# Ensemble Statistical and Heuristic Models for Unsupervised Word Alignment

Mahsa Mohaghegh
Department of Computing
Unitec
Auckland, New Zealand
mmohaghegh@unitec.ac.nz

Hossein Sarrafzadeh
Department of Computing
Unitec
Auckland, New Zealand
hsarrafzadeh@unitec.ac.nz

Mehdi Mohammadi
Department of Computer Science
Western Michigan University
MI, USA
mehdi.mohammadi@wmich.edu

***Abstract*—Statistical word alignment models need large amounts of training data while they are weak in small-sized corpora. This paper proposes a new approach of an unsupervised hybrid word alignment technique using an ensemble learning method. This algorithm uses three base alignment models in several rounds to generate alignments. The ensemble algorithm uses a weighed scheme for resampling training data and a voting score to consider aggregated alignments. The underlying alignment algorithms used in this study include IBM Model 1, 2 and a heuristic method based on Dice measurement. Our experimental results show that by this approach, the alignment error rate could be improved by at least 15% for the base alignment models.**

***Keywords—statistical word alignment; ensemble learning; heuristic word alignment***

## I. INTRODUCTION

As the main application of Natural Language Processing (NLP), Machine Translation (MT) is becoming a necessary tool in today's rapid and voluminous stream of digital content. This need could be better addressed by increasing cross-regional communication as well as information exchange. For example, many TV channels broadcast with closed caption to different nations who have different languages. Another example is some communities like the European Union require documents to be translated in several languages simultaneously.

Statistical Machine Translation (SMT) is the dominant approach for machine translation systems in recent years, and is attracting more attention from researchers due to its improvement and development. However, one prominent problem in this field is word alignment of bilingual training data. Word alignment can simply be defined as mapping source language words to their corresponding translation words in the target language. In a professional view, a word alignment is an applicable hidden parameter in Statistical Machine Translation [1]. This problem would be more challenging when the underlying resources for the training models are limited.

Based on Och and Ney [1], a general definition of alignment between two word strings of source and target languages can be represented by a subset of the Cartesian product of the word positions in both word strings. However, because of the difficulty in implementing such general models, most alignment models are restricted in some way. One typical approach is One-to-One alignment [2] in a sentence pair ($F= f_1 \ldots f_I$, $E= e_1 \ldots e_J$) in which $I$ and $J$ are the length of the source and target sentences in terms of words, respectively. The alignment $A$ would be represented as a subset of $\{1,2, \ldots, I\}\times\{1,2, \ldots,J\}$. A source word in the position $i$ is mapped to a word of target language in position $j$, if $(i, j) \in A$. Mappings in this model may contain assignment to an empty string in the target language as well.

The statistical alignment models are the basis of statistical translation models and were initially word-based. IBM Models 1-5 [3], HMM [4] and Model 6 [1] are some remarkable instances of this category. In the scope of bilingual term extraction and dictionary construction, sequence-based models (IBM models 1, 2 and HMM) are more attractive compared to fertility-based models (like Model 3 and thereafter), since sequence-based models are simple and fast and their implementation is not as complex as fertility-based models [5].

An important factor of good quality word alignment is a huge amount of bilingual sentences. However this resource is not typically available for any language pair. One promising approach that has proven to yield reasonable results with both limited data and large data sets is ensemble learning. However, it has been proven that weak learners performing only somewhat better than random can be combined to create a stronger ensemble learner [6]. In this approach several learners, known as weak learners, work over the same training data and their results are aggregated to produce the final output. Ensemble learning has been used in word alignment [7-11] as well as SMT systems [12-13][15-17].

Recent research tends to combine several machine translation systems with different levels of strength to improve translation quality. The idea is that stronger systems are able to cover the deficiency of weaker systems [12]. In other words, errors can be addressed by the correct prediction of other systems. This can be realized by generating translation using a voting algorithm in a set of relatively close translation

outputs. This idea could be applied to word alignment problems as well.

In this paper, we propose a new ensemble approach to achieve some improvements in word alignment problems for low-resource languages. Our approach is based on employing a combination of three different word aligners, two of them based on statistical models and one based on a heuristic model. Then we resample training data for these algorithms to have several weak word alignment learners. Then the results of these weak learners are combined together to produce the final alignment.

The rest of the paper is arranged as follows: In the next section, some related research that uses ensemble learning in the word alignment domain is reviewed. In section III, the underlying technology and algorithms which have been used in this research are presented. Sections IV and V explain the proposed approach and the experimental results respectively. Finally, section VI concludes the paper.

## II. Related work

Combination of multiple aligners has been studied and reported increasingly in recent research. In [5], the authors propose a model using two asymmetric word alignment models to build a stronger symmetric word aligner. Their approach is based on training each of the models IBM Model 1, Model 2 and HMM in both directions and considering intersection of generated alignments. They reportedly achieve up to 29% of reduction in Alignment Error Rate (AER), while their experiments did not gain a remarkable BLEU [18] score compared to a baseline system.

In [7], Wu and Wang proposed an ensemble learning method to improve word alignment based on bagging and cross-validation committees. They have used variations of these methods by exploiting weighted and unweighted voting. For their statistical word aligner, they used IBM Model 4. They also used two direction word alignments to overcome the problem of multi-word alignment. To give weight to the alignments, they measured the association of the source unit and the target unit in an alignment using a Dice coefficient relationship. Their experiments with an English-Chinese corpus showed that making an ensemble of aligners combined with weighted voting obtain much lower error rate – up to 7.4% better than the baseline system.

Wu and Wang developed their work [8] with the AdaBoost algorithm. In this research, they construct an alignment reference set automatically using intersection of bidirectional alignments obtained by IBM Model 4 over whole training data. To be eligible to be added to the reference set, a word alignment link must have a translation probability above a certain threshold as well as its occurring frequency. It has been shown in their results that word alignment is improved using boosting rather than the original word aligner. Their method is able to reach a reduction of 10.28% in error rate for the English to Chinese direction and 21.52% for the Chinese to English direction.

Another research over using boosting algorithms for word alignment has been reported in [11]. Their work implements a revised version of a boosting algorithm that relies on unsupervised learning and puts more concentration on sentence pairs that are identified as well-aligned. They use per link Viterbi alignment probability to weigh sentence pairs in each round of the boosting algorithm. They use IBM Model 4 as their base aligner and apply it in forward and backward directions along with the current set of weights to obtain alignments. Their experimental results compared to the baseline system illustrate some improvement in BLEU score as well as speed and phrase table size.

Xiao et. al. [12] focused on using an ensemble learning method for statistical machine translation. In order to generate the ensemble, they employ a pipeline of weak systems derived from a single SMT engine. They investigate two ensemble approaches: Bagging and Boosting. The training set for the Bagging method comes from sampling over the whole training data with replacement. For Boosting, the distribution over training data is changed to weigh more on samples that achieve a poor translation by weak systems. In their experiments, they used Chinese–English translation with a phrase-based, hierarchical phrase-based and syntax-based translation system. Their results illustrated that using bagging and boosting approaches outperforms in accuracy of translation rather than baseline systems in terms of BLEU metric.

Razmara and Sarkar [13] propose an ensemble learning method based on stacking for SMT in which a base SMT engine is used over a set of variations of training set generated by a k-fold cross-validation method. In their proposed approach, each of the k-1 folds are trained to produce a weak learner system. Then these weak systems are combined together to form the ensemble translator. They have reported an improvement of up to 4 BLEU scores using this approach.

## III. Underlying algorithms and models

Statistical word alignment of a bilingual aligned corpus is a core task of SMT. At the centre of these approaches a model of the translation process is created in which the word alignment is a hidden variable. Along with statistical models, some heuristic models like Dice coefficient are also exploited [19]. Computation of word alignments at these approaches are based on analyzing some association score of a link between the words of the source language and target language.

Each word mapping shows an association $i \rightarrow j = a_i$ in which the alignment is between the source position $i$ to the target position $j = a_i$. The alignment mappings may have some association of $a_i=0$ to indicate that there are no aligned words in the target language for the source word. Here $e_0$ is a symbol of empty words in the target language.

### A. Statistical Alignment Models

Having a source language sentence $f_1^J$ and a target language sentence $e_1^I$, to model the relationship between the source sentence and the target sentence in statistical machine translation, we rely on the translation probability $Pr(f_1^J | e_1^I)$. In this model, a hidden parameter $a=a_1^J$ is introduced that leads us to the alignment model $Pr(f_1^J, a_1^J | e_1^I)$ [14]. This parameter reveals an association from a source position $j$ to a target position $a_j$. The translation model and the alignment model are related based on the following equation:

$$Pr(f_1^J|e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \quad (1)$$

The statistical model is usually affected by some unknown parameters θ which are revealed by learning from the training data. The dependency of the model to the parameters could be stated as the following equation:

$$Pr(f_1^J, a_1^J|e_1^I) = p_\theta(f_1^J, a_1^J|e_1^I) \quad (2)$$

Using a parallel corpus consisting of S sentence pairs, we can perform the training of unknown parameters θ. These parameters are identified by likelihood maximization over the training corpus:

$$\hat{\theta} = \underset{\theta}{argmax} \left\{ \prod_{s=1}^{S} \left[ \sum_{a} p_\theta(\mathbf{f}_s, \mathbf{a}|\mathbf{e}_s) \right] \right\} \quad (3)$$

We use IBM Model 1 and Model 2 as two base statistical models. Model 1 is not affected by word order, while Model 2 uses word order in its probability. These models have a different decomposition for $Pr(f_1^J, a_1^J | e_1^I)$ as expressed in equation (4) and (5) for Model 1 and Model 2 respectively:

$$Pr(f_1^J, a_1^J|e_1^I) = \frac{p(J|I)}{(I+1)^J} \cdot \prod_{j=1}^{J} p\left(f_j \middle| e_{a_j}\right) \quad (4)$$

$$Pr(f_1^J, a_1^J|e_1^I) = p(J|I) \cdot \prod_{j=1}^{J} \left[ p(a_j|j, I, J) \cdot p\left(f_j \middle| e_{a_j}\right) \right] \quad (5)$$

To determine this maximization in statistical models, one useful tool is the EM algorithm [19]. There may be several alignments for a sentence pair, but the best alignment is always the desired one, given by:

$$\hat{a}_1^J = \underset{a_1^J}{argmax}\ p_{\hat{\theta}}(f_1^J, a_1^J|e_1^I) \quad (6)$$

In order to acquire alignment distribution, EM only considers the most likely word connections in the parameter space and ignores the other less likely contributions [20]. At the first step of EM algorithm, we build all possible connections between words of each sentence pair. The point here is that all connections are equally likely. Then we learn from the corpus that some connections occur more frequently. So, the inference would be that more frequent connections results in more likely alignments. After calculating all connection probabilities, the structure hidden in the parallel corpus will be revealed and all source words will be aligned to their counterparts in the target language.

### B. Heuristic Models

In these models, a simple method for extracting word alignments is used based on a similarity measurement between the units of text of the two languages. In many cases, the Dice coefficient is used for similarity measurement. All possible association between the words of the source sentence and those of the target sentence and their score are constructed:

$$dice(e_i, f_j) = \frac{2 \cdot C(e_i, f_j)}{C(e_i) \cdot C(f_j)} \quad (7)$$

At the above equation, *C(e)* shows the number of occurrences of word *e* in the target sentences and *C(f)* is associated to the count of words *f* in the source sentences. *C(e, f)* represents the co-occurrence count of word *e* and word *f* in the parallel corpus. Here, the word alignment could be determined using the largest score:

$$a_j = \underset{i}{\text{argmax}}\{dice(e_i, f_j)\} \quad (8)$$

[19] reports another version of this approach called competitive linking algorithm in which after aligning highest score associations, these alignments are eliminated from the alignment matrix until every word in the source language or those in the target language are aligned.

Unlike statistical models, heuristic models are simple to develop as well as easy to understand. However, some results show that the alignment quality of the Dice coefficient is lower than the statistical models [19]. Och gained the alignment error rate (AER) for the Dice model in the best case something about 30 percent. However, they demonstrated that statistical models outperform the simple Dice algorithm. Despite this, it is suitable for ensemble learning, since we need a learning algorithm that performs better than chance, or in our case an aligner that can align correctly more than 50% of alignments.

Our work is different from Wu et. al. [7] using the Dice coefficient. They have used a custom version of the Dice coefficient to compute the weight of each alignment link that has been provided by the IBM Model 4, and used these weights in an ensemble algorithm. Our work, however, relies on the Dice coefficient just as a base of word the alignment engine to generate word alignment links on training data.

### C. AdaBoost Algorithm

If we have some learners where each of which can perform slightly differently on a training data set, then by combining them together it is possible to produce better results rather than any of those learners individually. This is the main idea behind ensemble learning.

The main algorithm of ensemble learning is AdaBoost which is designed for supervised learning. This algorithm assigns weights to samples based on the difficulty of previous learners to classify the samples. These weights are part of the input for training and are initialized to the same value, 1/N, where N is the number of samples. Several learners are trained over the training set in separate rounds. The weights are updated by each learner based on the past results for each training data obtained from previous learners. An error ($e$) is computed at each iteration according to the summation of all the samples that are misclassified. Then, weights of incorrect predictions are modified by multiplying to $\alpha = e/(1-e)$, whereas the weight of correct predictions remains unchanged. The most important function in this algorithm is computing new weights, which is performed by each learner in its round. Each learner also checks the weights while it is performing classification. There is another variation of AdaBoost that uses weights to generate a subset of training data, and applies the learning algorithm over that subset [6]. We adapted this approach into the proposed algorithm.

## IV. Proposed Algorithm

Our Adaboost algorithm employs weights to generate a sample from the whole training data, and trains over that sample. Figure 1 shows the whole algorithm in detail. In each iteration of the boosting algorithm, we use three base aligners: IBM Model 1, Model 2, and the Dice coefficient. In each round, based on the previous weight set for training sentence pairs, a new subset of training set is considered for base aligners to produce their alignments. Then they consent to an alignment for each sentence pair by majority voting before updating the weights. At the first round, all weights are set to a same value: 1/N. The resampling module picks the data that has weights greater than zero. In this way, all sentences are contributed to the alignment process in the first iteration.

To update the weight of sentences in each round, we use a sentence alignment confidence measurement. Huang [21] defines alignment confidence measure as the geometric mean of the alignment posterior probabilities in bidirectional alignment models. However our alignment models are not homogeneous, so we define an alignment confidence based on voted alignments in each sentence pair.

Suppose for sentence pair (S, T), $A_k \subseteq \{(i, j) \mid 1 \le i \le I,\ 1 \le j \le J\}$ is the alignment set generated by model $k$ in which $I$ and $J$ are the length of the source and target sentences respectively. We combine these alignment sets in such a way that keeps the voting count (frequency) of each alignment link. The resulting set looks like $AT \subseteq \{((i, j), f) \mid 1 \le i \le I,\ 1 \le j \le J,\ f \ge 1\}$. By considering $M$ as the size of this set for the sentence pair (S, T), sentence alignment confidence is then defined as:

$$Confidence(S,T) = \frac{1}{L.M.D} \sum_{(i,j)\in AT} f \qquad (9)$$

in which $L$ is the total number of rounds in the ensemble algorithm and $D$ is the number of alignment models (in our case $D = 3$). In the best case, when all the alignment sets for a sentence pair are identical, the alignment confidence, *Confidence(S, T)*, should be equal to 1. Otherwise, when the consensus between models decreases, the confidence decreases as well. We use complement equation of confidence score (alignment uncertainty) as a factor of updating weights for the sentence pairs. Based on this factor, the ensemble algorithm will concentrate on sentence pairs which have not been aligned well:

$$Uncertainty(S,T) = 1 - Confidence(S,T) \qquad (10)$$

An error rate of the alignment $\varepsilon$ is calculated in each round in which the sentences whose alignment uncertainty is greater than 0.5 are considered. Weights of each sentence pair will be updated two times in each round. In the first instance, a pair's last weight, uncertainty, and model error rate will compute the new weight. At the second time, the average of updated weights is computed and the new weight of each pair is computed based on the distance of each weight to the average weight ($\delta$). If the distance is positive, a new weight will be updated to that, otherwise the new weight will be set to zero. This means, in the next round, the sentence pair with weight 0 will not be picked up to participate in the training process. After several iterations of the ensemble algorithm, and when the learning process terminates, the final alignment selection for each sentence pair is done based on a voting score that is computed as the equation 11:

Input: Bilingual parallel sentences $(S, T) = \{(s_j, t_j) \mid j \in (1...N)\}$
L: Maximum iterations

- $D = 3$
- $Model_1$=IBM-Model1, $Model_2$=IBM-Model2, $Model_3$=Dice,
- Initialize the weights: $W_0 = \{W_{0,j} = 1/N\}\ \forall\ j \in (1...N)$
- for $t = 1$ to $L$
  - For $i$=1 to $D$
    $A_i \leftarrow Model_i(S, T, W_{t-1})$
  - $AT_t$ = Combine $A_i$
  - $Unc_j = 1 - Confidence(S_j, T_j)\ \ \forall\ j \in (1...N)$
  - $\varepsilon_t = \frac{1}{N}\sum_{j=1}^{N} w_{t-1,j} \times Unc_j\ \ \forall\ Unc_j > 0.5$
  - if $\varepsilon_t > 0.5$ break loop
  - $\alpha_t = \frac{1}{2}\log\frac{1-\varepsilon_t}{\varepsilon_t}$
  - $W_{t,j} = W_{t-1,j} \times \exp(\alpha_t \times Unc_j)\ \ \forall\ j \in (1...N)$
  - $avgW = Avg(W_{t,j})$
  - $W_{t,j} = \delta(W_{t,j}, avgW)$

$\delta(x, y)$= **if** $(x-y > 0)$ **return** x-y **else return** 0

Output: Final word alignment
for each alignment (i, k) in $AT(s_j, t_j)$
consider (i, k) so that *v_score(i, k)* > 0.5

Figure 1. Proposed Alignment Ensemble Algorithm

$$v_{score(i,j)} = \frac{h}{R} \times \frac{a_{i,j}}{b_{i,j}} \quad (11)$$

In the above equation, *R* is the number of ensemble rounds in which sentence pair *(S, T)* has been selected for alignment, *h* is the number of times that alignment *(i, j)* appeared in *R* rounds, $a_{i,j}$ is the sum of the votes for alignment *(i, j)* that are greater than half of the number of models, and $b_{i,j}$ is the total sum of the votes for alignment *(i, j)*. The alignments that obtain a score equal or greater than 0.5 are eligible to be or at least be considered as final alignments.

## V. Evaluation

We evaluated a proposed word alignment over a sentence aligned Maori-English corpus which was prepared manually during this research. However, since Maori has very limited bilingual resources, we were only able to collect about 650 sentence pairs. The English side has 8173 words and 52689 characters while the Maori side has 10545 words and 51590 characters. Among these, we selected 50 sentence pairs as our test data and aligned them manually to produce a reference set. The remainder data is used to train the proposed algorithm.

We used the evaluation scheme of Wu and Wang [8] to evaluate the proposed ensemble alignment. Showing the alignment set produced by the proposed algorithm by S, and reference alignment set by R, the evaluation metrics will be as follows:

$$precision = \frac{|S \cap R|}{|S|} \quad (12)$$

$$recall = \frac{|S \cap R|}{|R|} \quad (13)$$

$$fmeasure = \frac{2 \times |S \cap R|}{|S| + |R|} \quad (14)$$

$$AER = 1 - fmeasure \quad (15)$$

The total number of alignments in the reference set is 526. The ensemble algorithm generated 425 total alignments for test data. In order to have a measure of the efficiency of this approach, we performed two other separate experiments with test data: one that just applies IBM Model 2 to generate alignments and the other that just applies the Dice model for word alignment generation. Table 1 presents the statistics of the alignments generated by these three experiments.

TABLE 1 ALIGNMENTS STATISTICS OF THREE EXPERIMENTS

| *Experiment* | *Total alignments* | *Correct alignments* |
|---|---|---|
| Ensemble Model | 425 | 196 |
| IBM Model 2 | 432 | 147 |
| Dice Model | 437 | 120 |

The total number of alignments generated by the ensemble method is somewhat less than the alignments generated by the two other methods, and this is due to the voting scheme we used in the output generation in the ensemble algorithm. On the other hand, the total correct alignments have been increased by this method for the same reason. Table 2 shows the precision, recall, and AER for these models.

TABLE 2- EVALUATION COMPARISON OF ALIGNMENT MODELS

| **Model** | **Precision** | **Recall** | **AER** |
|---|---|---|---|
| Ensemble | 0.46 | 0.37 | 0.59 |
| IBM Model 2 | 0.34 | 0.28 | 0.70 |
| Dice | 0.27 | 0.23 | 0.76 |

From Table 2, the ensemble model shows to achieve better results in all metrics than IMB Model 2 and Dice model. It has an improved alignment error rate by 15% and 22% compared to IBM Model2 and Dice model respectively. Improvement in precision and recall also has been gained by at least 34% and 32% respectively, compared to the two others.

## VI. CONCLUSION AND FUTURE WORKS

In this paper, we present a new approach for word alignment based on AdaBoost algorithm which uses statistical and heuristic alignment models as well as a voting model to produce the ensemble alignments of all underlying alignment candidates.

Our proposed approach demonstrates significant improvement for alignment error rate despite training the algorithm on a tiny set of bilingual sentence pairs. An obvious consequence of having a small-sized training data is that the alignment error rate will not be very low; however the point is that having different alignment models improves the quality of alignment.

Comparison of the proposed weighting mechanism to other weighting approaches is intended to be carried out in the next phases of this work. Applying the proposed ensemble model to a larger set of training sentence pairs and using the alignment model in the context of statistical machine translation are the other intended future works of this study.

## REFERENCES


[1] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," Computational Linguistics, vol. 29, no. 1, pp. 19-51, 2003.

[2] A. Lopez and P. Resnik, "Word-Based Alignment, Phrase-Based Translation: What's the Link?," Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pp. 90-99, Cambridge, 2006.

[3] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," Computational Linguistics, vol. 19, no. 2, pp. 263–311, 1993.

[4] S. Vogel, H. Ney and C. Tillmann, "HMM-based Word Alignment in Statistical Translation," In COLING '96: The 16th International Conference on Computational Linguistics, pp. 836-841, Copenhagen, Denmark, 1996.

[5] P. Liang, B. Taskar and D. Klein, "Alignment by agreement," In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pp. 104–111, New York City, USA, 2006.

[6] S. Marsland, “Machine Learning, An algorithmic perspective,” CRC Press, 2009.

[7] H. Wu and H. Wang, "Improving Statistical Word Alignment with Ensemble Methods," IJCNLP, 2005.

[8] H. Wu and H. Wang, "Boosting Statistical Word Alignment,” 10th machine translation summit, pp. 313-320, 2005.

[9] H. Wu, H. Wang and Z. Liu, "Boosting statistical word alignment using labeled and unlabeled data," In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 913–920, Sydney, Australia, July 2006.

[10] S. Huang, K. Li, X. Dai, J. Chen, "Improving Word Alignment by Semi-supervised Ensemble," Fourteenth Conference on Computational Natural Language Learning (CoNLL 2010), 07/2010.

[11] S. Ananthakrishnan, R. Prasad and P. Natarajan, "An unsupervised boosting technique for refining word alignment," Spoken Language Technology Workshop (SLT), 2010.

[12] T. Xiao, J. Zhu and T. Liu, "Bagging and Boosting statistical machine translation systems," Artificial Intelligence, vol. 195, pp. 496–527, 2013.

[13] M. Razmara and A. Sarkar, "Stacking for Statistical Machine Translation," Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 334–339, Sofia, Bulgaria, 2013.

[14] F. Och and H. Ney, "Improved Statistical Alignment Models," Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000.

[15] A. L. Lagarda and F. Casacuberta, "Applying boosting to statistical machine translation," 12th EAMT conference, Hamburg, Germany, 2008.

[16] T. Xiao, J. Zhu, M. Zhu and H. Wang, "Boosting-based System combination for machine translation," in Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), Uppsala, Sweden, pp. 739–748, 2010.

[17] M. Surdeanu and C. D. Manning, "Ensemble models for dependency parsing: cheap and good?," In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT ’10, pp. 649–652, Stroudsburg, PA, USA, 2010.

[18] K. Papineni , S. Roukos , T. Ward and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, 2002.

[19] I. D. Melamed, "Models of translational equivalence among words," *Computational Linguistics*, vol. 26, no. 2, pp. 221–249, 2000.

[20] K. Knight and P. Koehn, "What's new in statistical machine translation," University of South California, 2003.

[21] F. Huang, "Confidence measure for word alignment," In Proceedings of Association for Computational Linguistics, 2009.