A Translation Evaluation Function based on Neural Network

Ameur Douib, David Langlois, Kamel Smaili

SCHEDAE INFORMATICAE
XXXXXXXXX VOLUME nn Year

A Translation Evaluation Function based on Neural Network

Ameur Douib
David Langlois
Kamel Smaïli
ameur.douib@inria.fr , david.langlois@loria.fr ,
kamel.smaili@loria.fr

Abstract. In this paper, we study the feasibility of using a neural network to learn a fitness function for a machine translation based on a genetic algorithm termed GAMaT. The neural network is learned on features extracted from pairs of source sentences and translations. The fitness function is trained in order to estimate the BLEU of a translation as precisely as possible. The estimator has been trained on a corpus of more than 1.3 million data. The performance is very promising: the difference between the real BLEU and the one given by the estimator is equal to 0.12 in terms of Mean Absolute Error.

Keywords: Statistical Machine Translation, Genetic algorithm, Quality estimation, Neural network

1. Introduction

Nowadays, a lot of Statistical Machine Translation (SMT) systems use a Beam-search algorithm [5] in order to retrieve the best possible translation by taking into account different scores provided by several models: language, translation, distortion, etc. 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 (TT). Because the translation is built incrementally, it is then difficult to challenge a previous decision of translation, which can eliminate a partial hypothesis, even if it could propose a good final solution.
An alternative to this algorithm is to start with a complete translation hypothesis and try to refine it in order to retrieve the best solution. With complete translation

hypotheses, it is possible to revisit each part of the research space and modify it, if necessary.
GAMaT [4] is a new decoder for SMT based on a genetic algorithm. It has the advantage to 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 information (the source sentence segmented into phrases, a translation hypothesis also segmented into phrases, and alignment between source and target segments). Then, from a population of chromosomes, we estimate their fitness (score) in order to keep them, or not, for next generations. To do so, the fitness is a combination of several scores measuring how the different segments of a chromosome are coherent with each others. Nine scores corresponding to nine features are combined to score translations [4]. A weight proportional to the impact of the feature on the evaluation function is assigned to each feature. This combination has been held by a log-linear approach. In GAMaT, the weights corresponding to the nine scores are provided by Moses [5]. According to the BLEU [10] metric results, the translation performance is good, but not better than Moses. To get away from Moses and to propose a relevant solution for GAMaT, we propose to learn the function of the chromosome evaluation by using a neural network (NN) which predicts a BLEU value of the translation represented in a chromosome. In other words, we would like the fitness function to be correlated to BLEU. The NN is learned on the nine features mentioned before. We opted for BLEU metric because it is commonly used in the MT community to evaluate translations. This kind of learning algorithm is used in the Quality Estimation (QE) community [3], in order to estimate the translation quality without access to the reference translation.
The article is structured as follows. In Section 2 we describe the chromosome features. In Section 3 we present the NN used to learn the fitness function. Then, in Section 4 we describe how we generate the dataset for the NN. We give the results of several configurations in Section 5. Finally, we conclude and give some analyses and perspectives.

2. Related works

In this paper, we propose a new translation evaluation function for a phrase-based SMT decoders, and which is applied for our genetic-based decoder GAMaT [4]. This function is learned, using a neural network, on nine chromosome features and correlated with the BLEU value of the translation enclosed in the chromosome.
Therefore, our work can be classed at the intersection of two research disciplines; The first one is the optimization of decoder parameters for machine translation [7, 9]. Where, for the majority of decoders, the objective function combines log-linearly a set of translation features to evaluate the translation hypotheses (see equation (1)). Optimising the weights of this function allows a better translation accuracy. In the machine translation community to optimise these weights, the proposed algorithms are largely based on a grid search algorithm [9]. Where the goal is to find the best

set of weights which minimise a loss function adapted for the translation process [7]. The second domain is the Quality Estimation (QE). The main goal in this area is to estimate the translation quality without access to the reference translation [3]. To this end, in QE community, machine learning algorithms are trained on features extracted from pairs of source sentences and their translations, and they are correlated with an evaluation metric, which can be a metric with binary values (good/bad), or a metric with continuous values (BLEU, TER, ...etc.).
In this work, we use a neural network as in QE community, to combine optimally features used in the log-linear approach. The learned function estimate the quality of the translation by predicts its BLEU value without access to the reference translation, which is the case at decoding time.

3. The features of the chromosomes

In order to evaluate the relevance of a chromosome, we need to evaluate it by combining its different features, such as what was done in [4]. The features are log-linearly combined as follows:

$$Score(c) = \sum_i \lambda_i \times \log(h_i(e, f)) \quad (1)$$

Where f is the source sentence and e the translation represented in the chromosome c. λ_i is the weight of h_i determined by Moses and h_i is the score related to the i^{th} feature. The value of each weight defines the influence of the corresponding feature in the final score. Nine features related to the construction of a chromosome have been used and are described in the following.

- F_1: a language model feature, which estimates how the translation e is linguistically correct in the target language. In practice, F_1 is estimated by the likelihood $P(e)$ of a translation yielded by the classical n-gram. In our experiments, n is set to 3.
- F_2: the second feature concerns the translation probability. Given a chromosome c, F_2 is based on the alignment presented in c. This alignment links each source phrase to a target phrase. Then, F_2 is the product of translation scores (from the source towards the target languages, given by the translation table) between linked source and target phrases. This feature is called the direct translation probability.
- F_3: this feature concerns the inverse translation probability, which is equivalent to the previous one, except that the translation scores are from the target towards the source languages.
- F_4: this feature estimates the quality of a pair of segments at word level [6]. It is defined as the product of lexical probabilities inside one segment and over all the segments of the source sentence. This feature is called a direct lexical probability.

- F_5: symmetrically to F_4, an inverse lexical probability is estimated.
- F_6: this one concerns the length of the target sentence in the chromosome to produce. In fact, the translation should not be too much longer than the source sentence. This feature is set with the difference between the length of the source and the translation in terms of words.
- F_7: to reinforce the previous feature, a length model is trained [4] that assigns a probability to a pair of sentences depending on their lengths. In other words, this feature is estimated as the probability that a source sentence with a length l_s, will produce a translation of length l_t.
- F_8: longer sequences are linguistically more informative than smaller ones. Therefore, a chromosome with a few number of phrases should give a better translation. This feature is called the phrase penalty and is estimated as an exponential function of the number of phrases: e^k, where k is the number of phrases in the chromosome.
- F_9: it is the cost of permutations in the translation at the phrase level, when the target phrases are picked out of order [5].

To estimate some of the previous features, we need a translation table, which contains pairs of source and translation phrases with their associated probabilities.

4. A machine learning algorithm for prediction

As presented in the introduction, the goal of this work is to propose a new chromosome fitness function which combines optimally the features previously presented, and which must be correlated with BLEU. To do so, we decided to train a supervised neural network which takes as input the nine features of a chromosome, and as label in output the BLEU value of the corresponding translation (Figure 1). The learned fitness function is supposed to predict the BLEU of the translation of a chromosome.

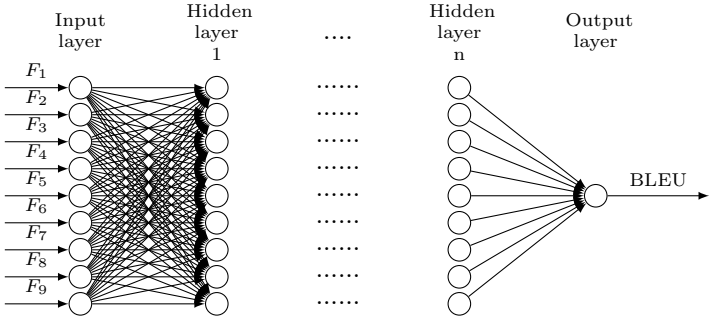


Figure 1.: The neural network architecture

We experimented different configurations of the neural network by varying the number of the layers and the number of neurons. We used the Sigmoid function for the neuronal activation. The experiments have been conducted by using Keras, a Deep Learning library [1].

5. The dataset for Neural Network

To learn the neural network, a training corpus is necessary. In the following, it will be composed by an important number of pairs $< v_c, \alpha >$. Where v_c is a vector of 9 features which characterize a chromosome c and α represents the BLEU value of the translation hypothesis represented in the chromosome c. To this end, we used the French-English parallel corpus of the 9^{th} task workshop on SMT [2]. We used this corpus to produce the translation table TT which is necessary to build the chromosomes and also to calculate some features of chromosomes.
The corpus contains pairs of French sentences and their English reference translations. We split it into two sets: the former, containing 1,323,382 pairs, is used to build the TT handled by GIZA++ [8] and the language model; while the latter set (C_{NN}), containing 165,422 pairs, is used to produce chromosomes.
For each source sentence in C_{NN} we built a set of chromosomes, e.g. a set of hypotheses with the segmentation of source and target sentences, and the alignment between source and target segments. We have to produce chromosomes representing perfect, good and bad translations in order to diversify the dataset because the fitness must predict correctly the translation quality of chromosomes, whatever this quality is. To produce chromosomes, we used several functions of GAMaT [4] which take a source sentence and produce one or more chromosomes.

The methods used to build chromosomes are presented in the following:

- From a source sentence, the builder of chromosomes proposes a segmentation based on the longest phrase within it. This longest phrase can be found anywhere in the source sentence. This phrase is picked up from the TT. The segmentation process is iteratively repeated until the whole sentence is segmented. Then, each source phrase is translated in the target language by choosing the most likely translation from TT. Source sentence and hypothesis have same phrase ordering. This produces one chromosome.
- The process is the same as the previous method, but the source sentence is segmented from left to right. This produces one chromosome.
- The process is the same as the previous method, but the source sentence is segmented from right to left. This produces one chromosome.
- Here, the source sentence is randomly segmented, where all the produced segments must exist in the TT. This random segmentation is done several times, in order to increase the number of chromosomes. This allows to increase the size of the training corpus.

- In this one, the builder performs similarly such as in the previous point, except that instead of selecting the best translation for a source phrase, it chooses randomly from TT one from the possible translations of this phrase.
- The producer proposes chromosomes such as in the previous point, except that the alignment between a source phrase and a target segmentation is not monotone: target phrases are mixed. This allows to build several bad chromosomes.
- For each source sentence, the producer of chromosomes keeps the best solution proposed by GAMaT. This allows to produce for each source sentence one chromosome with a good quality translation.
- Similarly to the previous one, the producer of chromosomes keeps the best solution proposed by Moses for the source sentence.

Using these methods, we produced 1,5 million of chromosomes, from 165,422 source sentences. For each of them, we produced a set v_c of 9 features, and we calculated α, which is the BLEU value in this work. The BLEU estimates the similarity between the hypothesis and the reference translation. This produced a training corpus for the NN composed of a set of pairs $< v_c, \alpha >$.

6. Experiments and results

In this section, we describe our experiments. We describe the different training and test corpus we used, and how we measured the performance of the neural network.

6.1. Evaluation metric

To evaluate the precision of the prediction function, the Mean Average Error (MAE) criterion [11] is calculated on the test set. The error is the difference between the real BLEU and the predicted value:

$$MAE_{BLEU} = \frac{1}{n}\sum_{i=1}^{n} |BLEU_r(c_i) - BLEU_p(c_i)| \quad (2)$$

Where $BLEU_r(c_i)$ is the real BLEU value of the translation in the chromosome c_i, and $BLEU_p(c_i)$ is the predicted BLEU value for the same translation. n represents the size of the test set.

6.2. Training and test corpus

In order to build the training and test corpus, we followed the process we described in Section 5. We segmented this corpus into a training and a test part respectively 90% and 10%. We show in Figure 2 the distribution of this corpus according to the BLEU value. This figure shows the proportion of chromosomes for five BLEU intervals: [0,0.25[(very bad translation according to reference), [0.25,0.5[, [0.5,0.75[, [0.75,1[and BLEU=1 (perfect translation). The bar *Unbalanced training set* corresponds to the proportions of the different qualities of translations presented above. The bar referenced as *Test set* corresponds to the proportions of the test corpus for which we keep the same distribution as in the training corpus. This corpus is not balanced, this could be a drawback, because the NN should estimate the translation quality with the same precision, whatever the correctness of a chromosome is. That is why, we trained also, the NN with a balanced training corpus (the bar *Balanced training set*). In GAMaT, at the translation time, the distribution presented above is not necessarily respected because the set of chromosomes (population) evolves at each iteration. Therefore, we achieve also experiments with another test corpus (bar *GAMaT test set* in Figure 2) built from a translation corpus used in [4]. This test corpus is representative of chromosomes obtained at end of the real translation process. This test corpus contains 50,000 chromosomes built from 1,000 source sentences. It contains very few perfect translations, and relatively more very bad translations compared to *Test set*.

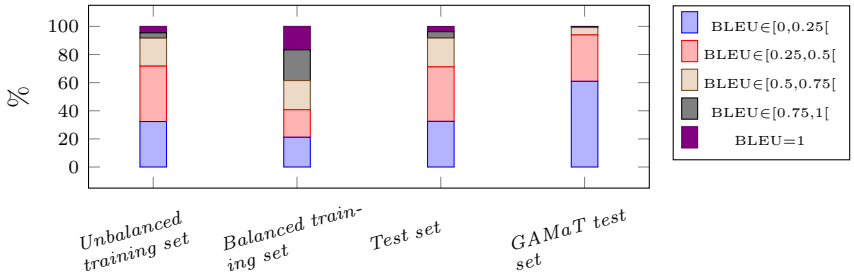


Figure 2.: Distribution of data in the training sets and test sets.

In the following, we present the performance of the estimator on *Test set* and *GAMaT test set*, trained with *Unbalanced training set* and *Balanced training set*.

6.3. Impact of the neural network architecture

In this section, we study the performance of the NN according to the number of layers (1 to 7) and neurons (8, 16, 32 and 64) in each layer. Figure 3 shows the MAE performance on *Test set* according to the number of layers, and to the number of neurons. In the left curve, for each number of layers the plot is the average performance according to the number of neurons. In the right curve, for each number

of neurons the plot is the average performance according to the number of layers. The figure shows that the best results are achieved for 4 layers and 32 neurons by layer.

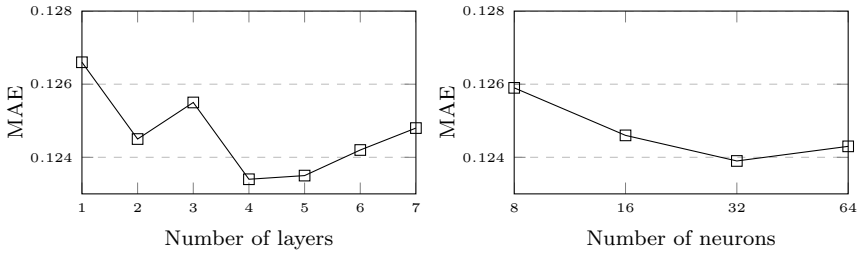


Figure 3.: The influence of the number of layers and neurons in the MAE score.

As these values are only averages, we give in Figure 4 the performance for each couple (number of layers, number of neurons by layer). This figure shows that the performance is better when the number of neurons grows with the number of layers. In the following experiments, we keep the best configuration: 4 layers, and 32 neurons by layer.

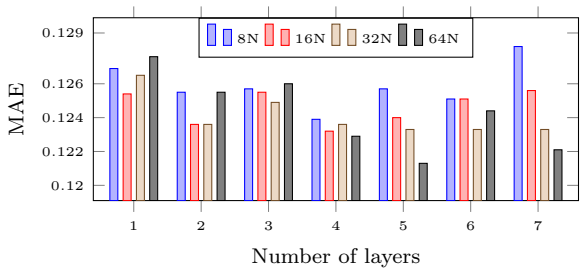
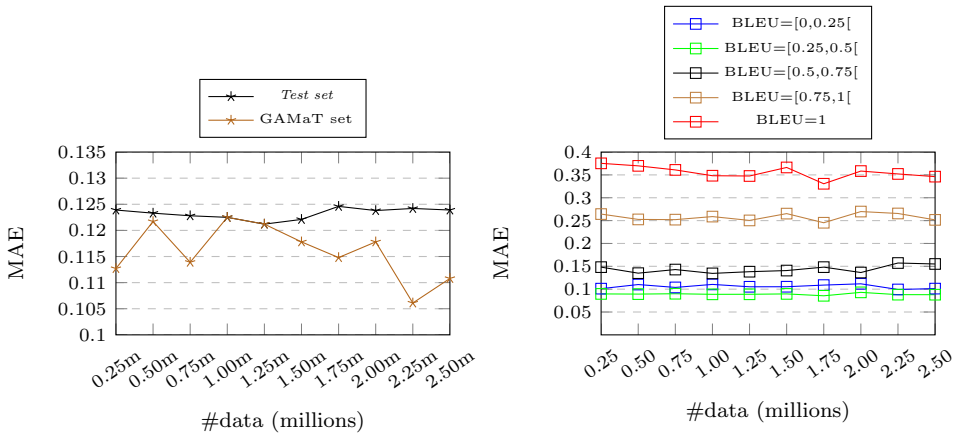


Figure 4.: Details of the influence of the number of layers and neurons in the MAE score.

6.4. Impact of the size and quality of the training corpus

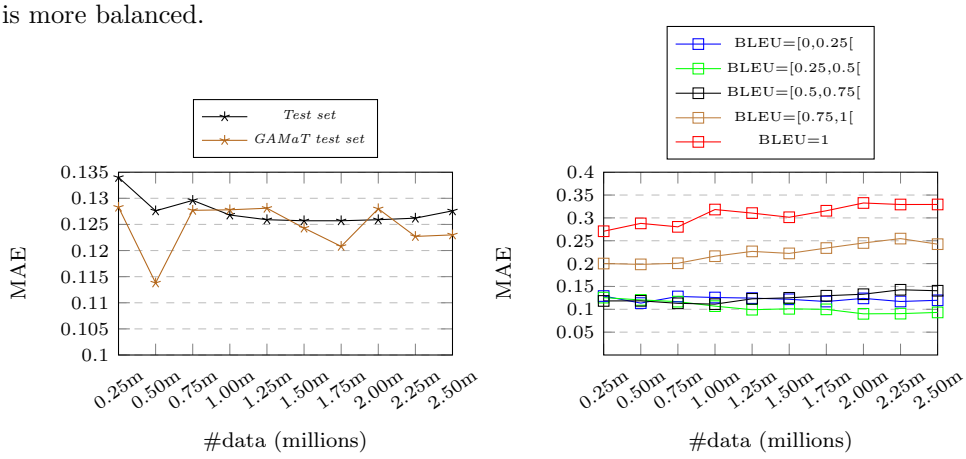
In machine learning, the size of the training data is crucial, and the test data should not be very different from the training data. To respect this constraint, we increased the training set, from 225,000 to 2,250,000 with and without paying attention to the proportion of the quality of the set of chromosomes (Figures 5 and 6). Figure 5-(a) shows the performance on *Test set* and *GAMaT test set* trained with several sizes of the *Unbalanced training set*.



(a) (b)

Figure 5.: The influence of the number of data in *Unbalanced training set*.

The performance on the corpus *Test set* is not affected by the size of the training corpus according to the results of Figure 5. On the contrary, the curve for *GAMaT test set* is more chaotic but the performance of the estimator tends to be better, as the size of the training increases. Figure 5-(b) shows that the estimator produces bad results for chromosomes for which the BLEU is greater than 0.75 and estimates correctly the others. This is probably due to the fact that bad chromosomes are more numerous than the good ones in *Test set*.
In Figure 6-(a), we did the same experiments, but with *Balanced training set*. The performance with good chromosomes is henceforth better, since the training corpus is more balanced.



(a) (b)

Figure 6.: The influence of the number of data in *Balanced training set*.

As in previous section, we give more details about MAE performance for test subsets according to translation quality in Figure 6-(b). This figure shows that balancing the training corpus allows to improve the prediction performance for good quality translations, but this is not sufficient to improve the overall performance because these good translations are not numerous in the *Test set*.

6.5. Results and Discussion

The results in the different campaigns of Quality Estimation are very close to each other [3]. Our results obey to this rule. Two main results emerge from this study. For a training corpus not selected smartly, in other words for the *Unbalanced training set*, the estimator needs 2.25 million of chromosomes to reach the best result 0.1061 (Table 1) on *GAMaT test set*. While for the *Test set*, we need only 1.25 million of chromosomes for a MAE of 0.1212. In spite of these low scores (lower is better for MAE), the estimator evaluates badly the best hypotheses of translation, for which the BLEU is greater than 0.75 (see Figure 5-(b)). When we use *Balanced training set*, we need only 0.5 million of chromosomes to reach the best performance for *GAMaT test set* with a MAE of 0.1138. The best result for the *Test set* is obtained for 1.5 million of chromosomes which leads to a MAE of 0.1257. With this *Balanced training set*, the best hypotheses are better evaluated than with the first training set (see Figure 6-(b)), but we lose in the quality estimation of the bad hypotheses, which are more numerous in the real translation process. This explains the fact that the performance of *Unbalanced training set* is better than those of *Balanced training set*.

		Test set	
Training set	**# of data**	**Test**	**GAMaT**
Unbalanced	1.25m	**0.1212**	0.1212
	2.25m	0.1242	**0.1061**
Balanced	1.50m	**0.1257**	0.1243
	0.50m	0.1276	**0.1138**

Table 1.: Best MAE scores for *Unbalanced training set* and *Balanced training set*.

To study the behavior of the proposed evaluation function in a real translation process, we used it as a chromosome evaluation function in our SMT decoder (GAMaT) in order to translate a set of 1.000 sources sentences. In Table 2 we present some translation results according to BLEU metric. Where for each training set we take the best function, learned by the neural network, according to the MAE score and we used it as a chromosome evaluation function in GAMaT.
The best translation performance are obtained when we train the neural network on 1.250.000 data with 5 hidden layers and 64 neurons in each layer. This configuration allowed us to achieve 21.05 in BLEU.
Through these results, we notice the fact that the neural network configuration, in terms of number of hidden layers and number of neurons, has the same influence on the quality of the learned function (Figure 5-(a)) as on the translation accuracy, when we use this function as a chromosome evaluation function in GAMaT. The function which has the best MAE score (Table 1), used as fitness function, it leads GAMaT to produce the best output translations.

Size of training set (Millions)	# of hidden layers	# of neurones	BLEU
0.25	5	32	**17.91**
0.50	7	20	**18.34**
0.75	6	64	**19.80**
1.00	5	64	**20.15**
1.25	5	64	**21.05**
1.50	6	64	**18.56**
1.75	3	64	**16.27**
2.00	4	64	**17.94**
2.25	5	64	**17.98**

Table 2.: Translation performance according to BLEU using the proposed function.

7. Conclusion and perspectives

In this paper, we investigated the use of a new function to evaluate chromosomes in GAMaT. The function is an estimator of BLEU for a chromosome. This estimator has been trained by a neural network which is learned on nine features extracted from a set of chromosomes built by GAMaT. The experiments show that the amount of the training data, and their distribution in terms of translation quality, impact the precision of the function. The prediction function achieved convincing results. In fact, the MAE calculated on a test set of 100,000 chromosomes is around 0.12. Also, the use of this function in GAMaT such as a fitness function achieves an encouraging translation performance. The presented method uses BLEU for evaluating a translation, but could be extended to other measures which might be combined in order to get a more robust fitness function. To improve this prediction function, sampling techniques should be used to select more representative data for the training.

References

[1] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, G. Desjardins, D. Warde-Farley, I. Goodfellow, A. Bergeron, et al. Theano: Deep learning on gpus with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain*. Citeseer, 2011.

[2] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics Baltimore, MD, USA, 2014.

[3] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, and P. Pecina, editors. *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal, September 2015.

[4] A. Douib, D. Langlois, and K. Smaïli. Genetic-based decoder for statistical machine translation. In *17th International Conference on Intelligent Text Processing and Computational Linguistics*. Springer LNCS series, 2016.

[5] 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*, pages 177–180. Association for Computational Linguistics, 2007.

[6] 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*, pages 48–54. Association for Computational Linguistics, 2003.

[7] E Moradi, SMT Fatemi Ghomi, and M Zandieh. Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem. *Expert systems with applications*, 38(6):7169–7178, 2011.

[8] F.-J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[9] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[10] 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*, pages 311–318. Association for Computational Linguistics, 2002.

[11] C.-J. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.