Log-linear weight optimization using discriminative ridge regression method in statistical machine translation

Mara Chinea-Rios[1,*], Germán Sanchis-Trilles[2], and Francisco Casacuberta[1]

[1] Pattern Recognition and Human Language Technology Research Center,
Universitat Politècnica de València , 46022 Valencia, Spain
{machirio,fcn}@prhlt.upv.es
[2] Sciling, Valencia, Spain gsanchis@sciling.es

Abstract. We present a simple and reliable method for estimating the log-linear weights of a state-of-the-art machine translation system, which takes advantage of the method known as discriminative ridge regression (DRR). Since inappropriate weight estimations lead to a wide variability of translation quality results, reaching a reliable estimate for such weights is critical for machine translation research. For this reason, a variety of methods have been proposed to reach reasonable estimates. In this paper, we present an algorithmic description and empirical results proving that DRR, as applied in a pseudo-batch scenario, is able to provide comparable translation quality when compared to state-of-the-art estimation methods (i.e., MERT [1] and MIRA[2]). Moreover, the empirical results reported are coherent across different corpora and language pairs. **Keywords:** statistical machine translation, log-linear model, discriminative ridge regression

1 Introduction

One important breakthrough in Statistical Machine Translation (SMT) was provided by the use of log-linear models for modelling the translation process [3, 4]. The log-linear models are defined as follow: given a source sentence $\mathbf{f} = f_1, \ldots, f_j, \ldots, f_J$ which is to be translated into a target sentence $\mathbf{e} = e_1, \ldots, e_i, \ldots, e_I$.

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} Pr(\mathbf{e} \mid \mathbf{f}) = \operatorname*{argmax}_{\mathbf{e}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}, \mathbf{e}) = \operatorname*{argmax}_{\mathbf{e}} \boldsymbol{\lambda} \cdot \mathbf{h}(\mathbf{f}, \mathbf{e}) \tag{1}$$

In this framework, we have a set of M features function $h_m(\mathbf{f}, \mathbf{e}), m = 1, \cdots, M$. For each function, there exists a weight parameter $\boldsymbol{\lambda}_m, m = 1, \cdots, M$. Common feature functions $h_m(\mathbf{f}, \mathbf{e})$ include different translation models (TM), but also distortion models or even the target language model (LM). Typically, $h(\cdot, \cdot)$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]$ are estimated by means of training and development sets, respectively.
The use of log-linear models implied an important break-through in SMT, allowing for a significant increase in the quality of the translations produced. The problem then arises of how to optimize the weights $\boldsymbol{\lambda}$, in other words how to find a set of weights which will offer the best translation quality. In this work, we used the Discriminative Ridge Regression [5] technique for estimating the weights of such log-linear models according to a development data set.

The main contributions of this paper are:

- We present an algorithmic description of Discriminative Ridge Regression in a batch setting, as applied to estimating the log-linear weights of a state-of-the-art.
- We evaluate empirically the DRR algorithm proposed in two different domains and with two different language pairs.
- We provide a thorough comparison with state-of-the-art $\boldsymbol{\lambda}$ estimation methods, such as Minimum Error Rate Training (MERT) [1], and batch Margin Infused Relaxed Algorithm (MIRA) [6].

The rest of this paper is structured as follows. In Section 2, we perform a brief review of current approaches to log-linear weight estimation in SMT. In Section 3, we describe the algorithmic approach for applying DDR in a batch scenario for estimating $\boldsymbol{\lambda}$. In Section 4, the experimental design and empirical results are detailed. Conclusions and future work are explained in Section 5

2 Related work

Once the bilingual phrases have been extracted from a sentence aligned bilingual corpus, the features $\mathbf{h}$ can already be computed. However, at this point it is still necessary to obtain an appropriate value for the scaling factors $\boldsymbol{\lambda}$. The process of obtaining such a vector is often called *tuning*. To this end, numerous methods have been proposed.

The most popular approach for adjusting the scaling factors is the one proposed in [1], commonly referred to as Minimum Error Rate Training (MERT). This algorithm implements a coordinate-wise global optimisation and consists on two basic steps. However, such algorithm has an important drawback. Namely, it requires a considerable amount of time to translate the development set several times, and in addition it has been shown to be quite unstable whenever the amount of adaptation data is small [7].

Various alternatives to MERT have been proposed, motivated primarily by scalability considerations. One popular alternative is the use of margin infused MIRA [2], [6] which is a perception-like online tuning algorithm with passive-aggressive updates. Tellingly, in the entire proceedings of ACL 2015[3], only one paper describing a statistical MT system cited the use of MIRA for tuning [8], while the others used MERT.

Alternatively, [9] proposed to view the problem as a ranking problem (PRO), where each step of the tuning procedure consists in deciding whether a given translation hypothesis should be ranked lower or higher within the set of possible hypotheses that are provided by the search procedure.

3 Discriminative rigde regression for SMT

In this section, the Discriminative Ridge Regression method is for estimating $\boldsymbol{\lambda}$ is reviewed. DRR was proposed by [5], uses the concept of ridge regression technique to develop a discriminative algorithm for estimating $\boldsymbol{\lambda}$ online, i.e., as new adaptation samples are introduced into the system. The key idea is to to find a configuration of the weight vectors using all the hypotheses within a given N-best list, so that good hypothesis are rewarded, and bad hypothesis are penalised, trying to narrow the correlation between the score function σ, and the quality criterion used. Since DRR was proposed for an online computer-assisted translation scenario, it requires an N-best list of hypotheses for

[3] `www.aclweb.org/anthology/P/P15/`

each one of the sentences that are evaluated by the professional translator post-editing the system's output. Algorithm 1 shows the procedure. Here, $A = \{\mathbf{f}1, \dots, \mathbf{f}_s, \dots, \mathbf{f}_S\}$ is a bilingual development corpus, S is the number of sentences in A and $s \in S$, and I is the maximum number of epochs desired.

```
Data: Development corpus $A$
Result: $\boldsymbol{\lambda}$
Initialize: $\boldsymbol{\lambda}^0$;
forall desired number of iterations $I$ do
    forall number sentences in dev-corpus $S = |A|$ do
        optimization: compute gradient vector $\check{\boldsymbol{\lambda}}_i^s$;
        estimation: $\boldsymbol{\lambda}_i^s = (1-\alpha)\boldsymbol{\lambda}_i^{s-1} + \alpha\check{\boldsymbol{\lambda}}_i^s$;
    end
end
selection: output vector $\boldsymbol{\lambda}_I^S$
```

Algorithm 1: Pseudo-code for DRR estimating $\boldsymbol{\lambda}$ as described in Section 3

During the **optimization** step, we obtain the gradient vector $\check{\boldsymbol{\lambda}}$ for each one of the development sentences a_s. Within DRR, this optimisation is performed by computing the solution to an overdetermined system, described in detail in next section, so that changes in the scoring function σ are correlated to changes in the objective function (potentially BLEU).

3.1 Sentence-based optimisation in DRR

As exposed in the previous section, DRR obtains $\boldsymbol{\lambda}$ based on obtaining the best log-gradient vector for each one of the sentences of a development corpus. In order to compute the new log-linear weight vector $\boldsymbol{\lambda}^s$, the previously learned $\boldsymbol{\lambda}^{s-1}$ needs to be combined with an appropriate update step $\check{\boldsymbol{\lambda}}^s$. The aim is to compute an appropriate update term $\boldsymbol{\lambda}^s$ that best fits the translation search space (approximated as an n-best list) of the development sentence pair observed at s. This is often done as a linear combination [10], where $\boldsymbol{\lambda}^s = (1-\alpha)\boldsymbol{\lambda}^{s-1} + \alpha\check{\boldsymbol{\lambda}}^s$ for a certain learning learning rate α. Let n-best($\mathbf{f}$) be such a list computed by our models for sentence $\mathbf{f}$. To obtain $\check{\boldsymbol{\lambda}}^s$, we define an $N \times M$ matrix $H_\mathbf{f}$ that contains the feature functions $\mathbf{h}$ of every hypothesis, where M is the number of features in Equation 1, and N is the size of n-best($\mathbf{f}$).

$$H_\mathbf{f} = [\mathbf{h}(\mathbf{f}, \mathbf{e}_1), \dots, \mathbf{h}(\mathbf{f}, \mathbf{e}_N)]' \tag{2}$$

Additionally, let $H^*_\mathbf{f}$ be a matrix such that

$$H^*_\mathbf{f} = [\mathbf{h}(\mathbf{f}, \mathbf{e}^*), \dots, \mathbf{h}(\mathbf{f}, \mathbf{e}^*)] \tag{3}$$

where all rows are identical and equal to the feature vector of the best hypothesis $\mathbf{e}^*$ within the n-best list. Then, $R_\mathbf{f}$ is defined as: $R_\mathbf{f} = H^*_\mathbf{f} - H_\mathbf{f}$

The key idea is to find a vector $\check{\boldsymbol{\lambda}}$ such that differences in scores are reflected as differences in the quality of the hypotheses. That is $R_\mathbf{f} \cdot \check{\boldsymbol{\lambda}} \propto \mathbf{l}_\mathbf{f}$ where $\mathbf{l}_\mathbf{f}$ is a column vector of N rows such that:

$$\mathbf{l}_\mathbf{f} = [l(\mathbf{e}_1), ..., l(\mathbf{e}_n), ..., l(\mathbf{e}_N)]', \forall \mathbf{e} \in \text{nbest}(\mathbf{f}) \tag{4}$$

The objective is to find $\check{\boldsymbol{\lambda}}^{s}$ such that:

$$\check{\boldsymbol{\lambda}}^{s} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} |\mathbf{R_f} \cdot \boldsymbol{\lambda} - \mathbf{l_f}| = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} ||\mathbf{R_f} \cdot \boldsymbol{\lambda} - \mathbf{l_f}||^2 \qquad (5)$$

where $\| \cdot \|^2$ is the Euclidean norm. Although 5 is equivalent (i.e., the $\check{\boldsymbol{\lambda}}^{s}$ that minimizes the first one also minimizes the second one), equation 5 allows for a direct implementation thanks to the ridge regression. $\check{\boldsymbol{\lambda}}^{s}$ can be computed as the solution to the overdetermined system $R_{\mathbf{f}} \cdot \check{\boldsymbol{\lambda}}^{s} = \mathbf{l_f}$, which is given by

$$\check{\boldsymbol{\lambda}}^{s} = (\mathbf{R_f'} \cdot \mathbf{R_f} + \beta \mathbf{I})^{-1} \cdot \mathbf{l_f} \qquad (6)$$

where a small β is used as a regularization term to stabilize $R_{\mathbf{x}}^{'} \cdot R_{\mathbf{x}}$ and to ensure that it is invertible.

$$H_{\mathbf{f}_s} \leftarrow [\mathbf{h}(\mathbf{f}_s, \mathbf{e}_{s,1}), \ldots, \mathbf{h}(\mathbf{f}_s, \mathbf{e}_{s,N})]', \forall \mathbf{e}_{s,i} \in \text{nbest}(\mathbf{f_s})$$

Algorithm 2 shows the pseudo-code for obtain $\check{\boldsymbol{\lambda}}^{s}$. In this work, we apply the original

```
for each of the sentences $\mathbf{f}_s$ in $A$ do
    $H_{\mathbf{f}_s} \leftarrow [\mathbf{h}(\mathbf{f}_s, \mathbf{e}_{s,1}), \ldots, \mathbf{h}(\mathbf{f}_s, \mathbf{e}_{s,N})]'$, $\forall \mathbf{e}_{s,i} \in \text{nbest}(\mathbf{f_s})$ ;
    $H^*_{\mathbf{f}_s} \leftarrow [\mathbf{h}(\mathbf{f}_s, \mathbf{e}^*_s), \ldots, \mathbf{h}(\mathbf{f}_s, \mathbf{e}^*_s)]'$ ;
    $R_{\mathbf{f}_s} \leftarrow H_{\mathbf{f}^*_s} - H_{\mathbf{f}_s}$ ;
    $\check{\boldsymbol{\lambda}}_s \leftarrow (\mathbf{R}'_{\mathbf{f}_s} \cdot \mathbf{R}_{\mathbf{f}_s} + \beta \mathbf{I})^{-1} \cdot \mathbf{l}_{\mathbf{f}_s}$ ;
    $\boldsymbol{\lambda}^s \leftarrow (1 - \alpha)\boldsymbol{\lambda}^{s-1} + \alpha \check{\boldsymbol{\lambda}}^s$
end
```

Algorithm 2: Pseudo-code for computing the vector $\boldsymbol{\lambda}^{s}$ as described in Section 3.1

DRR approach proposed by [5] to a batch scenario, so that the method proposed is effectively able to compete with state-of-the-art $\boldsymbol{\lambda}$ estimation approaches. In this case, DRR obtains an estimation of $\boldsymbol{\lambda}$ by previously adjusting the $\boldsymbol{\lambda}$ vector to each one of the sentences in a development corpus, i.e., the optimal $\boldsymbol{\lambda}$ is computed after performing a complete epoch on the development set.

4 Experiments

In this section, we describe the experimental framework employed. Then, we show a comparative by our strategy with two optimization methods (MERT and MIRA).

4.1 Experimental setup

All experiments were carried out using the SMT toolkit Moses [11]. The LM used was a 5-gram, with modified Kneser-Ney smoothing [12], built with the SRILM toolkit [13]. The phrase table was obtained with GIZA++ [3].

Translation quality was assessed by means of the BLEU [14]. BLEU measures n-gram precision with a penalty for sentences that are too short. However, it must be noted that BLEU is not well defined at the sentence level, since it implements a geometrical average of n-gram counts which is zero whenever there is no common 4-gram between reference and hypothesis, even if the reference has only three words. In our experiments, we used the *smoothed* BLEU approximation in [15] to calculate sentence-level BLEU. Note that the original work by [5] applied DRR to optimise TER scores [16], ignoring the problem of BLEU not being well defined at the sentence level. In this work we favour the use of BLEU because of its wider acceptance in the SMT community.

For each corpus, we trained baseline systems with which to compare the systems. This baseline was obtained by training the SMT system without tuning process obtaining the `baseline-emea` and `baseline-nc`. Since optimization methods require a random initialisation of $\boldsymbol{\lambda}$ that often lead to different local optima being reached, every point in each plot of this paper constitutes the average of 10 repetitions with 95% confidence intervals, with the purpose of providing robustness to the results.

DRR has different parameters that affect the experimental result, the most critical one being α. We conducted experiments with different α. Another meta-parameter is the regularization term β, which was fixed to 0.02 according to the work in [5]. The initial weight λ^0 used by our method was obtained using Moses random method.

4.2 Corpora

The experiments conducted in this paper were carried out on two different corpora (EMEA and NewComentary). The EMEA[4] corpus [17] contains documents from the European Medicines Agency. The News Commentary[5] (NC) corpus [18] is composed of translations of news articles. We focused on the English-French (Fr-En) and German-English (De-En) language pairs. Table 1 shows the main figures of the corpora.

Table 1: Main figures of the corpora. `Train` is the training set, `Dev` is development set, and `Test` is the test data. M denotes millions of elements and k thousands of elements, $|S|$ for number of sentences, $|W|$ for number of words and $|V|$ for vocabulary size.

Corpus	$\|S\|$	$\|W\|$	$\|V\|$	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
EMEA-`Train`	1.0M	12.1M	98.1k	NC-`Train`	120k	2.4M	27.6k
		14.1M	112k			2.8M	33.7k
EMEA-`Test`	1000	21.4k	1.8k	NC-`Test`	3000	56k	4.8k
		26.9k	1.9k			61k	5.0k
EMEA-`Dev`	501	9850	979	NC-`Dev`	1600	43k	3.9k
		11.6k	1.0k			47k	4.1k

4.3 Comparison between DRR and MERT and MIRA

We compare our method with MERT and MIRA. We study the effect of increasing the number of development samples made available to the system. Figure 1 and Figure 2 show the effect of adding sentences to development corpus and confidence intervals. These results show the quality translations in BLEU terms of each test corpus (EMEA-`Test` and NC-`Test`). Confidence intervals are displayed in different plots, instead of

[4] `www.opus.lingfil.uu.se/EMEA.php`

[5] `www.statmt.org/wmt13`

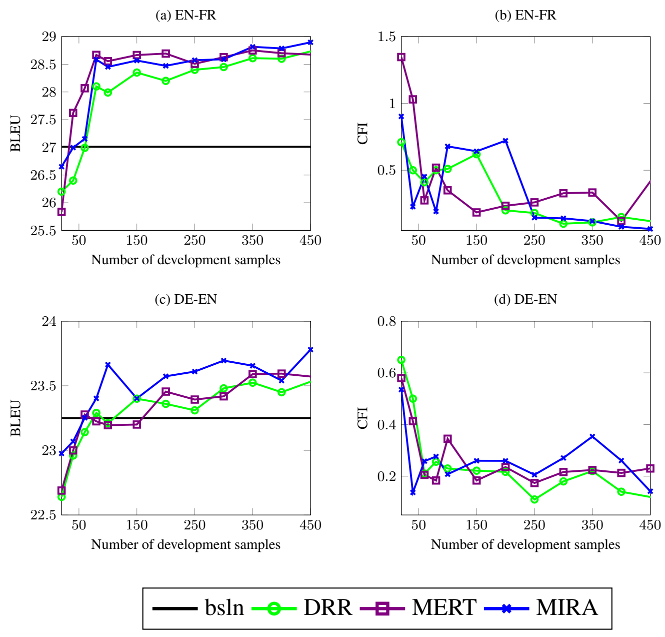

Fig. 1: Performance comparison across the corpus EMEA with different language pairs analysed. The two plots on the left display BLEU, while two plots on the right display the confidence intervals.

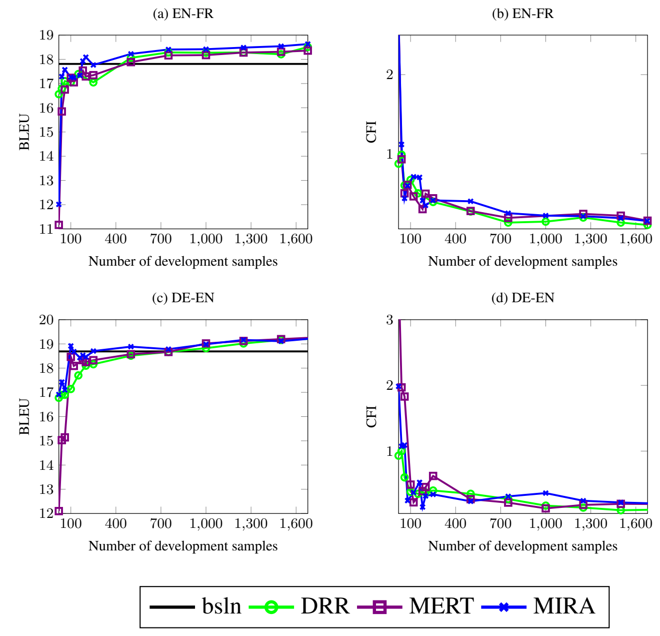

Fig. 2: Performance comparison across the corpus NC. (See Figure 1 for an explanation of the figures.)

using error bars, because otherwise the translation quality plots would present vertical lines across the complete plot, rendering it unreadable. We only show results for the best learning rate α for clarity and the best configuration obtain to MERT and MIRA. Figures 1 shows the principal result obtained using the EMEA corpus.

– Results obtained with our DRR method are similar than the ones obtained with MIRA and MERT. Behavior DRR results is increase when the amount of development corpus larger.
– Confidence interval sizes are shown in Figure 1b and 1d. Our DRR technique have a behaviour more stable that the MERT and MIRA that that shown more inestability.

Figures 2 shows the principal result obtained using the NC corpus.

– Increase the number of adaptation sample all the method obtain similar results, buy we can see MERT and MIRA are only to able to yield improvements when provided with at least 500 (MERT) and 200 (MIRA) development sample displaying a very chaotic behaviour until these point.
– Confidence interval sizes are shown in Figures 2b and 2d. MERT and MIRA yields large confidence intervals (as large as 3 BLEU points for less than 100 samples), turning a bit more stable from that point on, where the size of the confidence interval converges slowly to 0.5 BLEU point. In contrast, our DRR technique yields small confidence intervals, about 1 BLEU point in the worst case. This is worth emphasising, since estimating $\boldsymbol{\Lambda}$ by means of MERT or MIRA when very few development data is available may improve the final translation quality, but may also degrade it to a much larger extent. In contrast, our DRR technique shows stable and reliable improvements from the very beginning.

Table 2 shows the best results in terms of BLEU achieved by the three methods, i.e., DRR, MERT and MIRA. As shown, our method is able to yield competitive results in all scenarios considered. We understand that is important, since it proves the competitiveness of our proposal in this task, with respect to the other techniques state-of-the-art.

Table 2: Sumary of the best result obtained for each corpus and languege.

Corpus	Strategy	EN-FR BLEU	DE-EN BLEU
EMEA	`MERT`	28.7 ± 0.3	23.5 ± 0.2
	`MIRA`	28.9 ± 0.2	23.7 ± 0.1
	`DRR`	28.8 ± 0.2	23.6 ± 0.1
NC	`MERT`	18.4 ± 0.2	19.3 ± 0.2
	`MIRA`	18.6 ± 0.2	19.3 ± 0.1
	`DRR`	18.5 ± 0.1	19.2 ± 0.1

5 Conclusion and future work

We have proposed a simple technique for log-linear weight optimization an SMT system based in discriminative ridge regression method that is on par with the leading techniques, exhibits reliable behaviour and is remarkably easy to implement and use. We have demonstrated, via an empirical experiments, that our DRR method obtain comparable result than MERT and MIRA. In future work, we will carry out new experiments with large amounts of corpus, and languages diversity.

Acknowledgement. The research leading to these results has received funding from the Generalitat Valenciana under grant PROMETEOII/2014/030 and the FPI (2014) grant by Universitat Politècnica de València.

References

1. F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL*, pp. 160–167, 2003.
2. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.
3. F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational linguistics*, vol. 29, pp. 19–51, (2003).
4. P. Koehn, *Statistical Machine Translation*. Cambridge University Press, 2010.
5. P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta, "Online adaptation strategies for statistical machine translation in post-editing scenarios," *Pattern Recognition*, vol. 45, no. 9, pp. 3193–3203, 2012.
6. C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proc. of NAACL*, pp. 427–436, 2012.
7. G. Sanchis-Trilles and F. Casacuberta, "Log-linear weight optimisation via bayesian adaptation in statistical machine translation," in *Proc. of ACL*, pp. 1077–1085, 2010.
8. B. Marie and A. Max, "Multi-pass decoding with complex feature guidance for statistical machine translation," in *Proc. of ACL*, pp. 554–559, 2015.
9. M. Hopkins and J. May, "Tuning as ranking," in *Proc. of EMNLP*, pp. 1352–1362, 2011.
10. C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.
11. P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proc. of ACL*, pp. 177–180, 2007.
12. R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. of ICASSP*, pp. 181–184, 1995.
13. A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Proc. of ICSLP*, pp. 901–904, 2002.
14. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of ACL*, pp. 311–318, 2002.
15. B. Chen and C. Cherry, "A systematic comparison of smoothing techniques for sentence-level bleu," in *Proc. of WMT*, pp. 362–367, 2014.
16. M. Snover, B. J. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. of AMTA*, pp. 223–231, 2006.
17. J. Tiedemann, "News from opus-a collection of multilingual parallel corpora with tools and interfaces," in *Proc. of RANLP*, pp. 237–248, 2009.
18. J. Tiedemann, "Parallel data, tools and interfaces in opus," in *Proc. of LREC*, pp. 2214–2218, 2012.