BEER 1.1: ILLC UvA submission to metrics and tuning task

Miloš Stanojević
ILLC
University of Amsterdam
m.stanojevic@uva.nl

Khalil Sima'an
ILLC
University of Amsterdam
k.simaan@uva.nl

Abstract

We describe the submissions of ILLC UvA to the metrics and tuning tasks on WMT15. Both submissions are based on the BEER evaluation metric originally presented on WMT14 (Stanojević and Sima'an, 2014a). The main changes introduced this year are: (i) extending the learning-to-rank trained sentence level metric to the corpus level (but still decomposable to sentence level), (ii) incorporating syntactic ingredients based on dependency trees, and (iii) a technique for finding parameters of BEER that avoid "gaming of the metric" during tuning.

1 Introduction

In the 2014 WMT metrics task, BEER turned up as the best sentence level evaluation metric on average over 10 language pairs (Machacek and Bojar, 2014). We believe that this was due to:

1. *learning-to-rank* - type of training that allows a large number of features and also training on the same objective on which the model is going to be evaluated : ranking of translations

2. *dense features* - character n-grams and skip-bigrams that are less sparse on the sentence level than word n-grams

3. *permutation trees* - hierarchical decomposition of word order based on (Zhang and Gildea, 2007)

A deeper analysis of (2) is presented in (Stanojević and Sima'an, 2014c) and of (3) in (Stanojević and Sima'an, 2014b).

Here we modify BEER by

1. incorporating a better scoring function that give scores that are better scaled

2. including syntactic features and

3. removing the recall bias from BEER .

In Section 2 we give a short introduction to BEER after which we move to the innovations for this year in Sections 3, 4 and 5. We show the results from the metric and tuning tasks in Section 6, and conclude in Section 7.

2 BEER basics

The model underying the BEER metric is flexible for the integration of an arbitrary number of new features and has a training method that is targeted for producing good rankings among systems. Two other characteristic properties of BEER are its hierarchical reordering component and character n-grams lexical matching component.

2.1 Old BEER scoring

BEER is essentially a linear model with which the score can be computed in the following way:

$$score(h,r) = \sum_i w_i \times \phi_i(h,r) = \vec{\boldsymbol{w}} \cdot \vec{\boldsymbol{\phi}}$$

where $\vec{\boldsymbol{w}}$ is a weight vector and $\vec{\boldsymbol{\phi}}$ is a feature vector.

2.2 Learning-to-rank

Since the task on which our model is going to be evaluated is ranking translations it comes natural to train the model using *learning-to-rank* techniques.

Our training data consists of pairs of "good" and "bad" translations. By using a feature vector $\vec{\phi}_{good}$ for a good translation and a feature vector $\vec{\phi}_{bad}$ for a bad translation then using the following equations we can transform the ranking problem into a binary classification problem (Herbrich et al., 1999):

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 396–401,
Lisboa, Portugal, 17-18 September 2015. ©2015 Association for Computational Linguistics.

$$
\begin{aligned}
score(h_{good}, r) > score(h_{bad}, r) &\Leftrightarrow \\
\vec{w} \cdot \vec{\phi}_{good} > \vec{w} \cdot \vec{\phi}_{bad} &\Leftrightarrow \\
\vec{w} \cdot \vec{\phi}_{good} - \vec{w} \cdot \vec{\phi}_{bad} > 0 &\Leftrightarrow \\
\vec{w} \cdot (\vec{\phi}_{good} - \vec{\phi}_{bad}) > 0 & \\
\vec{w} \cdot (\vec{\phi}_{bad} - \vec{\phi}_{good}) < 0 &
\end{aligned}
$$

If we look at $\vec{\phi}_{good} - \vec{\phi}_{bad}$ as a positive training instance and at $\vec{\phi}_{bad} - \vec{\phi}_{good}$ as a negative training instance, we can train any linear classifier to find weight the weight vector $\vec{w}$ that minimizes mistakes in ranking on the training set.

2.3 Lexical component based on character n-grams

Lexical scoring of BEER relies heavily on character n-grams. Precision, Recall and F1-score are used with character n-gram orders from 1 until 6. These scores are more smooth on the sentence level than word n-gram matching that is present in other metrics like BLEU (Papineni et al., 2002) or METEOR (Michael Denkowski and Alon Lavie, 2014).

BEER also uses precision, recall and F1-score on word level (but not with word n-grams). Matching of words is computed over METEOR alignments that use WordNet, paraphrasing and stemming to have more accurate alignment.

We also make distinction between function and content words. The more precise description of used features and their effectiveness is presented in (Stanojević and Sima'an, 2014c).

2.4 Reordering component based on PETs

The word alignments between system and reference translation can be simplified and considered as permutation of words from the reference translation in the system translation. Previous work by (Isozaki et al., 2010) and (Birch and Osborne, 2010) used this permutation view of word order and applied Kendall τ for evaluating its distance from ideal (monotone) word order.

BEER goes beyond this *skip-gram* based evaluation and decomposes permutation into a hierarchical structure which shows how subparts of permutation form small groups that can be reordered all together. Figure 1a shows PET for permutation $\langle 2, 5, 6, 4, 1, 3 \rangle$. Ideally the permutation tree will be filled with nodes $\langle 1, 2 \rangle$ which would say that there is no need to do any reordering (everything is in the right place). BEER has features that compute the number of different node types and for each different type it assigns a different weight. Sometimes there are more than one PET for the same permutation. Consider Figure 1b and 1c which are just 2 out of 3 possible PETs for permutation $\langle 4, 3, 2, 1 \rangle$. Counting the number of trees that could be built is also a good indicator of the permutation quality. See (Stanojević and Sima'an, 2014b) for details on using PETs for evaluating word order.

3 Corpus level BEER

Our goal here is to create corpus level extension of BEER that decomposes trivially at the sentence level. More concretely we wanted to have a corpus level BEER that would be the average of the sentence level BEER of all sentences in the corpus:

$$BEER_{corpus}(c) = \frac{\sum_{s_i \in c} BEER_{sent}(s_i)}{|c|} \tag{1}$$

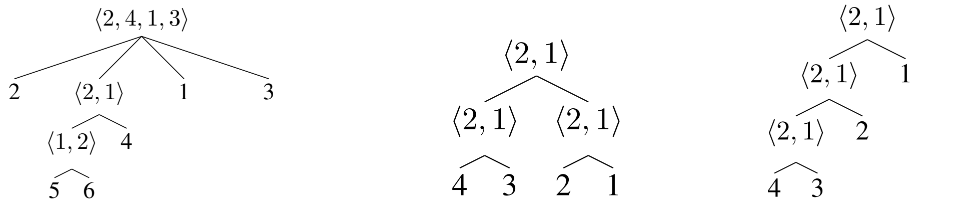
In order to do so it is not suitable to to use previous scoring function of BEER . The previous scoring function (and training method) take care only that the better translation gets a higher score than the worse translation (on the sentence level). For this kind of corpus level computations we have an additional requirement that our sentence level scores need to be scaled proportional to the translation quality.

3.1 New BEER scoring function

To make the scores on the sentence level better scaled we transform our linear model into a probabilistic linear model – logistic regression with the following scoring function:

$$score(h, r) = \frac{1}{1 + e^{-\sum_i w_i \times \phi_i(h,r)}}$$

There is still a problem with this formulation. During training, the model is trained on the difference between two feature vectors $\vec{\phi}_{good} - \vec{\phi}_{bad}$, while during testing it is applied only to one feature vector $\vec{\phi}_{test}$. $\vec{\phi}_{good} - \vec{\phi}_{bad}$ is usually very close to the separating hyperplane, whereas $\vec{\phi}_{test}$ is usually very far from it. This is not a problem for ranking but it presents a problem if we want well scaled scores. Being extremely far from the



(a) Complex PET (b) Fully inverted PET 1 (c) Fully inverted PET 2

Figure 1: Examples of PETs

separated hyperplane gives extreme scores such as 0.9999999999912 and 0.0000000000000213 as a result which are obviously not well scaled.

Our model was trained to give a probability of the "good" translation being better than the "bad" translation so we should also use it in that way – to estimate the probability of one translation being better than the other. But which translation? We are given only one translation and we need to compute its score. To avoid this problem we pretend that we are computing a probability of the test sentence being a better translation than the reference for the given reference. In the ideal case the system translation and the reference translation will have the same features which will make logistic regression output probability 0.5 (it is uncertain about which translation is the better one). To make the scores between 0 and 1 we multiply this result with 2. The final scoring formula is the following:

$$score(h,r) = \frac{2}{1 + e^{-\sum_i w_i \times (\phi_i(h,r) - \phi_i(r,r))}}$$

4 BEER + Syntax = BEER_Treepel

The standard version of BEER does not use any syntactic knowledge. Since the training method of BEER allows the usage of a large number of features, it is trivial to integrate new features that would measure the matching between some syntax attributes of system and reference translations.

The syntactic representation we exploit is a dependency tree. The reason for that is that we can easily connect the structure with the lexical content and it is fast to compute which can often be very important for evaluation metrics when they need to evaluate on large data. We used Stanford's dependency parser (Chen and Manning, 2014) because it gives high accuracy parses in a very short time.

The features we compute on the dependency trees of the system and its reference translation are:

1. POS bigrams matching
2. dependency words bigram matching
3. arc type matching
4. valency matching

For each of these we compute precision, recall and F1-score.

It has been shown by other researchers (Popović and Ney, 2009) that POS tags are useful for abstracting away from concrete words and measure the grammatical aspect of translation (for example it can captures agreement).

Dependency word bigrams (bigrams connected by a dependency arc) are also useful for capturing long distance dependencies.

Most of the previous metrics that work with dependency trees usually ignore the type of the dependency that is (un)matched and treat all types equally (Yu et al., 2014). This is clearly not the case. Surely subject and complement arcs are more important than modifier arc. To capture this we created individual features for precision, recall and F1-score matching of each arc type so our system could learn on which arc type to put more weight.

All words take some number of arguments (valency), and not matching that number of arguments is a sign of a, potentially, bad translation. With this feature we hope to capture the aspect of not producing the right number of arguments for all words (and especially verbs) in the sentence.

This model BEER_Treepel contains in total 177 features out of which 45 are from original BEER .

5 BEER for tuning

The metrics that perform well on metrics task are very often not good for tuning. This is because recall has much more importance for human judgment than precision. The metrics that put more weight on recall than precision will be better with

tuning metric	BLEU	MTR	BEER	Length
BEER	16.4	**28.4**	**10.2**	115.7
BLEU	**18.2**	28.1	10.1	103.0
BEER_no_bias	18.0	27.7	9.8	**99.7**

Table 1: Tuning results with BEER without bias on WMT14 as tuning and WMT13 as test set

correlation with human judgment, but when used for tuning they will create overly long translations.

This bias for long translation is often resolved by manually setting the weights of recall and precision to be equal (Denkowski and Lavie, 2011; He and Way, 2009).

This problem is even bigger with metrics with many features. When we have metric like BEER_Treepel which has 117 features it is not clear how to set weights for each feature manually. Also some features might not have easy interpretation as precision or recall of something. Our method for automatic removing of this recall bias, which is presented in (Stanojević, 2015), gives very good results that can be seen in Table 1.

Before the automatic adaptation of weights for tuning, tuning with standard BEER produces translations that are 15% longer than the reference translations. This behavior is rewarded by metrics that are recall-heavy like METEOR and BEER and punished by precision heavy metrics like BLEU. After automatic adaptation of weights, tuning with BEER matches the length of reference translation even better than BLEU and achieves the BLEU score that is very close to tuning with BLEU. This kind of model is disliked by METEOR and BEER but by just looking at the length of the produced translations it is clear which approach is preferred.

6 Metric and Tuning task results

The results of WMT15 metric task of best performing metrics is shown in Tables 2 and 3 for the system level and Tables 4 and 5 for segment level.

On the sentence level for out of English language pairs on average BEER was the best metric (same as the last year). Into English it got 2nd place with its syntactic version and 4th place as the original BEER .

On the corpus level BEER is on average second for out of English language pairs and 6th for into English. BEER and BEER_Treepel are the best for en-ru and fi-en.

System Name	**TrueSkill Score**		**BLEU**
	Tuning-Only	All	
BLEU-MIRA-DENSE	0.153	-0.177	12.28
ILLC-UvA	0.108	-0.188	12.05
BLEU-MERT-DENSE	0.087	-0.200	12.11
AFRL	0.070	-0.205	12.20
USAAR-Tuna	0.011	-0.220	12.16
DCU	-0.027	-0.256	11.44
METEOR-CMU	-0.101	-0.286	10.88
BLEU-MIRA-SPARSE	-0.150	-0.331	10.84
HKUST	-0.150	-0.331	10.99
HKUST-LATE	—	—	12.20

Table 6: Results on Czech-English tuning

The difference between BEER and BEER_Treepel are relatively big for de-en, cs-en and ru-en while for fr-en and fi-en the difference does not seem to be big.

The results of WMT15 tuning task is shown in Table 6. The system tuned with BEER without recall bias was the best submitted system for Czech-English and only the strong baseline outperformed it.

7 Conclusion

We have presented ILLC UvA submission to the shared metric and tuning task. All submissions are centered around BEER evaluation metric. On the metrics task we kept the good results we had on sentence level and extended our metric to corpus level with high correlation with high human judgment without losing the decomposability of the metric to the sentence level. Integration of syntactic features gave a bit of improvement on some language pairs. The removal of recall bias allowed us to go from overly long translations produced in tuning to translations that match reference relatively close by length and won the 3rd place in the tuning task. BEER is available at `https://github.com/stanojevic/beer`.

Acknowledgments

This work is supported by STW grant nr. 12271 and NWO VICI grant nr. 277-89-002. QT21 project support to the second author is also acknowledged (European Unions Horizon 2020 grant agreement no. 64545). We are thankful to Christos Louizos for help with incorporating a dependency parser to BEER Treepel.

Correlation coefficient	Pearson Correlation Coefficient					
Direction	fr-en	fi-en	de-en	cs-en	ru-en	Average
DPMFCOMB	.995 ± .006	.951 ± .013	**.949** ± .016	**.992** ± .004	.871 ± .025	**.952** ± .013
RATATOUILLE	.989 ± .010	.899 ± .019	.942 ± .018	.963 ± .008	**.941** ± .018	.947 ± .014
DPMF	**.997** ± .005	.939 ± .015	.929 ± .019	.986 ± .005	.868 ± .026	.944 ± .014
METEOR-WSD	.982 ± .011	.944 ± .014	.914 ± .021	.981 ± .006	.857 ± .026	.936 ± .016
CHRF3	.979 ± .012	.893 ± .020	.921 ± .020	.969 ± .007	.915 ± .023	.935 ± .016
BEER_TREEPEL	.981 ± .011	**.957** ± .013	.905 ± .021	.985 ± .005	.846 ± .027	.935 ± .016
BEER	.979 ± .012	.952 ± .013	.903 ± .022	.975 ± .006	.848 ± .027	.931 ± .016
CHRF	.997 ± .005	.942 ± .015	.884 ± .024	.982 ± .006	.830 ± .029	.927 ± .016
LEBLEU-OPTIMIZED	.989 ± .009	.895 ± .020	.856 ± .025	.970 ± .007	.918 ± .023	.925 ± .017
LEBLEU-DEFAULT	.960 ± .015	.895 ± .020	.856 ± .025	.946 ± .010	.912 ± .022	.914 ± .018

Table 2: System-level correlations of automatic evaluation metrics and the official WMT human scores when translating into English.

Correlation coefficient	Pearson Correlation Coefficient					
Metric	en-fr	en-fi	en-de	en-cs	en-ru	Average
CHRF3	.949 ± .021	**.813** ± .025	.784 ± .028	**.976** ± .004	.913 ± .011	**.887** ± .018
BEER	.970 ± .016	.729 ± .030	.811 ± .026	.951 ± .005	**.942** ± .009	.880 ± .017
LEBLEU-OPTIMIZED	.949 ± .020	.727 ± .030	**.896** ± .020	.944 ± .005	.867 ± .013	.877 ± .018
LEBLEU-DEFAULT	.949 ± .020	.760 ± .028	.827 ± .025	.946 ± .005	.849 ± .014	.866 ± .018
RATATOUILLE	.962 ± .017	.675 ± .031	.777 ± .028	.953 ± .005	.869 ± .013	.847 ± .019
CHRF	.949 ± .021	.771 ± .027	.572 ± .037	.968 ± .004	.871 ± .013	.826 ± .020
METEOR-WSD	.961 ± .018	.663 ± .032	.495 ± .039	.941 ± .005	.839 ± .014	.780 ± .022
BS	−.977 ± .014	.334 ± .039	−.615 ± .036	−.947 ± .005	−.791 ± .016	−.600 ± .022
DPMF	**.973** ± .015	n/a	.584 ± .037	n/a	n/a	.778 ± .026

Table 3: System-level correlations of automatic evaluation metrics and the official WMT human scores when translating out of English.

Direction	fr-en	fi-en	de-en	cs-en	ru-en	Average
DPMFCOMB	.367 ± .015	**.406** ± .015	**.424** ± .015	**.465** ± .012	**.358** ± .014	**.404** ± .014
BEER_TREEPEL	.358 ± .015	.399 ± .015	.386 ± .016	.435 ± .013	.352 ± .013	.386 ± .014
RATATOUILLE	**.367** ± .015	.384 ± .015	.380 ± .015	.442 ± .013	.336 ± .014	.382 ± .014
BEER	.359 ± .015	.392 ± .015	.376 ± .015	.417 ± .013	.336 ± .013	.376 ± .014
METEOR-WSD	.347 ± .015	.376 ± .015	.360 ± .015	.416 ± .013	.331 ± .014	.366 ± .014
CHRF	.350 ± .015	.378 ± .015	.366 ± .016	.407 ± .013	.322 ± .014	.365 ± .014
DPMF	.344 ± .014	.368 ± .015	.363 ± .015	.413 ± .013	.320 ± .014	.362 ± .014
CHRF3	.345 ± .014	.361 ± .016	.360 ± .015	.409 ± .012	.317 ± .014	.359 ± .014
LEBLEU-OPTIMIZED	.349 ± .015	.346 ± .015	.346 ± .014	.400 ± .013	.316 ± .015	.351 ± .014
LEBLEU-DEFAULT	.343 ± .015	.342 ± .015	.341 ± .014	.394 ± .013	.317 ± .014	.347 ± .014
TOTAL-BS	−.305 ± .013	−.277 ± .015	−.287 ± .014	−.357 ± .013	−.263 ± .014	−.298 ± .014

Table 4: Segment-level Kendall's τ correlations of automatic evaluation metrics and the official WMT human judgments when translating into English. The last three columns contain average Kendall's τ computed by other variants.

Direction	en-fr	en-fi	en-de	en-cs	en-ru	Average
BEER	.323 ± .013	**.361** ± .013	**.355** ± .011	**.410** ± .008	**.415** ± .012	**.373** ± .011
CHRF3	.309 ± .013	.357 ± .013	.345 ± .011	.408 ± .008	.398 ± .012	.363 ± .012
RATATOUILLE	**.340** ± .013	.300 ± .014	.337 ± .011	.406 ± .008	.408 ± .012	.358 ± .012
LEBLEU-DEFAULT	.321 ± .013	.354 ± .013	.345 ± .011	.385 ± .008	.386 ± .012	.358 ± .011
LEBLEU-OPTIMIZED	.325 ± .013	.344 ± .012	.345 ± .012	.383 ± .008	.385 ± .012	.356 ± .011
CHRF	.317 ± .013	.346 ± .012	.315 ± .013	.407 ± .008	.387 ± .012	.355 ± .012
METEOR-WSD	.316 ± .013	.270 ± .013	.287 ± .012	.363 ± .008	.373 ± .012	.322 ± .012
TOTAL-BS	−.269 ± .013	−.205 ± .012	−.231 ± .011	−.324 ± .008	−.332 ± .012	−.273 ± .011
DPMF	.308 ± .013	n/a	.289 ± .012	n/a	n/a	.298 ± .013
PARMESAN	n/a	n/a	n/a	.089 ± .006	n/a	.089 ± .006

Table 5: Segment-level Kendall's τ correlations of automatic evaluation metrics and the official WMT human judgments when translating out of English. The last three columns contain average Kendall's τ computed by other variants.

References

Alexandra Birch and Miles Osborne. 2010. LRscore for Evaluating Lexical and Reordering Quality in MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 327–332, Uppsala, Sweden, July. Association for Computational Linguistics.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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*, WMT '11, pages 85–91, Stroudsburg, PA, USA. Association for Computational Linguistics.

Y. He and A. Way. 2009. Improving the objective function in minimum error rate training. *Proceedings of the Twelfth Machine Translation Summit*, pages 238–245.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 1999. Support Vector Learning for Ordinal Regression. In *In International Conference on Artificial Neural Networks*, pages 97–102.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 944–952, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matous Machacek and Ondrej Bojar. 2014. Results of the wmt14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the ACL 2014 Workshop on Statistical Machine Translation*.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Maja Popović and Hermann Ney. 2009. Syntax-oriented evaluation measures for machine translation output. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 29–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Miloš Stanojević and Khalil Sima'an. 2014a. BEER: BEtter Evaluation as Ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Miloš Stanojević and Khalil Sima'an. 2014b. Evaluating Word Order Recursively over Permutation-Forests. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 138–147, Doha, Qatar, October. Association for Computational Linguistics.

Miloš Stanojević and Khalil Sima'an. 2014c. Fitting Sentence Level Translation Evaluation with Many Dense Features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206, Doha, Qatar, October. Association for Computational Linguistics.

Miloš Stanojević. 2015. Removing Biases from Trainable MT Metrics by Using Self-Training. *arXiv preprint arXiv:1508.02445*.

Hui Yu, Xiaofeng Wu, Jun Xie, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2014. Red: A reference dependency based mt evaluation metric. In *COLING'14*, pages 2042–2051.

Hao Zhang and Daniel Gildea. 2007. Factorization of synchronous context-free grammars in linear time. In *In NAACL Workshop on Syntax and Structure in Statistical Translation (SSST*.