An Empirical Study of Segment Prioritization for Incrementally Retrained Post-Editing-Based SMT

Jinhua Du jdu@computing.dcu.ie
Ankit Srivastava asrivastava@computing.dcu.ie
Andy Way away@computing.dcu.ie
ADAPT Centre, School of Computing, Dublin City University, Ireland

Alfredo Maldonado-Guerra maldonaa@tcd.ie
David Lewis dave.lewis@cs.tcd.ie
ADAPT Centre, Trinity College Dublin, Ireland

Abstract

Post-editing the output of a statistical machine translation (SMT) system to obtain high-quality translation has become an increasingly common application of SMT, which henceforth we refer to as post-editing-based SMT (PE-SMT). PE-SMT is often deployed as an incrementally retrained system that can learn knowledge from human post-editing outputs as early as possible to augment the SMT models to reduce PE time. In this scenario, the order of input segments plays a very important role in reducing the overall PE time. Under the active learning-based (AL) framework, this paper provides an empirical study of several typical segment prioritization methods, namely the cross entropy difference (CED), n-grams, perplexity (PPL) and translation confidence, and verifies their performance on different data sets and language pairs. Experiments in a simulated setting show that the confidence of translations performs best with decreases of 1.72-4.55 points TER absolute on average compared to the sequential PE-based incrementally retrained SMT.

1 Introduction

In recent years, SMT systems have been widely deployed into the translator's workflow in the localization and translation industry to improve productivity, refereed to as post-editing-based SMT. However, in most cases, current SMT systems cannot generate high-quality translations, so human effort is usually required. With the help of incrementally improved SMT systems, the productivity of translators/post-editors can be significantly increased due to the early learning of knowledge from the previously post-edited segments (Guerberof, 2009; Plitt and Masselot, 2010; Carl et al., 2011; OBrien, 2011; Zhechev, 2012; Guerberof, 2013). Furthermore, the order of input segments has been found to have a significant impact on the overall PE-time, i.e., an optimized sequence of input segments can reduce the overall PE-time compared to the typical chronological sequence (Dara et al., 2014).

Regarding the PE-SMT, the incremental retraining can be roughly categorized into two different scenarios, namely the segment-level online incremental retraining (segment mode) (Levenberg et al., 2010; Denkowski et al., 2014) and batch-level incremental retraining (batch mode) (Hardt and Elming, 2010; Henríquez Q. et al., 2011; Mathur et al., 2013; Simard and Foster, 2013; Dara et al., 2014; Bertoldi et al., 2014). The former takes one post-edited segment per retraining cycle to immediately update the models, which requires rapid incremental processing of the word alignment, phrase/rule generation, language model and parameters tuning etc., while the latter firstly accumulates a batch of segments, and then performs the incremental retraining process to update the system. The batch-level mode can perform the incremental retraining process in the background while the translators/post-editors continue to work on the next batch of segments. From the point of view of parameter estimation, the former can promptly adapt its feature weights to the newly post-edited segment and learn the translator's knowledge, but the frequent change of weights might make the system unstable; the latter adapts the parameters on an average level of segments in a batch, which can relatively keep the system more robust, however, it cannot learn the knowledge as early as possible and cannot demonstrate a quick response to translator's practice and preference. In our task, in order to better show the impact of the order of the input segments on the PE time, we select the batch-level incremental retraining SMT as our experimental platform.

The order in which post-editors review and correct machine-translated segments has an impact on the evaluation score (PE time in our case) of the incrementally retrained PE-SMT systems. That is, assuming that post-editors work on batches, and after post-editing each batch the SMT system is dynamically retrained, the order of segments in these batches will have an impact on how quickly the overall translation performance grows. The expectation is that if post-editors work first on the segments that are most informative or most difficult to translate for SMT, the SMT system will learn most from the corrections, and as a consequence, translation quality will increase more steeply in the following retraining iterations. In doing so, it is possible to devise a process in which the most experienced and potentially more expensive post-editors/translators tackle the first few batches of segments, leaving the rest of the segments to either be worked upon by less experienced and potentially cheaper post-editors/translators, or to be left completely unedited, depending on the quality vs. cost requirements of the actual translation project at hand. Therefore, in this paper, we carry out an empirical study on several different mainstream segment prioritization strategies, and then investigate the factors that closely correlate to the effectiveness of the methods.

The main contributions of this paper include:

- Confidence of translation and perplexity methods are proposed to reorder the input segments in the AL-based dynamically retrained SMT.

- A deep comparison and investigation of different segment prioritization methods for PE-SMT using different data sets and language pairs.

- A detailed data and results analysis of the correlation between the reordering score and the factors.

- Our experiments show that the unnormalized confidence of translations performs best in all tasks and gains around 1.72 to 4.55 TER (Snover et al., 2006) absolute on average.

2 Related Work

The purpose of the input segment prioritization is to reduce the overall PE time to improve productivity and to reduce the cost. In this scenario, the involvement of human effort implies that the segment prioritization process can be regarded as AL framework-based PE-SMT. In this framework, the input segments are ranked based on the information or uncertainty contained therein. In this section, we will introduce the related work in terms of two aspects: AL-based framework for PE-SMT, and the incrementally retrained PE-SMT.

The practical active learning framework for SMT was firstly proposed in Haffari et al. (2009) where a number of high-quality parallel data are acquired from large-scale monolingual data. Relatively inexpensive human costs are iteratively used to translate information-rich sentences. Experimental results show that generally the translation unit-based selection strategies, namely phrases and n-grams, performed best compared to other methods such as random selection, translation confidence, inverse model etc. However, in their work, the AL framework is used for low-resource SMT rather than the PE-SMT scenario. Furthermore, it is a static retraining process in which the test set is constant per iteration, and the retraining procedure is not incremental.

Gonzalez-Rubio et al. (2012) apply AL to the interactive MT in which AL techniques are used to select the most informative sentences to reduce human effort for a given translation quality. Experimental results show that applying AL techniques in an interactive MT setting can prove a better tradeoff between required human effort and final translation quality.

To the best of our knowledge, the most relevant previous work is that of Dara et al. (2014), which proposes a Cross Entropy Difference (CED) criterion to prioritize input segments in an AL framework for PE-based incremental MT update applications. The fundamental goal is to reduce the overall PE time rather than aiming at reducing human effort. The proposed CED method calculates the rank score by the entropy difference of a sentence s in the untranslated corpus (or the incremental data) U and the current training corpus L. The higher the score, the more informative the sentence is and the greater the possibility of the sentence being more highly ranked. Experimental results on the industrial data in a simulated setting show that the proposed method significantly reduces the TER score compared to the random and sequential order. In their work, Dara et al. (2014) used batch mode for the incrementally retrained PE-SMT with the CED only considering the information of the source side of the data in order to keep the costs to a minimum for the commercial PE MT applications. However, in the practical scenario, we can take the information of the target side (e.g. translations) into account in batch mode without a significant increase in extra time and human costs by pre-translating the remaining batches in the background while post-editing the current batch. In doing so, we propose to use the confidence of MT translations to rank the segments.

Regarding the incrementally retrained SMT, the most challenging and time-consuming steps are the word alignment and the phrase/rule generation. Ortiz-Martinez et al. (2010) incrementally update the feature values of the phrase table by extracting new phrases from the new sentence pairs based on the pre-stored statistics related to the feature scores. Hardt and Elming (2010) propose a sentence-level retraining scheme in which a local phrase table is created and incrementally updated as a file is translated and post-edited. In their work, a modified revision of GIZA++ (Och and Ney, 2003) is used to approximate word alignments of a newly translated sentence to reduce the incremental training time, and then an additional phrase table is produced from the newly aligned sentences with higher priority. The experiments show the efficiency of the incremental retraining system.

In the incrementally retrained PE-SMT system, suffix arrays (Callison-Burch et al., 2005; Zhang and Vogel, 2005) are a very efficient technique for the incremental retraining process. Levenberg et al. (2010) introduce a dynamic suffix array to incorporate new training text to the current training data. Denkowski et al. (2014) propose an online model adaptation for PE-SMT in which three methods are used for incremental model adaptation: adding new data to a suffix array-indexed bitext from which grammars are extracted, updating a Bayesian language model with incremental data, and using an online MIRA (Crammer and Singer, 2003) to update the parameters. The simulated experiments show that significant improvement in MT quality is achieved when these methods are used individually and in tandem. Germann (2014) proposes a dynamic phrase table strategy for an interactive PE-SMT that computes phrase table entries on demand by sampling a suffix array-indexed bitext. Experiments show that without loss of translation quality, the sampling phrase table achieves good performance in terms of speed. In our task, we use this dynamic phrase table for incremental retraining in Moses (Koehn et al., 2007).

3 The Incrementally Retrained PE-SMT Paradigm

In the post-editing scenario, humans are involved to continuously edit MT outputs into high-quality translations. As discussed in Dara et al. (2014), the fundamental goal of input segment prioritization for PE-SMT is to reduce the overall PE time taken to complete a translation job. The crucial step is to first select the most uncertain sentences or most informative sentences for post-editing in order to learn as much knowledge as possible from these sentences. The workflow of an AL-based incrementally retrained PE-SMT system is as shown in Figure 1.

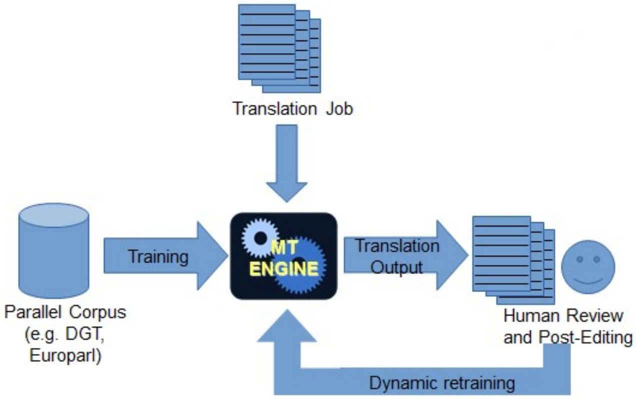

Figure 1: The workflow of active learning-based incrementally retrained PE-SMT

In Figure 1, the translations of the input segments are post-edited and the corrected translations are used for incremental update of the models. The process is repeated until the incremental data (or translation job) is finished. In a typical PE scenario, post-editors are presented with SMT outputs in chronological order (i.e. sequentially) of the input segments. However, an optimized order of the input segments in a translation job can significantly reduce the overall PE time.

In our scenario, the PE time is simulated using TER score between the MT output and the reference translations for the sentences in each batch. The overall performance of the segment prioritization method is evaluated by the average TER score for all the batches.

4 Methodology

To prioritize the input segments, an importance score or uncertainty score for the sentence s must be calculated under some metric, which can be formalized as follows:

Given an initial parallel training corpus $L := \{(f_i, e_i)\}$ and a monolingual corpus (translation job) $U := \{f_j\}$, the goal of the segment prioritization is to rank a sentence s with the score $\phi(s)$ under a scoring metric F. This process can be defined as a *triple* in (1):

$$\phi(s) = F(s, U, L) \tag{1}$$

Clearly, we can see that the scoring metric F is most important in a prioritization algorithm.

We use the sequential order of input segments as our baseline.[1] In the following sections,

[1] The random and sequential methods have similar performance in Dara et al. (2014), so we only use sequential as the baseline.

we carry out an empirical study on different information-driven prioritization methods.

4.1 Confidence of Translations (Confidence)

In the decoding process, a translation output $\hat{e}$ is produced with the probability $p(e|f)$ that is calculated by different features, such as bidirectional lexical probabilities, language model etc. It can be treated as a confidence score for the translation because it reflects the translation difficulty or uncertainty of the source segment in some sense.

Generally, the probability $p(e|f)$ is influenced by two aspects, namely the out-of-vocabulary (OOV) words and the sentence length (c.f. Section 6). For human translators, these two aspects are often more time-consuming. That is, a long sentence with many OOVs will take much more time to post-edit. Therefore, intuitively, the unnormalized confidence score of translations can better measure the uncertainty of a sentence.

Based on the confidence of translations, we rank the input segments in an inverted order, i.e. those segments with the lowest MT confidence scores are at the top and those with higher confidence scores are at the bottom.

4.2 Geometry n-gram (Geom n-gram)

n-grams are often used as an information unit to measure the importance score of a sentence. Dara et al. (2014) used an "n-gram Overlap method" that computes the unseen score of a sentence s in U by the ratio of n-grams not seen in the training data. Particularly, n-grams that are seen fewer than V times in the training data are defined as 'unseen'. However, the "n-gram Overlap" method does not consider the information in the incremental data U. In our experiments, we utilize the "Geometry n-gram" method in Haffari et al. (2009) to calculate the sentence score as in (2):

$$\phi(s) = \sum_{n=1}^{N} \frac{\omega_n}{|X_s^n|} \sum_{x \in X_s^n} \log \frac{P(x|U,n)}{P(x|L,n)} \tag{2}$$

where $X_s^n\{n = 1, \ldots, N\}$ denotes n-grams in the sentence s, and $P(x|U,n)$ and $P(x|L,n)$ are the probability of x occurring in the set of n-grams in U and L, respectively, which can be computed via maximum likelihood estimation. ω_n is the weight that adjusts the importance of the scores of n-grams with different lengths. The weights for ω_n are same as in (Haffari et al., 2009).

From the equation, we can see that "Geom n-gram" takes into account the training corpus L and the untranslated corpus U at the same time.

4.3 Perplexity of Sentences (PPL)

In NLP tasks, the perplexity (PPL) is closely related to the concept of *entropy*, which reflects the degree of uncertainty of the information in a sentence: the larger the entropy, the greater the perplexity, and the more informative the sentence. Thus, we use PPL to calculate the importance score of a sentence s in U as in (3):

$$\phi(s) = 10^{-\frac{\log p(s)}{N - OOVs}} \tag{3}$$

where N is the number of words in the sentence s. In our experiments, the language model is trained by SRILM (Stolcke, 2002) using the source side of the parallel data with trigrams.

4.4 Cross Entropy Difference (CED)

This metric is proposed in Dara et al. (2014) for the sentence reranking in the incrementally retrained PE-SMT scenario. In this scenario, given the training corpus L and an incremental corpus U, language models (3-grams) are built from both, and each sentence s in U is scored according to the entropy H difference as in (4):

$$\phi(s) = H_U(s) - H_L(s) \tag{4}$$

where $H_U(s)$ is the entropy of the sentence s in U and the $H_L(s)$ is the entropy of s in L.

The higher the score given to a sentence, the more useful it is to L. That is, CED selects sentences from U that are different from L and similar to the overall corpus U.

5 Experiments

5.1 Data Settings

In order to have a full and fair study of the prioritization methods, we run our incremental retraining experiments on two open data sets, namely the Europarl[2] and DGT[3] corpora. For DGT data, we use four language pairs, namely English–German (En-De), English–Spanish (En-Es), English–French (En-Fr) and English–Polish (En-Pl), in one direction, i.e. the source language is English. For Europarl data, we use two language pairs bidirectionally, namely English–German and English–Spanish.

For each language pair, we extract 50k pairs of sentences as the parallel training data L for the initial SMT systems, and 10k pairs of sentences as the incremental data U that will be translated, (quasi-) post-edited[4] and added into the parallel training data iteratively in the retraining cycle. For the Europarl data, we use Newswire 2012 set as the development set (devset) to tune the initial SMT systems. For the DGT data, we extract 2,000 pairs of sentences as the devset to tune the initial SMT systems.

5.2 PE-SMT System Settings

The work flow of our incrementally retrained PE-SMT is shown in Figure 2.

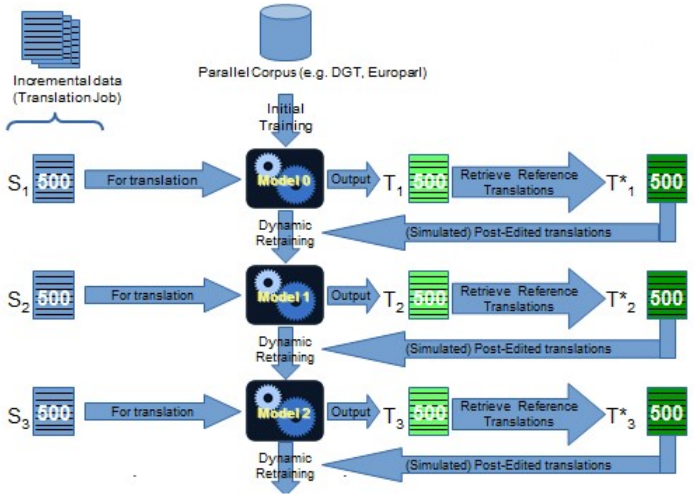
Incremental data
(Translation Job)
Parallel Corpus (e.g. DGT, Europarl)
Initial Training
S1
500
For translation
Model 0
Output
T1
500
Retrieve Reference Translations
T*1
500
Dynamic Retraining
(Simulated) Post-Edited translations
S2
500
For translation
Model 1
Output
T2
500
Retrieve Reference Translations
T*2
500
Dynamic Retraining
(Simulated) Post-Edited translations
S3
500
For translation
Model 2
Output
T3
500
Retrieve Reference Translations
T*3
500
Dynamic Retraining
(Simulated) Post-Edited translations

Figure 2: The work flow of the incrementally retrained PE-SMT in our experiments

[2] `http://www.statmt.org/wmt15/translation-task.html`

[3] `http://open-data.europa.eu/data/dataset/dgt-translation-memory`

[4] As mentioned before, we use the references of the translated segments as the PEs.

Figure 2 shows a simulated post-editing workflow in which we use the references of the translated segments instead of post-edited MT output per se. S_i indicates the batch of sentences for translation, which is determined by the segment prioritization algorithm. $Model_i$ represents the initial SMT system and the incrementally updated SMT system by the newly post-edited data. T_i indicates the MT outputs of the input segment batch S_i, and $T{*}_i$ the post-edited MT translations that return to the SMT system for dynamic retraining. This process is repeated until the incremental data set or the translation job is finished. At each retraining cycle, the incremental batch contains 500 segments that are sorted and selected from the incremental data set according to the prioritization method.

A practical incremental retrained PE-SMT system requires a quick update for its related components at each retraining cycle, such as the translation model, language model and parameter weights. In our experiments, we use Moses to build up an incrementally retrained PE-SMT system as:

- the word alignment is performed using incremental GIZA++;[5]
- the translation model is implemented by the dynamic phrase tables based on sampling word-aligned bitexts (Germann, 2014);
- the language mode is updated by appending the newly post-edited data to the training data;
- in our experiments, the weights for the features are kept unchanged. The update of parameter weights is time-consuming and is not suitable for real-time incremental retraining. From the viewpoint of system stability, the parameters can perform robustly in a limited range when the data changes. Experiments are conducted on DGT data sets to verify our assumption. The results in terms of BLEU (Papineni et al., 2002) score are shown in Table 1.

Pair	Static (%)	Incremental-Seq. (%)	Incremental-Confidence (%)
En–De	32.72	32.85	32.88
En–Es	47.29	47.18	47.14
En–Fr	44.94	44.68	44.76
En–Pl	36.15	35.91	36.04

Table 1: Robustness test of parameter weights for PE-SMT (BLEU score)

In Table 1, the numbers are BLEU scores evaluated on a constant test set (or progress set) that contains 2,000 sentence pairs. “Static” indicates that the system is built by adding all incremental data into the initial training data, tuned on the devset and tested on the progress set. “Incremental-Seq.” and “Incremental-Confidence” indicate that the parameter weights are tuned by the initial training data, and kept unchanged during the whole retraining process. The BLEU scores for these two systems are obtained at the last iteration.

We can see that the differences between the incremental systems and the static system are not significant in terms of BLEU score, which show that for the same domain data, the weights are robust in a limited data scale so that it is not necessary for them to be updated per iteration.

5.3 Statistics of Experimental Data

The statistics of entries in two data sets are shown in Table 2 and Table 3. It can be seen that we have similar and consistent distributions of entries for all language pairs and the data sets.

[5] `http://code.google.com/p/inc-giza-pp/`

Pair	Training Data	Devset	Incremental Data
En-De	43,475/85,403	7,307/10,540	17,992/31,131
En-Es	43,037/50,178	7,251/8,619	17,724/21,360
En-Fr	42,852/46,553	7,263/8,372	17,737/20,094
En-Pl	43,517/75,764	7,354/12,001	17,963/31,587

Table 2: Entries of the DGT data sets in our experiments

Pair	Training Data	Devset	Incremental Data
En-De	50,893/99,206	9,532/14,078	22,383/35,017
En-Es	53,765/75,117	100,98/12,165	21,863/29,100

Table 3: Entries of the Europarl data sets in our experiments

5.4 Prioritization Experiments

The prioritization experiments are mainly to simulate PE time by the TER score per iteration. The test set at each retraining cycle is dynamic, and contains 500 segments selected from the incremental data according to the prioritisation criteria. The average TER score of the incremental test sets for different language pairs and data sets are shown in Tables 4 and 5.

Pair	Sequential	Geom n-gram	PPL	CED	Confidence	Gains
En–De	55.94	56.50	56.58	55.23	**51.39**	*4.55*
En–Es	41.65	42.40	42.48	41.58	**38.69**	*2.96*
En–Fr	44.86	46.92	46.75	44.62	**41.42**	*3.44*
En–Pl	51.38	51.53	51.63	51.16	**48.09**	*3.29*

Table 4: Incremental results on DGT data set (TER Score)

Pair	Sequential	Geom n-gram	PPL	CED	Confidence	Gains
De–En	73.73	73.87	72.60	72.48	**70.56**	*3.17*
En–De	80.12	80.00	79.38	79.02	**77.83**	*2.29*
Es–En	84.14	84.20	83.75	83.25	**81.82**	*2.32*
En–Es	64.10	64.10	63.37	63.11	**62.38**	*1.72*

Table 5: Incremental results on Europarl data set (TER Score)

In Tables 4 and 5, the "Gains" are computed by the best result and the baseline (Sequential). We can see that the best result is obtained by the "Confidence" criterion for all tasks. The decrease in TER score for the "Confidence" criterion range from 1.72 to 4.55 absolute points (2.68$\sim$8.13 relative points) compared to the baseline.

It can also be seen that 1) the "CED" criterion beats the baseline in all tasks, and it performs better than other prioritization methods except "Confidence"; 2) the "Geom n-gram" method performs worst in all experiments; 3) the "PPL" method performs slightly better than the baseline only in the Europarl "De–En" and "En–De" tasks.

Figures 3 and 4 show the TER scores of the En–De language pair per iteration for each of the criteria in terms of the DGT and Europarl data sets.[6]

From the figures we can see that there is no obvious decrease (i.e. improvement) for the baseline in terms of TER score. However, the other four prioritization criteria have a trend of

[6] The trends are similar for the other langauge pairs.

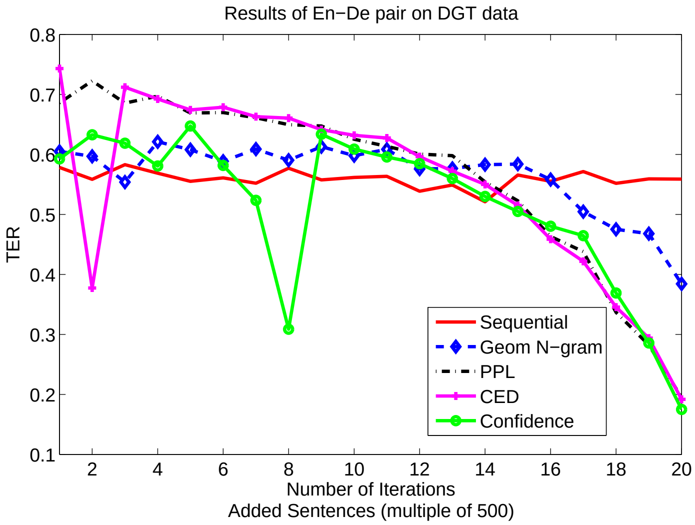

Figure 3: Results of En–De pair on DGT data

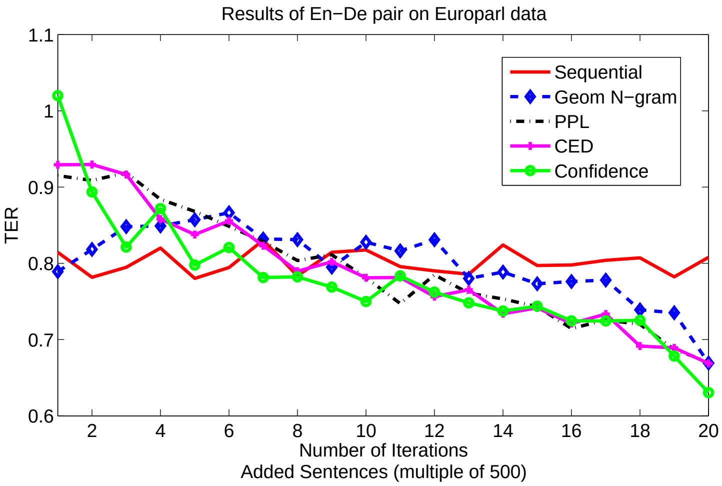

Figure 4: Results of En–De pair on Europarl data

decreasing the TER score, i.e. starting from a higher score and arriving at a lower score for the last iterations. The decreasing trends show the effectiveness of these methods to prioritize the input segments. As in Dara et al. (2014), the improvements over the baseline are shown after the initial 8-9 iterations. In our scenarios, the "Confidence" results in a noticeable decrease of the overall TER score.

In Figure 3, the "CED" and "Confidence" methods have a fluctuation at Iteration 2 and Iteration 8, respectively, but the overall trend decreases in the TER score. In Figure 4, we can see that the TER score at Iteration 1 for the "Confidence" method is over 1 which indicates the MT translation is quite poor and needs too many edits to transform it into a good output sentence.

6 Analysis

The "Confidence" method performs best in our incremental retraining experiments, which motivates us to investigate the hidden reasons by examining: the sentence length distribution and OOVs as well as the correlation between them.

6.1 Sentence Length Distribution

As in Section 5.4, we take "En–De" as an illustration of the sentence length distribution shown in Figures 5 and 6. From both figures, we find that the sentence length distribution of the "Confidence" criterion strongly fluctuates per iteration, i.e. starting from very long sentence length and arriving at very short sentence length. Furthermore, the fluctuations of the length distribution of "Confidence" consistently correspond to the TER score trend in Figures 3 and 4, i.e., when the length is short, the TER score is low and vice versa.

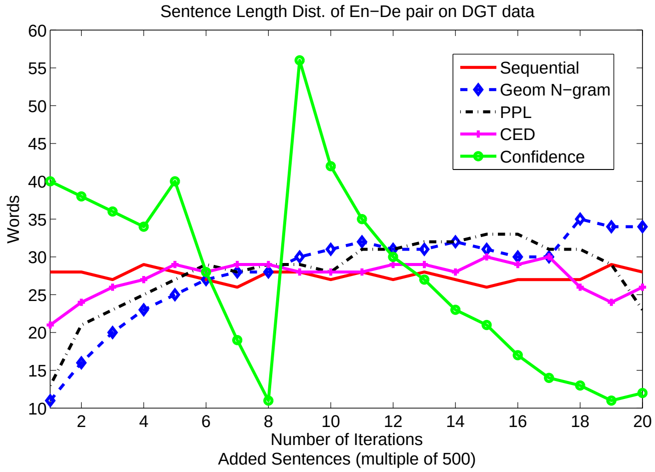

Figure 5: Sentence length distribution of En–De pair on DGT data

The length distributions of the "Geom n-gram", "PPL" and "CED" methods are more smooth than that of "Confidence". The "Geom n-gram" always starts from shorter sentences and then the length increases that indicates this method prefers to select short sentences as most informative candidates at the beginning. The distributions of the "PPL" and "CED" methods are quite similar as they both correlate with the entropy.

From the sentence length distributions, we hypothesize that the prioritization score $\phi(s)$ of "Confidence" may correlate to the sentence length of the input segment. We then calculate correlations between the score $\phi(s)$ and the sentence length by the Pearson Correlation,[7] and the results for the En-De language pair are shown in Table 6.

	Geom n-gram	PPL	CED	Confidence
DGT	0.21	0.009	0.02	**0.29**
Europarl	0.14	0.06	0.10	**0.48**

Table 6: Correlation between the prioritization score and the sentence length of input segments

In Table 6, we can see that the "Confidence" method is more correlated to the sentence

[7] http://en.wikipedia.org/wiki/Pearsoncorrelationcoefficient

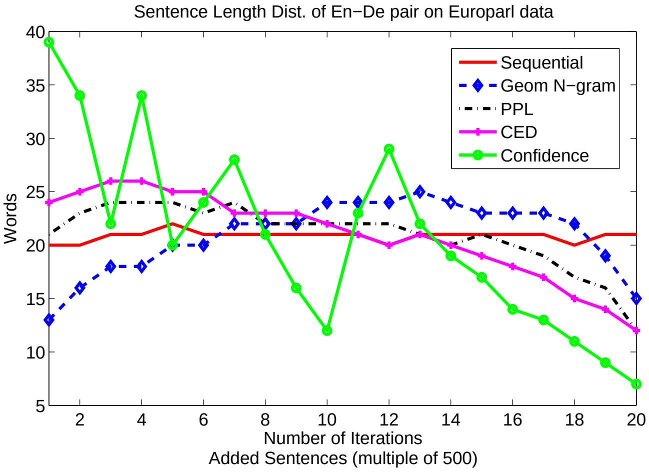

Figure 6: Sentence length distribution of En–De pair on Europarl data

length of the input segment than the other methods, which shows that the longer the sentence is, the more difficult it is to be translated.

The relationship between the score $\phi(s)$ and sentence length poses a question: should we normalize the score by the sentence length or not in the segment prioritization task? In order to answer this question and verify that the unnormalized "Confidence" method is more effective to the incremental retrained PE-SMT, we perform a further experiment using the normalized "Confidence". The results for En–DE on DGT data between these two "Confidence" methods are shown in Figure 7. In Figure 7, we can see that the trends of these two "Confidence" criteria are similar, but the normalized "Confidence" curve is more smooth. The average TER score for the normalized "Confidence" is 56.09 which is much higher (i.e. worse) than the baseline. Based on these results, we can say that the unnormalized "Confidence" method is more effective to reduce the PE time. Some other language pairs in our experiments have similar results.

6.2 OOVs

In SMT, it is known that OOVs are a big problem and significantly influence translation quality. In the Moses decoding process, when an OOV occurs, the probability $p(e|f)$ will be significantly decreased, i.e. the confidence of the translation becomes lower. Thus, the MT output score $\phi(s)$ is not only correlated with the sentence length, but is more closely correlated with the number of OOVs in the sentence.

We then calculate the correlation between the score $\phi(s)$ of "Confidence" and the OOVs by the Pearson Correlation. For En–De scenario, $\rho = 0.9993$ for the DGT data and $\rho = 0.9997$ for the Europarl data. It can be seen that as expected the more OOVs a sentence contains, the lower the confidence score, and the greater the possibility that it is ranked at the top.

6.3 Pros and Cons

The "Confidence" criterion achieved the best performance in our segment prioritization experiments for the incrementally retrained PE-SMT. However, it has some potential disadvantages that should be considered from a practical point of view:

- at each iteration, all the incremental segments need to be translated beforehand, which might be a problem for sentence-level incremental PE-SMT.

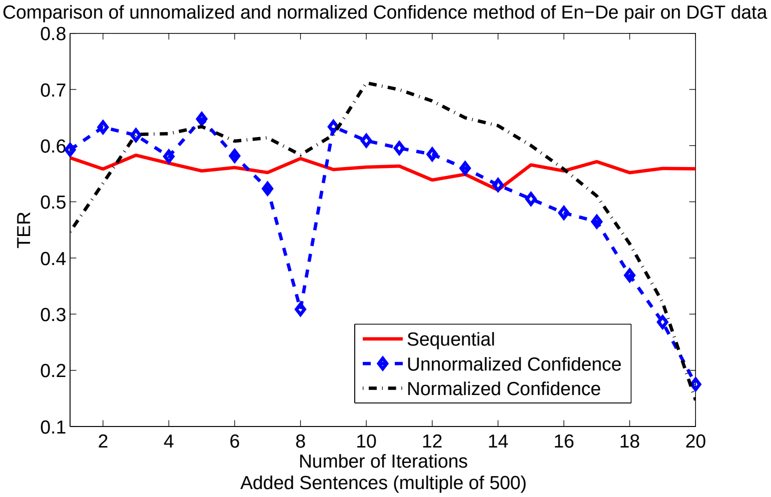

Figure 7: Comparison of unnormalized and normalized Confidence method of En–De pair on DGT data

- the sentence length at the first several iterations is much longer than that of the last several iterations, which might significantly increase the amount of post-editing which in turn may adversely affect the perception of translators/post-editors as to the utility of this approach.

However, based on the analysis in the sections above, we know that the performance of the “Confidence” is strongly correlated with the sentence length and OOVs, so we can design a new practical segment prioritization algorithm that only takes into account the training data rather than the translations according to these two crucial factors.

7 Conclusions and Future Work

In this paper, we conducted an empirical study on four different segment prioritization algorithms, namely the Sequential, Geom n-gram, PPL, CED and Confidence methods for incrementally retrained PE-SMT. Experiments conducted on two data sets and several language pairs show that the “Confidence” method achieved the best results in all tasks that reduced the TER score of 1.72-4.55 absolute points. An investigation was carried out to examine the crucial factors that make the “Confidence” effective. Finally, some suggestions are proposed for the design of new algorithms going forward.

In future work, we intend to carry out further studies on incrementally retrained PE-SMT regarding 1) the context problem: the sorted input segments lose the sequential context that is helpful to the translators; 2) developing a new algorithm which fully considers the influence of sentence length and OOVs; 3) carrying out actual PE experiments using our different segment prioritization algorithms.

Acknowledgments

We thank the reviewers for their insightful comments. We also thank our ex-colleague Dr. Sandipan Dandapat for his preliminary work on this topic. This research is supported by Science Foundation Ireland through the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Dublin City University and Trinity College Dublin, and by Grant 610879 for the Falcon project funded by the European Commission.

References

Bertoldi, N., Simianer, P., Cettolo, M., Wschle, K., Federico, M., and Riezler, S. (2014). Online adaptation to post-edits for phrase-based statistical machine translation. *Machine Translation*, 28:309–339.

Callison-Burch, C., Bannard, C., and Schroeder, J. (2005). A compact data structure for searchable translation memories. In *Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT)*, Budapest, Hungary.

Carl, M., Dragsted, B., Elming, J., Hardt, D., and Jakobsen, A. L. (2011). The process of post-editing: A pilot study. *Copenhagen Studies in Language*, 41:131–142.

Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Dara, A., van Genabith, J., Liu, Q., Judge, J., and Toral, A. (2014). Active Learning for Post-Editing Based Incrementally Retrained MT. In *Proceedings of the 14th conference of the European Chapter of the Association for Computational Linguistics*, pages 185–189, Gothenburg, Sweden.

Denkowski, M., Dyer, C., and Lavie, A. (2014). Learning from Post-Editing: Online Model Adaptation for Statistical Machine Translation. In *Proceedings of the 14th conference of the European Chapter of the Association for Computational Linguistics*, pages 395–404, Gothenburg, Sweden.

Germann, U. (2014). Dynamic phrase tables for machine translation in an interactive post-editing scenario. In *Proceedings of the workshop on interactive and adaptive machine translation*, pages 20–31, Vancouver, Canada.

Gonzalez-Rubio, J., Ortiz-Martnez, D., and Casacuberta, F. (2012). Active learning for interactive machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 245–254, Avignon, France.

Guerberof, A. (2009). Productivity and quality in mt post-editing. In *Proceedings of MT Summit XII - Workshop: Beyond Translation Memories: New Tools for Translators MT*, Ottawa, Canada.

Guerberof, A. (2013). What do professional translators think about post-editing? *Journal of Specialised Translation*, 19:75–95.

Haffari, G., Roy, M., and Sarkar, A. (2009). Active learning for statistical phrase-based machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 415–423, Colorado, USA.

Hardt, D. and Elming, J. (2010). Incremental Re-training for Post-editing SMT. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, Denver, USA.

Henríquez Q., C. A., Mariño, J. B., and Banchs, R. E. (2011). Deriving translation units using small additional corpora. In *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, pages 121–128, Leuven, Belgium.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., abd Wade Shen, B. C., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 2007*, pages 177–180, Prague, Czech Republic.

Levenberg, A., Callison-Burch, C., and Osborne, M. (2010). Stream-based translation models for statistical machine translation. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 394–402, Los Angeles, USA.

Mathur, P., Cettolo, M., and Federico, M. (2013). Online learning approaches in computer assisted translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 301–308, Sofia, Bulgaria.

OBrien, S. (2011). Towards predicting post-editing productivity. *Machine Translation*, 25(3):197–215.

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

Ortiz-Martinez, D., Garcia-Varea, I., and Casacuberta, F. (2010). Online learning for interactive statistical machine translation. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 546–554, Los Angeles, USA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 311–318, Philadelphia, USA.

Plitt, M. and Masselot, F. (2010). A productivity test of statistical machine translation post-editing in a typical localisation context. *Prague Bulletin of Mathematical Linguistics*, 93:7–16.

Simard, M. and Foster, G. (2013). Pepr: Postedit propagation using phrase-based statistical machine translation. In *Proceedings of the XIV Machine Translation Summit*, pages 191–198, Nice, France.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, USA.

Stolcke, A. (2002). SRILM – An extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, USA.

Zhang, Y. and Vogel, S. (2005). An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT)*, Budapest.

Zhechev, V. (2012). Machine Translation Infrastructure and Post-editing Performance at Autodesk. In *Proceedings of AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP2012)*, pages 87–96, San Diego, USA.