GAWO: Genetic-based optimization algorithm for SMT

Ameur Douib, David Langlois, Kamel Smaili

▶ **To cite this version:**

Ameur Douib, David Langlois, Kamel Smaili. GAWO: Genetic-based optimization algorithm for SMT. ICNLSSP 2017 - International Conference on Natural Language, Signal and Speech Processing, ISGA, Dec 2017, Maroc, Morocco. hal-01660010

HAL Id: hal-01660010
https://inria.hal.science/hal-01660010

Submitted on 9 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GAWO: Genetic-based optimization algorithm for SMT

Ameur Douib[1], *David Langlois*[1], *Kamel Smaili*[1]

[1]University of Lorraine, LORIA, France

ameur.douib@inria.fr, david.langlois@loria.fr , kamel.smaili@loria.fr

Abstract

In this work, we propose GAWO, a new method for SMT parameters optimization based on the genetic algorithms. Like other existing methods, GAWO performs the optimization task through two nested loops, one for the translation and the other for the optimization. However, our proposition is especially designed to optimize the feature weights of the fitness function of GAMaT, a new genetic-based decoder for SMT. We tested GAWO to optimize GAMaT for French-English and Turkish-English translation tasks, and the results showed that we outperform the previous performance by +4.0 points according to the BLEU for French-English and by +2.2 points for Turkish-English.

Index Terms: Statistical Machine Translation, log-linear approach, optimization of feature weights, genetic algorithms

1. Introduction

Statistical machine translation (SMT) systems combine a set of features in order to evaluate the quality of a translation hypothesis at the decoding time. Each feature estimates the quality of the hypothesis in a particular aspect. The best translation in the output of the system is the one that maximizes this evaluation score. The feature values are mainly probabilities, i.e. language model probability, translation model probabilities, but other kinds of features are calculated and not interpreted as probabilities, i.e. phrase and word penalties.

In the SMT community, the log-linear approach [1, 2] is largely used to combine these features as follows:

$$Score(e) = \sum_{i=1}^{|\lambda|} \lambda_i \times \log(h_i(e, f)) \quad (1)$$

Where e and f are, respectively, a translation hypothesis and the source sentence, and h_i is the i^{th} feature function. To define the influence of each feature in the final score, a weight λ_i is associated to the feature h_i. The weight values have an effect on the scoring of the translation hypotheses at the decoding time, so, they have an effect on the choice of the best one in the output of the decoder. Therefore, within SMT systems the optimization of the weights of the features is a crucial step to insure a satisfactory translation quality.

To optimize these weights for a SMT system, we assume that we have a $development$ set consisting of source sentences and their reference translations. Then, using the decoder of the system, we translate the source sentences. The goal of the optimization is to find the set of weights λ that minimizes or maximizes the error function (Err) defined to estimate the difference between the output translations and the references. Ideally, to perform this setting, we apply a classical optimization algorithm by running the decoder many times and adjusting the set of weights, until a convergence, and as error function we use an evaluation machine translation metric. But, the run of the decoder several times is very expensive.

This is why, in practice, the weight optimization algorithms for SMT minimize the run of the decoder. They perform in two nested loops; in the outer loop, the decoder uses a set of weights to translate the $development$ set and produces the k-$best$ translations for each source sentence. In the inner loop, an optimization algorithm is applied to perform the weight tuning. In the inner loop, the main goal is to find the optimal weights that select the best translation for each source sentence from the k-$best$. The selected translations must minimize or maximize the error function. The optimal obtained weights are re-injected into the outer loop to run the decoder again. The process is stopped when the weights can no longer be improved, or no new translations can be produced.

MERT (Minimum Error-Rate Training) [3] is the default algorithm used to optimize the feature weights in the SMT community, and especially applied on MOSES [4], the reference translation system. As explained before the algorithm performs in two nested loops, where MOSES is used as a decoder and in the inner loop, a line-search optimization algorithm is applied to perform the tuning. Other algorithms have been proposed to optimize the weights of the log-linear approach. The main difference in all these algorithms lies in the inner loop, where different approaches are proposed to solve the problem of weight optimization. For instance, in [5], Hasler et al. used Margin Infused Relaxed Algorithm (MIRA) as an optimization algorithm. Hopkins and May proposed PRO [6], which works by learning a weight set that ranks translation hypotheses in the same order as the translation metric. More recently, Kocur and Bojar proposed to use a Particle Swarm Optimization algorithm (PSO) [7] to solve the problem in the inner loop.

However, the common point, and the weakness, of all these algorithms is the fact that they are specially designed to optimize the feature weights for the MOSES decoder. This makes their application difficult to another decoder.

In this paper, a Genetic Algorithm for weight optimization (GAWO) for SMT decoders is proposed. GAWO has the same principle as the others algorithms, with an outer loop for the translation and an inner loop for the optimization. However, our proposition is designed to optimize the parameters of GAMaT [8], a new genetic-based SMT decoder, which differs from MOSES (see Sections 2.1 and 2.2). In a previous work [8], GAMaT was tested and compared to MOSES, but by using a set of weights optimized by MERT for MOSES. In this work, GAWO will be used to optimize these weights for GAMaT to obtain an evaluation function more robust and better adapted to GAMaT which will improve the translation performance.

Genetic algorithms are known for their capacity of exploration and exploitation of the search space [9], particularly in the case of optimization problems [10]. So, a genetic algorithm is an interesting candidate to implement, in the inner loop, for the optimization of feature weights.

The paper is structured as follows. In Section 2, we present

the MOSES and GAMaT decoders. In Section 3 we present the tuning process for optimization of feature weights. In Section 4, we present GAWO the genetic-based algorithm for the optimization of feature weights. In Section 5, we present some experimental and comparative results. We conclude in Section 6.

2. SMT Decoder

In SMT systems the translation problem at the decoder level is considered as an optimization problem. The goal of the decoder is to find the best possible translation $\hat{e}$ that maximizes the translation evaluation score:

$$\hat{e} = \underset{e}{argmax}\Big[\sum_{i=1}^{|\lambda|} \lambda_i \times \log(h_i(e, f))\Big] \quad (2)$$

As presented in the introduction, the weights λ_i must be optimized to obtain a robust evaluation function.

2.1. MOSES

MOSES is a SMT decoder, which uses a Beam-search algorithm [4] in order to retrieve the best possible translation. Starting with an empty set, the solution building process consists in producing incrementally a set of complete solutions from partial ones provided by a translation table. Because the translation is built incrementally, it is then difficult to challenge a previous decision of translation that can eliminate a partial hypothesis, even if it could lead to a good final solution.

2.2. GAMaT

An alternative to MOSES is to start with a complete translation hypothesis and try to refine it to retrieve the best solution. With complete translation hypotheses, it is possible to revisit each part of the search space and modify it, if necessary.

GAMaT is a new decoder for SMT based on a genetic algorithm. It has the advantage that it can refine several complete solutions in an iterative process and produce acceptable solutions. In fact, a possible solution is encoded as a chromosome, where the chromosome encloses several pieces of information (the source sentence segmented into phrases, a translation hypothesis also segmented into phrases, and the alignment between source and target segments) [8]. Then, from an initial population of chromosomes, we produce new ones by applying crossover and mutation functions [8]. The crossover function takes two chromosomes from the population as parents, and crosses them to produce two new chromosomes considered as children of the parents. The produced chromosomes share the chromosomal information of the parents. On the other hand, the mutation functions are applied to diversify the existing population. A mutation function takes one chromosome and modifies it at the phrase level to produce new one. At the end of each iteration, we estimate the fitness (score) of chromosomes to select which of them will be kept, for next generations. This is called the selection process. The same process is repeated from one generation to another, until convergence. So, the final translation is the one that is encoded in the best chromosome from the final population.

To evaluate the chromosomes, the fitness is calculated using the log-linear approach to combine a set of feature values, as presented in the Equation 1. In this work, eight basic features [8, 4] are combined:

- Language model probability
- Direct and inverse translation model probabilities
- Direct and inverse lexical weighting translation model probabilities
- Word penalty
- Phrase penalty
- Reordering score

3. Tuning

The tuning of a SMT system consists in optimizing the weights λ of the evaluation function (Equation 1). To this end, let's assume a $development$ (Dev) set of source sentences $\{f_1, ..., f_n\}$ with their reference translations $\{r_1, ..., r_n\}$. The decoder produces a set of k translation hypotheses $\{e_i^1, ..., e_i^k\}$ for each source sentence f_i.

The optimal set of weights is the one which minimizes the sum of errors between the translations $\{\hat{e_1}, ..., \hat{e_n}\}$ and the references:

$$\hat{\lambda} = \underset{\lambda}{argmin}\Big[\sum_{i=1}^{n} Err(\hat{e}_i, r_i)\Big] \quad (3)$$

Where Err is the loss function to optimize at tuning, and $\hat{e_i}$ is the highest scored hypothesis from the set of k-translations of f_i using the set of weights λ. In other words, $\hat{e_i}$ is the best translation of f_i according to the weights λ.

The loss function estimates the quality of a set of translation hypotheses compared to the references. In the state-of-the-art, there are several choices [11] to define Err, ex: error function, soft-max loss, ranking loss. But the studies showed [11, 3] that the use of the translation evaluation metric BLEU (Bilingual Evaluation Understudy) [12] gives the best optimization performance. Therefore, in this work we use the BLEU metric to perform the weight optimization.

4. GAWO

As explained in the introduction, GAWO is a genetic algorithm for weight optimization. Similarly to the existing algorithms, GAWO works through two nested loops (see Figure 1): the outer loop, in which we use GAMaT as a decoder to translate the Dev set and the inner loop, in which the genetic algorithm is applied to optimize the weights. The process of these two loops is shown in the Figure 1 and detailed in what follows.

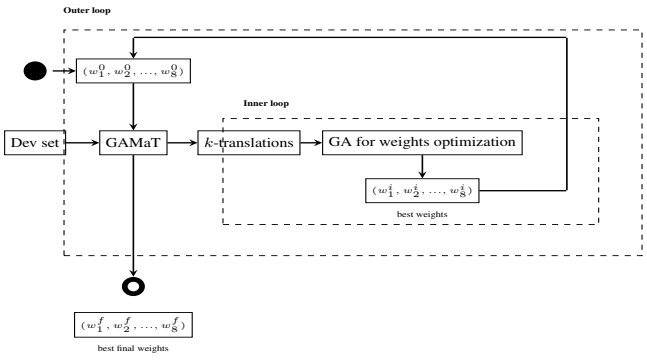

Figure 1: *The GAWO process through the outer and the inner loop.*

4.1. GAWO: outer loop

The Dev set is composed of n source sentences $\{f_1, ..., f_n\}$ with their reference translations $\{r_1, ..., r_n\}$. In the outer loop we use GAMaT to translate the source sentences, and produce for each one (f_i) a set of k-translations $E_i = \{e_i^1, ..., e_i^k\}$. Therefore, the result of this loop is sets of k-translations $\{E_1, ..., E_n\}$. These sets are injected into the inner loop for the optimization process. For the first iteration of the outer loop, we use a randomly generated set of weights. For the following iterations, the set of weights obtained by the inner loop is used to run GAMaT to produce novel sets of k-translations. The outer loop is stopped when the set of weights does not lead to new improvement.

4.2. GAWO: inner loop

In the inner loop, a classical implementation of the genetic algorithm is applied to find the best set of weights that selects a set of translations $\{\hat{e}_1, ..., \hat{e}_n\}$ maximizing the BLEU score (see Equation 3). The main idea is to start with a population of chromosomes (solutions) i.e. a population of vectors of feature weights, and iteratively improve the quality of the population by producing new chromosomes.

To ensure the good evolution of the population in a genetic algorithm, a fitness function must be defined according to the task to solve, in order to evaluate the chromosomes and apply the selection process at the end of each iteration. As we deal with the task of optimizing weights, the fitness function is the BLEU score (see Equation 3). The process continues iteratively, until it reaches the best possible weights for the current sets of k-translations. In the next sections, we explain with more details the genetic process used in GAWO.

4.2.1. Chromosome encoding

In the genetic algorithm, a possible solution is encoded in a chromosome. Therefore, in our case a chromosome represents a vector of eight elements ($c = \lambda$), where each position in c represents the weight value λ_i of a feature function h_i used in the evaluation function in GAMaT (see Section 2.2).

4.2.2. Population Initialization

The chromosomes (solutions) of the first population are randomly generated. Where, each feature weight takes its value in the range [-1, 1]. To this randomly generated population, we add the set of weights used in the outer loop to run GAMaT.

4.2.3. Crossover function

The crossover function is applied to couple the chromosomes of the existing population to enhance the quality of this population. The function takes randomly two chromosomes, c_a and c_b, from the population and selects a random position s in the chromosome, to cross them. The crossover function produces two new chromosomes c_c and c_d, where $c_c = \{c_a\}^{left(s)} \frown \{c_b\}^{right(s)}$ and $c_d = \{c_b\}^{left(s)} \frown \{c_a\}^{right(s)}$.

In practice the crossover function is applied to couple all the possible pairs of chromosomes in the population.

4.2.4. Mutation function

As presented previously, the crossover function couples the existing chromosomes in the population, which limits the search space to the values of the weights generated at initialization. Consequently, the algorithm has a high probability to converge towards a local optimum. This is why the mutation function is applied, in order to diversify the population. So, the mutation function selects a chromosome c_a from the population and modifies randomly one of its weight values to produce a new one c_b. For a better diversification, the mutation function is applied on all the chromosomes of the population.

4.3. Chromosome evaluation function

As presented before, the fitness function is the BLEU score. Therefore, to evaluate a chromosome c_a from the population, we process as follows. First, using the set of weights λ encoded in c_a, we recalculate the log-linear scores (see Equation 1) of every translation produced by GAMaT in the outer loop. After, for each source sentence f_i, we select from E_i the translation $\hat{e}_i$ that maximizes the log-linear score. The result of this step is a set of n translations ($\{\hat{e}_1^\lambda, ..., \hat{e}_n^\lambda\}$). Finally, we calculate the BLEU score between the selected translations and the references. The obtained BLEU score represents the fitness of the chromosome c_a.

In this way, the optimal set of weights is the one that is encoded in the chromosome maximizing the BLEU score.

5. Experiments & Results

5.1. Corpora

For the experiments, we use the translation data task of the workshop on Statistical Machine Translation. We take two pairs of languages to test GAWO in order to optimize the weights for GAMaT. The first pair is the French-English (FR-EN) a classical translation task. The second pair is the Turkish-English (TR-EN) which is a new task and poor in data. In addition, there is a stronger syntactic difference between Turkish and English. The training corpus of the language pairs FR-EN and TR-EN is composed of 1,3M and 280K parallel sentences respectively. Concerning the tuning and the test corpus, for both pairs of languages, both of them are composed of 1,000 parallel sentences. We use GIZA++ [13] to generate the translation model, and SRILM [14] to produce a $4\text{-}gram$ language model.

As presented in Section 4.3, the BLEU metric is used to perform the optimization. to evaluate the translation quality of the system on the test, we use BLEU, TER [15] and METEOR [16] metrics.

5.2. Results

In the Figures 2 and 3, we analyze the evolution of the translation quality of the Dev set throughout the optimization process. To this end we draw two curves, the first one (inner-loop) represents the evolution of the BLEU score through the inner loop, where each point represents the BLEU value of the best translations $\{\hat{e}_1, ..., \hat{e}_n\}$ selected from the sets of k-translations by using the best weights produced by the inner loop. In other words, each point represents the fitness score of the best solution produced by GAWO in the inner loop. The second curve (outer-loop) represents the evolution of the BLEU value of the k-translations produced by GAMaT in the outer loop. This last curve allows us to analyze the evolution of the translation population quality produced by GAMaT.

After some experiments and like what is made in MERT [3] we fixed the number of the translations for each source sentence to 100 (k=100), these translations are injected in GAWO to perform the optimization process. To have a large variety of translations for each source, we add to the translations produced

by GAMaT at the iteration i those of the iteration i-1.

Knowing that the genetic algorithms that we use have a random behaviour, we analyzed the robustness and the stability of our algorithm to verify if GAWO achieves the convergence or not in any case. For this we did multiple runs (5) starting with the same weight values at each run (see Figures 2 and 3).

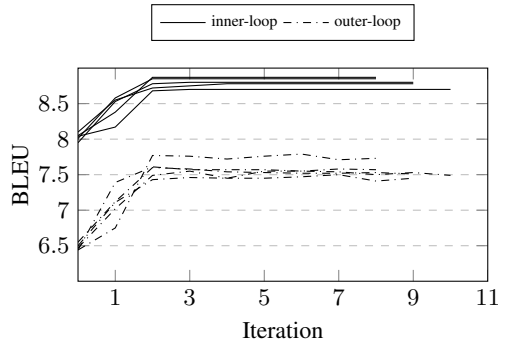

Figure 2: *The evolution of the BLEU on the Dev set for TR-EN. The 1-best output from the inner loop and the 100-best output from the outer loop (GAMaT).*

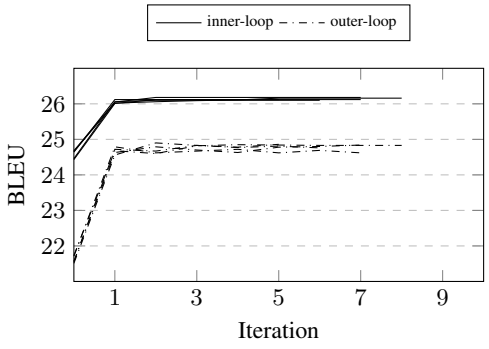

Figure 3: *The evolution of the BLEU on the Dev set for FR-EN. The 1-best output from the inner loop and the 100-best output from the outer loop (GAMaT).*

The first remark that can be made through these curves, is that the process of the optimization converges for all the runs and this for the both language pairs. Thanks to this convergence, we can conclude that GAWO allows GAMaT to produce better translations, by generating weight values more adapted for GAMaT.

We can also see that on average, three iterations are sufficient for the process to reach the maximum scores. This can be explained by the fact that we add the translations produced in the previous iteration for the current one. Therefore, after three iterations, GAWO deals with a huge number of translations for each source sentence (k=300). This allows to make better decisions and find the best weight values that helps to select the best translation for each source from 300 possible translations.

In terms of performance in the inner loop (inner-loop curves in Figures 2 and 3), for the TR-EN pair, the process achieves an average BLEU score of 8.8 for the five runs. for the FR-EN pair, it achieves an average BLEU score of 26.11. On the other hand, for the performance of the outer loop (outer-loop curves), for the TR-EN pair, GAMaT produces a population of translations with an average BLEU quality around 7.6 and an average BLEU quality around 24.83 for the FR-TR pair.

In Table 1, we present the translation performance on the $Test$ set, according to the three evaluation metrics previously cited. To estimate the improvement provided by the optimization of the weights by GAWO on the translation performance of GAMaT, we run GAMaT with the feature weights optimized by MERT for MOSES ($GAMaT\text{-}MERT$), and we run it also with the weights obtained from the different runs of GAWO. We show in Table 1 the average score of the five runs ($GAMaT\text{-}GAWO$). The confidence intervals are calculated with 95% of confidence.

Table 1: *Translation performance on the test set according BLEU, TER and METEOR.*

Language	Decoder	BLEU ↑	TER ↓	METEOR ↑
TR-EN	MOSES+MERT	10.29	83.03	20.2
	GAMaT-MERT	6.46	82.05	18.39
	GAMaT-GAWO	$8,73$ (± 0.1)	$80,27$ (± 0.41)	$18,93$ (± 0.04)
FR-EN	MOSES+MERT	31.29	52.14	29.97
	GAMaT-MERT	24.84	57.18	28.55
	GAMaT-GAWO	28.91 (± 0.14)	$53,68$ (± 0.07)	$28,87$ (± 0.07)

The obtained results show that the optimization of the weights using GAWO improve considerably the translation performance of GAMaT. Indeed, according to the BLEU score, the metric used at the tuning time, we outperform $GAMaT - MERT$ by more than 2.2 points for the TR-EN pair and more than 4.0 points for the FR-EN pair. But we do not exceed those of the reference system decoder MOSES. The improvement is visible also according to the TER, where we outperform $GAMaT - MERT$ by more than 1.7 point and those of MOSES by 2.76 points for the TR-EN pair. For the FR-EN we outperform also $GAMaT - MERT$ by more than 3.5 points. The improvement is less visible according to the METEOR, because this metric is less sensitive than BLEU, when they compare the translations with the references.

On the other hand, the small intervals of the confidence intervals prove that GAWO is stable and allows to GAMaT to achieve the same translation performance each time we run it.

6. Conclusions

We presented GAWO, a new method for SMT parameter optimization based on the genetic algorithms. GAWO was tested to optimize the feature weights of the fitness function of GAMaT for two different pairs of languages. The obtained results of the different experiments demonstrated the feasibility of our proposition, where the algorithm allows to converge towards an optimum set of weights. Moreover, the translation performance, according to three evaluation metrics, showed that the optimization of the weights allows GAMaT to outperform the previous configuration.

For future work, we will make more experiments by testing other metrics to perform the optimization and use GAWO to optimize the feature weights for MOSES, and compare our proposition with the different optimization algorithms.

7. References

[1] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *Annual Conference on Artificial Intelligence*. Springer, 2002, pp. 18–32.

[2] P. Koehn, F. J. Och, and D. Marcu, "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*. Association for Computational Linguistics, 2003, pp. 48–54.

[3] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 160–167.

[4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[5] E. Hasler, B. Haddow, and P. Koehn, "Margin infused relaxed algorithm for moses," *The Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 69–78, 2011.

[6] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1352–1362.

[7] V. Kocur and O. Bojar, "Particle swarm optimization submission for wmt16 tuning task," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 518–524. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2344

[8] A. Douib, D. Langlois, and K. Smaili, "Genetic-based decoder for statistical machine translation," in *Springer LNCS series, Lecture Notes in Computer Science*, Dec. 2016. [Online]. Available: https://hal.inria.fr/hal-01336546

[9] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.

[10] S. Binitha, S. S. Sathya *et al.*, "A survey of bio inspired optimization algorithms."

[11] G. Neubig and T. Watanabe, "Optimization for statistical machine translation : survey," *Computational Linguistics*, pp. 1–54, 2016.

[12] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[13] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[14] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: Update and outlook," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, vol. 5, 2011.

[15] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of Association for Machine Translation in the Americas*, vol. 200, no. 6, 2006.

[16] S. Banerjee and A. Lavie, "METEOR: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, vol. 29, 2005, pp. 65–72.