Fast incremental active learning for statistical machine translation

Jesús González-Rubio[1], Daniel Ortiz-Martinez[2], and Francisco Casacuberta[2]

[1] Instituto Tecnológico de Informática,
Universitat Politècnica de València
46021 Valencia, Spain
jegonzalez@iti.upv.es
[2] Departamento de Sistemas Informáticos y Computación,
Universitat Politècnica de València
46021 Valencia, Spain
{dortiz,fcn}@dsic.upv.es

Resumen Different works show that the application of active learning techniques within statistical machine translation improves the quality of the final translations while minimizing the number of bilingual sentences required to train the system. All these previous works look for the best sentence sampling strategy while using the batch learning paradigm to retrain the translation model. Unfortunately, batch learning for statistical machine translation typically requires many hours to train a system of reasonable size. This fact limits the practical application of active learning for statistical machine translation. In this work, we propose to apply incremental learning techniques to retrain the translation model in an active learning scenario for statistical machine translation. Experiments show that incremental learning allows us to reduce by several orders of magnitude the training time per sentence while yielding similar improvements in the translation quality with respect to batch learning.

1. Introduction

Active learning[3] (AL) is a form of supervised machine learning in which the learning algorithm is able to interactively query the user (or some other information source) to obtain the desired output at new data points. AL algorithms aim to achieve high accuracy using as few labeled instances as possible. Therefore, it is well-motivated in those machine learning problems where data may be abundant but labels are scarce or expensive to obtain and the cost to retrain the learner is low. In this paper, we study the application of AL to statistical machine translation (SMT). The AL scenario for SMT differs from the typical AL scenario in the high computational cost required to retrain the models. This fact limits the widespread application of AL in commercial SMT systems, since companies offering translation services require not only good translations but also an increasing productivity.

Recently, different AL strategies [6, 7, 1, 2] have been applied to SMT. These previous works deal with the problem of labelling the fewest new data to obtain an SMT

[3] In statistics literature it is sometimes also called optimal experimental design

system of a certain quality. Starting from an initial SMT model (empty or initialized with bilingual data) new monolingual data is selected to be translated and added to the SMT model until a certain level of performance is reached. To add new data to the SMT model, these works make use of the well-known batch learning paradigm. In the batch learning paradigm, the addition of new training data implies a full retraining of the translation models with all the training samples that have been seen. This process is extremely inefficient in computational terms.

In this paper, we focus on the problem of retraining an SMT model when new bilingual data is made available. Typically, the batch learning approach requires many hours to retrain a system of reasonable size. We present an AL scenario for SMT where the SMT system is retrained following incremental learning techniques. These incremental learning techniques dramatically reduce the time required to retrain the translation model, allowing us to efficiently apply AL to SMT.

The rest of the paper is organized as follows: section 2 introduces the SMT framework, section 3 describes the AL scenario for SMT using incremental training and presents the applied incremental learning techniques, section 3 describes the experimentation setup we followed and presents our main results and, finally, section 5 summarizes the contributions of our work.

2. Statistical Machine Translation

Given a sentence $\mathbf{f}$ from a source language $\mathcal{F}$ to be translated into a target sentence $\mathbf{e}$ in a target language $\mathcal{E}$, the fundamental equation of SMT [3] is the following:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \{Pr(\mathbf{e} \mid \mathbf{f})\} \tag{1}$$

$$= \arg\max_{\mathbf{e}} \{Pr(\mathbf{f} \mid \mathbf{e})\, Pr(\mathbf{e})\} \tag{2}$$

where $Pr(\mathbf{f} \mid \mathbf{e})$ is approximated by a *translation model* that represents the correlation between the source and the target sentence and where $Pr(\mathbf{e})$ is approximated by a *language model* representing the well-formedness of the candidate translation $\mathbf{e}$.

State-of-the-art SMT systems follow a log-linear approach [14], where direct modelling of the posterior probability $Pr(\mathbf{e} \mid \mathbf{f})$ of Equation (1) is used. In this case, the decision rule is given by the expression:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f}) \right\} \tag{3}$$

where each $h_m(\mathbf{e}, \mathbf{f})$ is a feature function representing a statistical model and λ_m its weight.

Current SMT systems are based on the use of phrase-based models [10] as translation models. The basic idea of phrase-based translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally to reorder the translated target phrases in order to compose the target sentence. If we summarize all

the decisions made during the phrase-based translation process by means of the hidden variable $\tilde{a}_1^K$, we obtain the expression:

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{K,\tilde{a}_1^K} Pr(\tilde{f}_1^K, \tilde{a}_1^K \mid \tilde{e}_1^K) \tag{4}$$

where each $\tilde{a}_k \in \{1 \dots K\}$ denotes the index of the target phrase $\tilde{e}$ that is aligned with the k-th source phrase $\tilde{f}_k$, assuming a segmentation of length K.

According to Equation (4), and following a maximum approximation, the problem stated in Equation (2) can be reframed as:

$$\hat{\mathbf{e}} \approx \arg\max_{\mathbf{e},\mathbf{a}} \left\{ p(\mathbf{e}) \cdot p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \right\} \tag{5}$$

Following the log-linear approach in Equation (3), Equation (5) can be rewritten as:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e},\mathbf{a}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{a}, \mathbf{f}) \right\} \tag{6}$$

which is the approach that we follow in this work.

3. Incremental Active Learning for SMT

Starting from an initial SMT model, the AL problem aim at minimizing the human effort in annotating new sentences that, when added to the training data, make the retrained SMT model achieve a certain level of performance. Thus, given a (possibly small) set of bilingual data $\mathcal{L}$ and a (possibly large) pool of monolingual data $\mathcal{U}$ the objective is to iterativelly select a subset of highly informative sentences from $\mathcal{U}$ to be presented to a human expert for translation. Highly informative sentences are those that, together with their translations, help the retrained SMT system "quickly" reach a certain level of translation quality. This scenario is known as AL with pool-based sampling [11].

Algorithm 1 Pseudocode of the proposed incremental AL scenario for SMT.

input : M_0 (SMT model trained on $\mathcal{L}_0$)
$\mathcal{U}_0$ (unlabeled set)
$\mathcal{T}$ (test set)

begin
  for $s = 0, 1, \dots$ **do**
    $\mathbf{f}_s$=SelectBest($\mathcal{U}_s$) $\mathbf{e}_s$=Annotate($\mathbf{f}_s$) $\mathcal{U}_{s+1} = \mathcal{U}_s - \mathbf{f}_s$
    M_{s+1}=IncrTrain(M_s, ($\mathbf{f}_s$, $\mathbf{e}_s$)) CheckQuality(M_s, $\mathcal{T}$)
end

Algorithm 1 describes the AL scenario we propose for SMT. We start with a monolingual dataset $\mathcal{U}_0$ and an initial SMT model M_0 trained on the initial bilingual dataset $\mathcal{L}_0$. Then an iterative process starts, at step s, we score all the sentences in $\mathcal{U}_s$ according

to a sentence selection strategy and retrieve the single best-scoring sentence $\mathbf{f}_s$ (line 3). This sentence is translated (line 3) and removed from the monolingual dataset (line 3). Finally, the current SMT model M_s is incrementally updated adding the sentence pair $(\mathbf{f}_s, \mathbf{e}_s)$ (line 3). This process is repeated iteratively until the desired level of translation quality (measured in a test set $\mathcal{T}$) is met (line 3).

Some sentence selection strategies require the translation of the given monolingual data by the SMT model. For those strategies, Algorithm 1 would include an extra step translating the monolingual data before the selection of the best scoring sentence. However, that does not affect the discussion in the rest of the paper, so, we remove that step for simplicity.

The main difference between our AL scenario with incremental training and the rest of AL scenarios in the SMT literature relies on the retraining step (line 3 in Algorithm 1). Previous works [6, 1, 2] perform a full retraining of the SMT model using the bilingual data so far and the new sentence pairs, i.e. at each iteration, the SMT model is learned from scratch, which is, obviously, a highly inefficient process. On the contrary, we propose to incrementally update the SMT model adding new sentence pairs as they are selected. At each iteration, we update the current model with the new sentence pair.

Our approach allows a much more efficient use of the computational resources. Additionally, as we add the sentences one by one we can stop the annotation process as soon as the quality level is met, thus, minimizing the number of annotated sentences. This approach is computationally infeasible using batch learning [1].

The next two sections describe the incremental techniques used to update the SMT model and the sentence selection strategies that we implemented in the experimentation.

3.1. SMT Incremental Training

We propose to import to our AL scenario the incremental training techniques described in [15]. In that work, the authors define an incrementally updateable SMT model for its application in the interactive machine translation framework. Such SMT model is able to process new training samples one by one, with constant computational complexity (i.e. its complexity does not depend on the number of previously seen training samples).

The SMT system described in [15] uses a log-linear model to generate its translations. According to Equation (6), we introduce a set of seven feature functions (from h_1 to h_7): an n-gram language model (h_1), an inverse sentence-length model (h_2), inverse and direct phrase-based models (h_3 and h_4 respectively), a target phrase-length model (h_5), a source phrase-length model (h_6), and a distortion model (h_7). The details for each feature function are listed below:

- **n-gram language model (h_1)**:
  $h_1(\mathbf{e}) = \log(\prod_{i=1}^{|\mathbf{e}|+1} p(e_i|e_{i-n+1}^{i-1}))$,[4] h_1 can be implemented by means of smoothed n-gram language models. Here we adopt an interpolated n-gram model with Kneser-Ney smoothing.

[4] $|\mathbf{e}|$ is the length of $\mathbf{e}$, e_0 denotes the *begin-of-sentence* symbol, $e_{|e|+1}$ is the *end-of-sentence* symbol and $e_i^j \equiv e_i ... e_j$

- **source sentence-length model (h_2)**:
  $h_2(\mathbf{f}, \mathbf{e}) = \log(p(|\mathbf{f}| \mid |\mathbf{e}|))$, h_2 can be implemented by means of a set of gaussian distributions whose parameters are estimated for each source sentence length.
- **inverse and direct phrase-based models (h_3, h_4)**:
  $h_3(\mathbf{e}, \mathbf{a}, \mathbf{f}) = \log(\prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k}))$, where h_3 is implemented with an inverse phrase-based model. This phrase-based model is smoothed with an HMM-based alignment [17] model by means of linear interpolation.
  Analogously h_4 is defined as: $h_4(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \log(\prod_{k=1}^{K} p(\tilde{e}_{\tilde{a}_k}|\tilde{f}_k))$
- **target phrase-length model (h_5)**:
  $h_5(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \log(\prod_{k=1}^{K} p(|\tilde{e}_k|)$, this feature is modelled by means of a geometric distribution. The geometric distribution penalizes the length of the target phrases.
- **source phrase-length model (h_6)**:
  $h_6(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \log(\prod_{k=1}^{K} p(|\tilde{f}_k| \mid |\tilde{e}_{\tilde{a}_k}|))$, a geometric distribution can be used to model h_6, such distribution penalizes the difference between the source and target phrase lengths.
- **distortion model (h_7)**:
  $h_7(\mathbf{a}) = \log(\prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}))$, again, this feature function can be modelled by means of a geometric distribution that penalizes reordering.

In order to incrementally train the log-linear model, a set of sufficient statistics that can be incrementally updated is maintained for each feature function[5]. If the estimation of the statistical model does not require the use of the expectation–maximization (EM) algorithm [5] (e.g. n-gram language models), then it is generally easy to incrementally update the model given a new training sample. By contrast, if the EM algorithm is required (e.g. word alignment models), the estimation procedure has to be modified, since the conventional EM algorithm is designed for its use in batch learning scenarios. For those models, the incremental version of the EM algorithm [12] is applied.

The parameters of the n-gram language model with Kneser-Ney smoothing that implements the h_1 feature function can be incrementally adjusted with an appropriate algorithm described in [15]. Since the estimation does not involve the EM algorithm, the algorithm is relatively simple.

Regarding the h_2 feature function, its incremental estimation requires updating the parameters of a set of gaussian distributions. This problem has been extensively studied in the literature, specifically, we apply the incremental update rules given in [8].

Feature functions h_3 and h_4 implement inverse and direct smoothed phrase-based models respectively. Since phrase-based models are symmetric models, only an inverse phrase-based model is maintained (direct probabilities can be efficiently obtained using appropriate data structures). To incrementally update the phrase-based models, for each training pair, we need to extract a set of phrase pairs that are consistent with a word-level alignment matrix generated by means of a word alignment model. This alignment model is updated using the incremental version of the EM algorithm.

Finally, the parameters of the geometric distributions associated to the feature functions h_5, h_6 and h_7 are left fixed. Because of this, there are no statistics to store for these feature functions.

[5] A sufficient statistic for a statistical model is a statistic that captures all the information that is relevant to estimate the model.

A detailed description of the update algorithm for each of the models in the log-linear combination is presented in [15].

3.2. Sentence Selection Strategies

Previous works on AL applied to SMT study the optimal sentence selection strategy to be used in order to minimize the number of required sentences. On the contrary, we designed the sentence selection strategies to point out the differences in terms of computational complexity between incremental training and batch training.

To this end, we implement three different sentence selection strategies: *longest*, *shortest* and *maxcoverage*. The first two strategies are self explanatory. Maxcoverage strategy prefers the sentences that contain phrases that are frequent in the monolingual data $\mathcal{U}$ and do not appear in the bilingual data $\mathcal{L}$. Specifically, maxcoverage selects the sentence that maximizes the score $s_m(\mathbf{f})$, which is defined as follows:

$$s_m(\mathbf{f}) = \sum_{x \in \mathcal{N}(\mathbf{f})} A_{\mathcal{L}}(x) * C_{\mathcal{U}}(x) \,, \tag{7}$$

where $\mathcal{N}(\mathbf{f})$ stands for the set of n-grams in sentence $\mathbf{f}$, $A_{\mathcal{L}}(x)$ equals 1 if n-gram x does not appear in the bilingual data so far and 0 otherwise, and $C_{\mathcal{U}}(x)$ represent the count of x in the monolingual data.

These sentence selection strategies are designed to stress different aspects of the retraining process. Longest and shortest aim at testing the impact of sentence size in retraining time, whereas maxcoverage is intended to maximize the new information being added to the model, thus, increasing the time cost of model updates (adding new parameters to the statistical model is more time consuming than updating the values of existing parameters).

4. Experiments

We tested the proposed framework in a simulated AL setting. We trained two initial SMT models using the data from the Europarl corpus as distributed for the shared task in the NAACL 2006 workshop on SMT. Specifically, we trained an incremental model with the features described in section 3.1 to test incremental retraining and a standard Moses model [9] for the batch retraining. The weights of the different translation features were tuned by means of minimum error–rate training [13] executed on the Europarl development corpus. Once the initial SMT models were trained, we simulated the interactive translation of a set of sentences extracted from the News Commentary corpus [4], using the reference translations as the final translations that the simulated user wants to obtain. We follow the techniques given in section 3.1 to update the incremental SMT model and the standard Moses pipeline for the batch SMT retraining. Finally, we used the test partition of News Commentary corpus to monitor the quality of the retrained system after each iteration of the AL algorithm. The size of these corpora is shown in Table 1. We chose the News Commentary corpus to carry out our experiments for two reasons. First, its size is large enough to test our AL techniques in the long term.

corpus	use	sentences	words (Spa/Eng)
Europarl	$\mathcal{L}$	731K	15M/15M
News Commentary	$\mathcal{U}$	51K	1490K/1220K
	$\mathcal{T}$	2K	56K/49K

Cuadro 1. Size of the Spanish–English corpora used in the experiments. K and M stand for thousands and millions of elements respectively.

Second, since it constitutes an out-of-domain test set, it allowed us to define a real-world difficult task for applying our techniques.

We measure translation quality by means of the BLEU [16] score. In addition to this, we will also measure the time complexity of our proposed incremental learning techniques with respect to batch learning. To this end, we report the time, measured in seconds, required to retrain the SMT system with one new sentence pair.

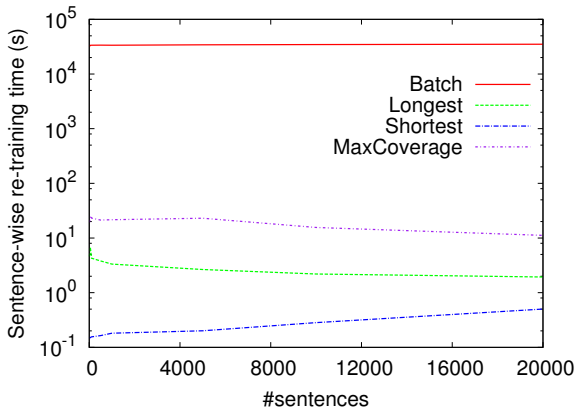
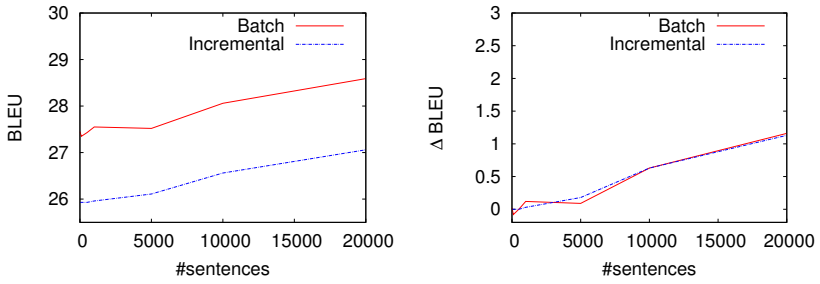


Figura 1. Retraining time to add one sentence pair for the incremental training technique with different sentence selection strategies (shortest, longest and maxcoverage) and for the batch training technique using the maxcoverage strategy.

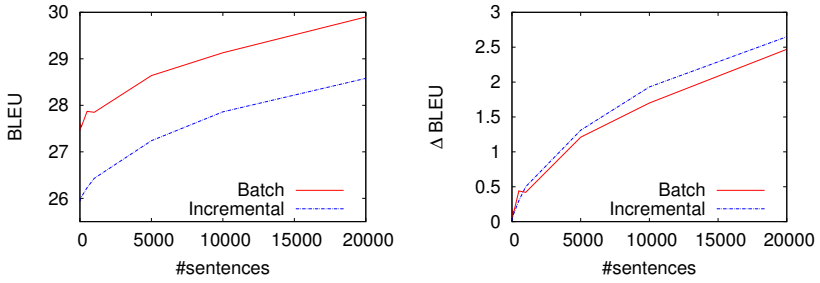
In our first experiment, we evaluated the performance, measured in computational time, of the batch and the incremental training techniques. Figure 1 shows the time required to retrain the SMT model when a new sentence pair is added[6]. We present results for incremental training using the three different sentence selection strategies described in section 3.2 (shortest, longest and maxcoverage). Regarding batch training, its performance was very similar for the three different sentence selection strategies. Therefore, for the sake of simplicity, we report results only for the maxcoverage strategy. According to the figure, the proposed incremental learning techniques reduced the batch retraining time by several orders of magnitude.

Regarding the different sentence selection strategies, results show significant differences in the retraining time. If we compare the results of the shortest and longest strategies, we see that longest took much more retraining time. This confirms the intuition that the

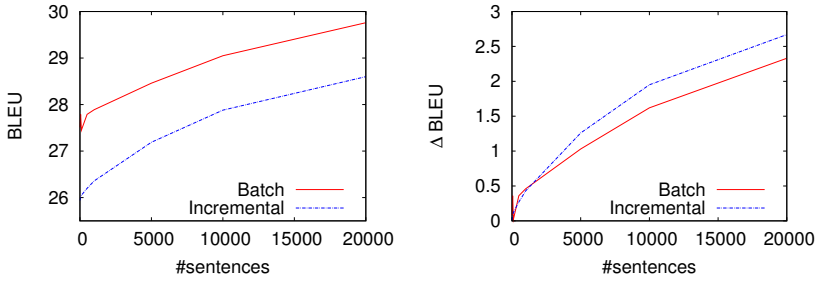
[6]For batch training, it is the time needed to retrain the model with all the available bilingual data.



(a) shortest



(b) longest



(c) maxcoverage

Figura 2. Absolute (left) and relative (right) improvements in translation quality using batch and incremental retraining for the three sentence selection strategies.

number of words in the sentence is directly related with the retraining time. Results for the maxcoverage showed even longer retraining times. This result shows that the number of new words (or *n*-grams) added to the SMT model has a strong impact in the time cost of incremental training.

In our second experiment, we studied the translation quality of the SMT systems trained with the incremental and the batch training techniques. Our incremental SMT system and the Moses system use different statistical models and decoding strategies, thus, they achieved different translation quality. Due to this differences in translation quality, Figure 2 displays both absolute and relative translation quality improvements. Specifically, the left column shows the absolute BLEU score of the incremental and batch retrained systems for the three different sentence selection strategies, while the right column shows the relative BLEU improvements of the retrained systems with

respect to the systems trained only with the initial bilingual data. Despite the differences in absolute BLEU scores, relative improvements show that both retraining techniques have similar performance.

These results show that our proposed incremental training approach for AL reduces the retraining time by several orders of magnitude with respect to the batch learning approach while obtaining similar improvements in translation quality.

5. Summary

In this paper, we have presented techniques to efficiently apply the AL scenario to SMT. For this purpose, in our AL scenario the SMT system is retrained following incremental learning techniques.

According to the experimentation we carried out, the use of incremental training techniques instead of the more widespread batch training algorithm makes a much more efficient use (several orders of magnitude) of the computational resources while obtaining similar improvements in the translation quality.

Acknowledgements

Work supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), the iTrans2 (TIN2009-14511) project and the UPV under grant 20091027. Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Generalitat Valenciana under grant Prometeo/2009/014.

References

1. Ambati, V., Vogel, S., Carbonell, J.: Active learning and crowd-sourcing for machine translation. In: Proceedings of the 7th international conference on Language Resources and Evaluation. pp. 2169–2174 (2010)
2. Bloodgood, M., Callison-Burch, C.: Bucking the trend: large-scale cost-focused active learning for statistical machine translation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 854–864 (2010)
3. Brown, P., Della Pietra, S., Della Pietra, V., Mercer, R.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19(2), 263–311 (1993)
4. Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J.: (Meta-) evaluation of machine translation. In: Proceedings of the Second Workshop on Statistical Machine Translation. pp. 136–158 (2007)
5. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
6. Haffari, G., Roy, M., Sarkar, A.: Active learning for statistical phrase-based machine translation. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 415–423 (2009)

7. Haffari, G., Sarkar, A.: Active learning for multilingual statistical machine translation. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 181–189 (2009)
8. Knuth, D.E.: Seminumerical Algorithms, The Art of Computer Programming, vol. 2. Addison-Wesley, Massachusetts, 2nd edn. (1981)
9. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions. pp. 177–180 (2007)
10. Koehn, P., Och, F., Marcu, D.: Statistical phrase-based translation. In: Proceedings of Human Language Technologies: The 2003 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 48–54 (2003)
11. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 3–12 (1994)
12. Neal, R.M., Hinton, G.E.: A view of the em algorithm that justifies incremental, sparse, and other variants. In: Proceedings of the NATO Advanced Study Institute on Learning in graphical models. pp. 355–368 (1998)
13. Och, F.: Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. pp. 160–167 (2003)
14. Och, F., Ney, H.: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 295–302 (2002), best paper award
15. Ortiz-Martínez, D., García-Varea, I., Casacuberta, F.: Online learning for interactive statistical machine translation. In: Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 546–554 (2010)
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
17. Vogel, S., Ney, H., Tillmann, C.: HMM-based word alignment in statistical translation. In: Proceedings of the 16th International Conference on Computational Linguistics. pp. 836–841 (1996)