type	Token	Type
en-cs	2570	2032	2003	1655
cs-en	3891	3415	3381	3011

Table 2: Out of vocabulary word counts

aspects of model optimization: (1) the set of features in the model, (2) optimization algorithm, and (3) MT quality metric used in optimization.

For (1), we provide a fixed set of "dense" features and also allow participants to add additional "sparse" features. For (2), the optimization algorithm, task participants are free to use one of the available algorithms for direct loss optimization (Och, 2003; Zhao and Chen, 2009), which are usually capable of optimizing only a dozen of features, or one of the optimizers handling also very large sets of features (Cherry and Foster, 2012; Hopkins and May, 2011), or a custom algorithm. And finally for (3), participants can use any established evaluation metric or a custom one.

1.1 Tuning Task Assignment

Tuning task participants were given a complete model for the hierarchical variant of the machine translation system Moses (Hoang et al., 2009) and the development set (newstest2014), i.e. the source and reference translations. No "dev test" set was provided, since we expected that participants will internally evaluate various variants of their method by manually judging MT outputs. In fact, we offered to evaluate a certain number of translations into Czech for free to ease the participation for teams without any access to speakers of Czech; only one team used this service once.

A complete model consists of a rule table extracted from the parallel corpus, the default glue grammar and the language model extracted from the monolingual data. As such, this defines a fixed set of dense features. The participants were allowed to add any sparse features implemented in Moses Release 3.0 (corresponds to Github commit 5244a7b607) and/or to use any optimization algorithm and evaluation metric. Fully manual optimization was also not excluded but nobody seemed to take this approach.

Each submission in the tuning task consisted of the configuration of the MT system, i.e. the additional sparse features (if any) and the values of all the feature weights, λ_m.

2 Details of Systems Tuned

The systems that were distributed for tuning are based on Moses (Hoang et al., 2009) implementation of hierarchical phrase-based model (Chiang, 2005). The language models were 5-gram models with Kneser-Ney smoothing (Kneser and Ney, 1995) built using KenLM (Heafield et al., 2013). For word alignments, we used Mgiza++ (Gao and Vogel, 2008).

The parallel data used for training translation models consisted of the Europarl v7, News Commentary data (`parallel-nc-v9`) and CommonCrawl, as released for WMT14.[3] We excluded CzEng because we wanted to keep the task small and accessible to more groups.

Since the test set (newstest2015) and the development set (newstest2014) are in the news domain, we opted to exclude Europarl from the language model data. We did not add any monolingual news on top of News Commentary, which are quite close to the news domain. In retrospect, we should have added also some of the monolingual news data as released by WMT, esp. since we used a 5-gram LM.

Before any further processing, the data was tokenized (using Moses tokenizer) and lowercased. We also removed sentences longer than 60 words or shorter than 4 words. Table 1 summarizes the final dataset sizes and Table 2 provides details on out-of-vocabulary items.

Aside from the dev set provided, the participants were free to use any other data for tuning (making their submission "unconstrained"), but no participant decided to do that. All tuning task submissions are therefore also constraint in terms of

[3] `http://www.statmt.org/wmt14/translation-task.html`

System	Participant
BLEU-*	baselines
AFRL	United States Air Force Research Laboratory (Erdmann and Gwinnup, 2015)
DCU	Dublin City University (Li et al., 2015)
HKUST	Hong Kong University of Science and Technology (Lo et al., 2015)
ILLC-UvA	ILLC – University of Amsterdam (Stanojević and Sima'an, 2015)
METEOR-CMU	Carnegie Mellon University (Denkowski and Lavie, 2011)
USAAR-TUNA	Saarland University (Liling Tan and Mihaela Vela; no corresponding paper)

Table 3: Participants of WMT15 Tuning Shared Task

the WMT15 Translation Task (Bojar et al., 2015).

We leave all decoder settings (n-best list size, pruning limits etc.) at their default values. While the participants may have used different limits during tuning, the final test run was performed at our site with the default values. It is indeed only the feature weights that differ.

3 Tuning Task Participants

The list of participants and the names of the submitted systems are shown in Table 3, along with references to the details of each method.

USAAR-TUNA by Liling Tan and Mihaela Vela has no accompanying paper, so we sketch it here. The method sets each weight as the harmonic mean ($\frac{2xy}{x+y}$) of the weight proposed by batch MIRA and MERT. Batch MIRA and MERT are run side by side and the harmonic mean is taken and used in `moses.ini` at every iteration. The optimization stops when the averaged weights change only very little, which happened around iteration 17 or 18 in this case (Liling Tan, pc).

ILLC-UvA (Stanojević and Sima'an, 2015) was tuned using KBMIRA with modified version of BEER evaluation metric. The authors claim that standard trained evaluation metrics learn to give too much importance to recall and thus lead to overly long translations in tuning. For that reason they modify the training of BEER to value recall and precision equally. This modified version of BEER is used to train the MT system.

DCU (Li et al., 2015) is tuned with RED, an evaluation metric based on maching of dependency n-grams. Authors have tried tuning with both MERT and KBMIRA and found that KBMIRA gives better results so the submitted system uses KBMIRA.

HKUST (Lo et al., 2015) is with an improved version of MEANT. MEANT is an evaluation metric that pays more attention to semantic aspect of translation. Better correlation on the sentence level was achieved by integrating distributional semantics into MEANT and handling failures of the underlying semantic parser. The submission of HKUST contained a bug that was discovered after human evaluation period so the corrected submission HKUST-LATE is evaluated only with BLEU.

METEOR-CMU (Denkowski and Lavie, 2011) is a system tuned for an adapted version of Meteor. Meteor's parameters are set to give an equal importance to precision and recall.

AFRL (Erdmann and Gwinnup, 2015) is the only submission trained with a new tuning algorithm "Drem" instead of the standard MERT or KBMIRA. Drem uses scaled derivative-free trust-region optimization instead of line search or (sub)gradient approximations. For weight settings that were not tested in the decoder yet, it interpolates the decoder output using the information of which settings produced which translations. The optimized metric is a weighted combination of NIST, Meteor and Kendall's τ.

In addition to the systems submitted, we provided three baselines:

- BLEU-MERT-DENSE – MERT tuning with BLEU without additional features
- BLEU-MIRA-DENSE – KBMIRA tuning with BLEU without additional features
- BLEU-MIRA-SPARSE – KBMIRA tuning with BLEU with additional sparse features

Since all the submissions including the baselines were subject to manual evaluation, we did not run the MERT or MIRA optimizations more than once (as is the common practice for estimating variance due to optimizer instability). We simply used the default settings and stopping criteria and picked the weights that performed best on the dev set according to BLEU.

Of all the submissions, only the submission METEOR-CMU used sparse features. For a more interesting comparison, we set our baseline

(BLEU-MIRA-SPARSE) to use the very same set of sparse features. These features are automatically constructed using Moses' feature templates named PhraseLengthFeature0, SourceWordDeletionFeature0, TargetWordInsertionFeature0 and WordTranslationFeature0. They were made for the 50 most frequent words in the training data. For both language pairs these feature templates produce around 1000 features.

4 Results

We used the submitted `moses.ini` and (optionally) sparse `weights` files to translate the test set. The test set was not available to the participants at the time of their submission (not even the source side). We used the Moses recaser trained on the target side of the parallel corpus to recase the outputs of all the models.

Finally, the recased outputs were manually evaluated, jointly with regular translation task submissions of WMT (Bojar et al., 2015). This was not enough to reliably separate tuning systems in the Czech-to-English direction, so we asked task participants to provide some further rankings.

The resulting human rankings were used to compute the overall manual score using the TrueSkill method, same as for the main translation task (Bojar et al., 2015). We report two variants of the score: one is based on manual judgements related to tuning systems only and one is based on all judgements. Note that the actual ranking tasks shown to the annotators were identical, mixing tuning systems with regular submissions.

Tables 4 and 5 contain the results of the submitted systems sorted by their manual scores.

The horizonal lines represent separation between clusters of systems that perform similarly. Cluster boundaries are established by the same method as for the main translation task. Interestingly, cluster boundaries for Czech-to-English vary as we change the set of judgements.

Some systems do not have the TrueSkill score because they were either submitted after the deadline (HKUST-LATE) or served as additional baselines and performed similarly to our baselines (USAAR-BASELINE-MIRA and USAAR-BASELINE-MERT).

5 Discussion

There are a few interesting observations that can be made about the baseline results. Various details of the submissions including the exact weight settings are in Table 6.

System Name	TrueSkill Score		BLEU
	Tuning-Only	All	
BLEU-MIRA-DENSE	0.153	-0.182	12.28
ILLC-UVA	0.108	-0.189	12.05
BLEU-MERT-DENSE	0.087	-0.196	12.11
AFRL	0.070	-0.210	12.20
USAAR-TUNA	0.011	-0.220	12.16
DCU	-0.027	-0.263	11.44
METEOR-CMU	-0.101	-0.297	10.88
BLEU-MIRA-SPARSE	-0.150	-0.320	10.84
HKUST	-0.150	-0.320	10.99
HKUST-LATE	—	—	12.20

Table 4: Results on Czech-English tuning

System Name	TrueSkill Score		BLEU
	Tuning-Only	All	
DCU	0.320	-0.342	4.96
BLEU-MIRA-DENSE	0.303	-0.346	5.31
AFRL	0.303	-0.342	5.34
USAAR-TUNA	0.214	-0.373	5.26
BLEU-MERT-DENSE	0.123	-0.406	5.24
METEOR-CMU	-0.271	-0.563	4.37
BLEU-MIRA-SPARSE	-0.992	-0.808	3.79
USAAR-BASELINE-MIRA	—	—	5.31
USAAR-BASELINE-MERT	—	—	5.25

Table 5: Results on English-Czech tuning

5.1 Dense vs. Sparse Features

It is surprising how well the baseline based on KBMIRA and BLEU tuning (BLEU-MIRA-DENSE) performs on both language pairs. On Czech-English, it is better than all the other submitted systems while on English-Czech, only one system outperforms it (staying in the same performance cluster anyway).

Using BLEU-MIRA-DENSE for tuning dense features is becoming more common in the MT community, compared to the previous standard of using MERT. Our results confirm this practice. Preferring KBMIRA to MERT is often motivated by possibility to include sparse features, but we see that even for dense features only KBMIRA is better than MERT.

The sparse models, BLEU-MIRA-SPARSE and METEOR-CMU, however, perform rather poorly even though they were trained with KBMIRA. Both of the sparse submissions use the same set of features and the same tuning algorithm, although the optimization was run at different sites. The only difference is the metric they optimize. Tuning for Meteor (Denkowski and Lavie, 2011) gives better results than tuning for BLEU (Papineni et al., 2002). Unfortunately, we had no system with

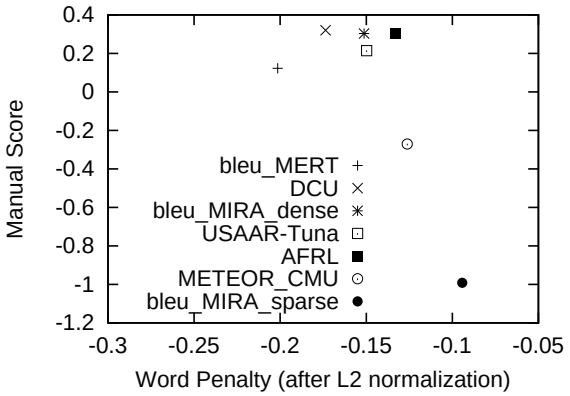

Figure 1: Relation between the word penalty and the final performance of systems translating from English to Czech.

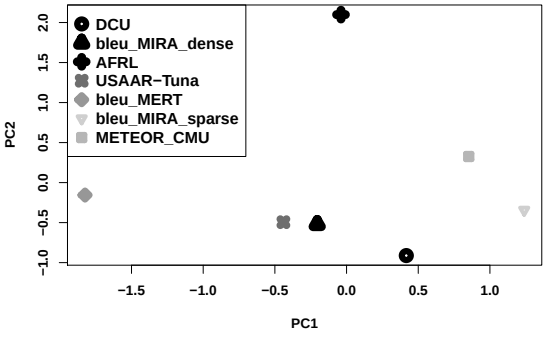

Figure 2: PCA for English-Czech. The darker the point, the higher the manual score.

dense features tuned for Meteor so we could not see if Meteor outperforms BLEU in the dense-only setting as well.

It is not clear why the sparse methods perform badly. One explanation could be the relatively small development set or some pruning settings. In any case, we find it unfortunate that sparse features in the hierarchical model harm performance in the default configuration[4].

5.2 Some Observations on Weight Settings

We tried to find some patterns in the weight settings and the performance of the system, but admittedly, it is difficult to make much sense of the few points in the 8-dimensional space.

For English-to-Czech, we can see a gist of a bell-like shape when normalizing the weights with L2 norm and plotting the word penalty and the

[4]MERT and two MIRA runs reached BLEU of not more than +0.02 points higher when the size of n-best list was increased from 100 to 200. So n-best list size does not seem to be the problem.

Czech-to-English	Type	Manual Score	Test BLEU	Dev BLEU	LM0	PhrPen	TM_0	TM_1	TM_2	TM_3	Glue	WrdPen
AFRL	dense	0.0700	12.20	14.83	0.1588	-0.3330	0.0545	0.0859	0.1958	0.1716	0.6309	-0.6227
bleu_MERT	dense	0.0870	12.11	14.64	0.0992	-0.0507	0.0688	0.0350	0.1296	0.0919	0.1820	-0.3428
bleu_MIRA_dense	dense	0.1530	12.28	14.85	0.0671	-0.1689	0.0363	0.0413	0.0747	0.0680	0.2982	-0.2454
bleu_MIRA_sparse	sparse	-0.1500	10.84	13.16	0.0906	-0.0568	0.0431	0.0556	0.0928	0.0933	0.3584	-0.2093
DCU	dense	-0.0270	11.44	13.58	0.0558	-0.1407	0.0360	0.0517	0.0856	0.0671	0.2481	-0.3150
HKUST_MEANT	dense	-0.1500	10.99	13.23	0.1333	0.0868	0.1318	0.0115	0.0534	0.1221	0.0500	-0.4110
HKUST_MEANT_LATE	dense	—	12.20	14.42	0.0638	-0.1696	0.0655	0.0217	0.0713	0.0677	0.3074	-0.2330
ILLC_UvA	dense	0.1080	12.05	14.57	0.0918	-0.1215	0.0452	0.0624	0.1103	0.0697	0.2295	-0.2696
METEOR_CMU	sparse	-0.1010	10.88	13.35	0.0936	-0.0103	0.0602	0.0509	0.1162	0.1187	0.2946	-0.2556
USAAR-Tuna	dense	0.0110	12.16	14.57	0.0789	-0.0715	0.0383	0.0575	0.1039	0.0744	0.1839	-0.2952

English-to-Czech	Type	Manual Score	Test BLEU	Dev BLEU	LM0	PhrPen	TM_0	TM_1	TM_2	TM_3	Glue	WrdPen
AFRL	dense	0.3030	5.34	6.96	0.0543	-0.4326	-0.0025	0.0382	0.2696	0.0788	0.8332	-0.1878
bleu_MERT	dense	0.1230	5.24	7.11	0.0510	-0.1353	0.0048	0.0169	0.1772	0.0408	0.3508	-0.2231
bleu_MIRA_dense	dense	0.3030	5.31	7.20	0.0380	-0.2046	-0.0004	0.0286	0.1338	0.0320	0.3936	-0.1689
bleu_MIRA_sparse	sparse	-0.9920	3.79	5.19	0.0364	-0.1232	-0.0053	0.0350	0.0905	0.0480	0.5524	-0.1093
DCU	dense	0.3200	4.96	6.87	0.0247	-0.1949	-0.0022	0.0367	0.1370	0.0345	0.3767	-0.1932
METEOR_CMU	sparse	-0.2710	4.37	5.86	0.0394	-0.0935	-0.0087	0.0331	0.1611	0.0673	0.4548	-0.1421
Saarland_baseline_mert	dense	—	5.25	7.16	0.0394	-0.1619	-0.0011	0.0218	0.1947	0.0211	0.3973	-0.1628
Saarland_baseline_mira	dense	—	5.31	7.11	0.0377	-0.2023	-0.0007	0.0293	0.1304	0.0344	0.3936	-0.1714
USAAR-Tuna	dense	0.2140	5.26	7.15	0.0386	-0.1799	-0.0008	0.0250	0.1562	0.0262	0.3954	-0.1670

Table 6: Detailed scores and weights of Czech-to-English (left) and English-to-Czech (right) systems.

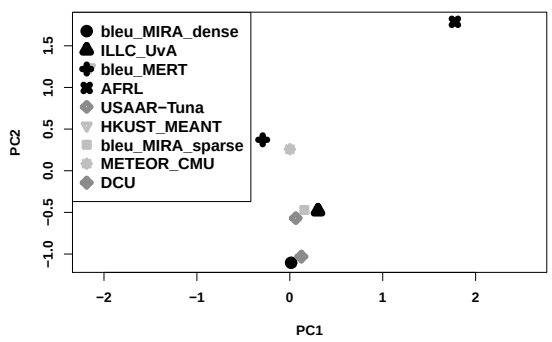

Figure 3: PCA for Czech-English. The darker the point, the higher the manual score.

	PC1	PC2
LM0	**-0.69**	0.44
PhrasePenalty0	0.15	**-0.63**
TranslationModel0_0	**-0.91**	-0.13
TranslationModel0_1	**0.91**	-0.03
TranslationModel0_2	-0.55	**0.72**
TranslationModel0_3	0.36	**0.75**
TranslationModel1	0.42	**0.84**
WordPenalty0	**0.84**	0.27

Table 7: Loadings (correlations) of each component with each feature function for English-Czech

manual score, see Figure 1. The middle values seemed to be a good setting. For the other translation direction or other weights, no such clear relation is apparent.

We tried to interpret the weight settings also using principal component analysis (PCA), despite the low number of observations. (Ideally, we would like to have at least 40–80 systems, we have 7 or 9). Before running PCA, we normalized the weights with L2 norm. After running Cattell Scree test, the results showed that two components would be appropriate to summarize the dataset. To make components more interpretable, we applied varimax rotation.

Figure 2 plots the two principal components of the set of systems for English-to-Czech. We see that the first component (PC1) explains the performance almost completely with middle values being the best. Looking at loadings (correlations of components with the original feature function dimensions) in Table 7, we learn, that PC1 primarily accounts for the first two weights of translation model (TM_0 and TM_1, which correspond to phrase and lexically-weighted inverse probabilities, resp.) and the word penalty (WrdPen) and language model weight (LM0). Knowing that in almost all systems the weight of word penalty is several times bigger than weights of TM_0, TM_1, and LM0, we conclude that tuning of word penalty (in balance with LM weight) was the most apparent decisive factor of English-Czech tuning task. The second component (PC2) primarily covers the weights of the remaining features, that is the direct translation probabilites and phrase penalty. Unfortunately, PC2 is not very informative about the final quality of the translation.

The Czech-to-English results in Figure 3 do not seem to lend themselves to any simple conclusion.

Based on closeness of systems in the PCA plots, we can say that for English-Czech, two out of three best systems (BLEU-MIRA-DENSE and DCU) found similar settings while AFRL stands out. Czech-English results show that systems of very similar weight settings give translations of very different quality. Again, AFRL stands out while leading to very good outputs.

6 Conclusion

This paper presented the WMT shared task in optimizing parameters of a given hierarchical phrase-based system (WMT Tuning Task) when translating from English to Czech and vice versa. The underlying system was intentionally restricted to small data setting and somewhat unusually, the data for the language model were smaller than for the translation model.

Overall, six teams took part in one or both directions, sticking to the constrained setting, with only METEOR-CMU and our baseline BLEU-MIRA-SPARSE using sparse features.

The submitted configurations were manually evaluated jointly with the systems of the main WMT translation task. Given the small data setting, we did not expect the tuning task systems to perform competitively to other submissions in the WMT translation task.

The results confirm that KBMIRA with the standard (dense) features optimized towards BLEU should be preferred over MERT. Two other systems (DCU and AFRL) performed equally well in English-to-Czech translation. The two systems using sparse features (METEOR-CMU and BLEU-MIRA-SPARSE) performed poorly, but the sample is too small to draw any conclusions from this. Overall, the variance in translation quality obtained using various weight settings is apparent and justifies the efforts put into optimization tech-

niques.

Since the task attracted a good number of submissions and was generally considered interesting and useful by our colleagues, we plan to run the task again for WMT in 2016. The next year's underlying systems will use all data available in the WMT constraint setting, to test the tuning methods in the range where state-of-the-art systems operate.

Acknowledgments

We are grateful to Christian Federmann and Matt Post for all the processing of human evaluation and to the annotators who quickly helped us in getting additional judgements. Thanks also go to Matthias Huck for a thorough check of the paper, all outstanding errors are our own. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements n^o 645452 (QT21) and n^o 644402 (HimL). The work on this project was also supported by the Dutch organisation for scientific research STW grant nr. 12271.

References

Meriem Beloucif, Chi-kiu Lo, and Dekai Wu. 2014. Improving MEANT Based Semantically Tuned SMT. In *Proc. of 11th International Workshop on Spoken Language Translation (IWSLT 2014)*, pages 34–41, Lake Tahoe, California.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July. Association for Computational Linguistics. Revised August 2010.

Daniel Cer, Christopher D. Manning, and Daniel Jurafsky. 2010. The best lexical metric for phrase-based statistical mt system optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 555–563, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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, June.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July. Association for Computational Linguistics.

Grant Erdmann and Jeremy Gwinnup. 2015. Drem: The AFRL Submission to the WMT15 Tuning Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *In Proc. of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

Hieu Hoang, Philipp Koehn, and Adam Lopez. 2009. A Unified Framework for Phrase-Based, Hierarchical, and Syntax-Based Statistical Machine Translation. In *Proceedings of IWSLT*, pages 152–159, Tokyo, Japan, December.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1352–1362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),*, volume 1, pages 181–184. IEEE.

Liangyou Li, Hui Yu, and Qun Liu. 2015. MT Tuning on RED: A Dependency-Based Evaluation Metric. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2011. Better Evaluation Metrics Lead to Better Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 375–384, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chi-kiu Lo, Philipp Dowling, and Dekai Wu. 2015. Improving evaluation and optimization of MT systems against MEANT. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 on Association for Computational Linguistics*, ACL '02, pages 311–318.

Miloš Stanojević and Khalil Sima'an. 2015. BEER 1.1: ILLC UvA submission to metrics and tuning task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 Metrics Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Bing Zhao and Shengyuan Chen. 2009. A simplex armijo downhill algorithm for optimizing statistical machine translation decoding parameters. In *HLT-NAACL (Short Papers)*, pages 21–24. The Association for Computational Linguistics.