Active learning for interactive machine translation

Jesús González-Rubio and **Daniel Ortiz-Martínez** and **Francisco Casacuberta**
D. de Sistemas Informáticos y Computación
U. Politècnica de València
C. de Vera s/n, 46022 Valencia, Spain
{jegonzalez,dortiz,fcn}@dsic.upv.es

Abstract

Translation needs have greatly increased during the last years. In many situations, text to be translated constitutes an unbounded stream of data that grows continually with time. An effective approach to translate text documents is to follow an interactive-predictive paradigm in which both the system is guided by the user and the user is assisted by the system to generate error-free translations. Unfortunately, when processing such unbounded data streams even this approach requires an overwhelming amount of manpower. Is in this scenario where the use of active learning techniques is compelling. In this work, we propose different active learning techniques for interactive machine translation. Results show that for a given translation quality the use of active learning allows us to greatly reduce the human effort required to translate the sentences in the stream.

1 Introduction

Translation needs have greatly increased during the last years due to phenomena such as globalization and technologic development. For example, the European Parliament[1] translates its proceedings to 22 languages in a regular basis or Project Syndicate[2] that translates editorials into different languages. In these and many other examples, data can be viewed as an incoming unbounded stream since it grows continually with time (Levenberg et al., 2010). Manual translation of such streams of data is extremely expensive given the huge volume of translation required, therefore various automatic machine translation methods have been proposed.

However, automatic *statistical machine translation* (SMT) systems are far from generating error-free translations and their outputs usually require human post-editing in order to achieve high-quality translations. One way of taking advantage of SMT systems is to combine them with the knowledge of a human translator in the *interactive-predictive machine translation* (IMT) framework (Foster et al., 1998; Langlais and Lapalme, 2002; Barrachina et al., 2009), which is a particular case of the computer-assisted translation paradigm (Isabelle and Church, 1997). In the IMT framework, a state-of-the-art SMT model and a human translator collaborate to obtain high-quality translations while minimizing required human effort.

Unfortunately, the application of either post-editing or IMT to data streams with massive data volumes is still too expensive, simply because manual supervision of all instances requires huge amounts of manpower. For such massive data streams the need of employing *active learning* (AL) is compelling. AL techniques for IMT selectively ask an oracle (e.g. a human translator) to supervise a small portion of the incoming sentences. Sentences are selected so that SMT models estimated from them translate new sentences as accurately as possible. There are three challenges when applying AL to unbounded data streams (Zhu et al., 2010). These challenges can be instantiated to IMT as follows:

1. The pool of candidate sentences is dynamically changing, whereas existing AL algorithms are dealing with static datasets only.

[1] http://www.europarl.europa.eu
[2] http://project-syndicate.org

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 245–254,
Avignon, France, April 23 - 27 2012.

2. Concepts such as optimum translation and translation probability distribution are continually evolving whereas existing AL algorithms only deal with constant concepts.

3. Data volume is unbounded which makes impractical to batch-learn one single system from all previously translated sentences. Therefore, model training must be done in an incremental fashion.

In this work, we present a proposal of AL for IMT specifically designed to work with stream data. In short, our proposal divides the data stream into blocks where AL techniques for static datasets are applied. Additionally, we implement an incremental learning technique to efficiently train the base SMT models as new data is available.

2 Related work

A body of work has recently been proposed to apply AL techniques to SMT (Haffari et al., 2009; Ambati et al., 2010; Bloodgood and Callison-Burch, 2010). The aim of these works is to build one single optimal SMT model from manually translated data extracted from static datasets. None of them fit in the setting of data streams.

Some of the above described challenges of AL from unbounded streams have been previously addressed in the MT literature. In order to deal with the evolutionary nature of the problem, Nepveu et al. (2004) propose an IMT system with dynamic adaptation via cache-based model extensions for language and translation models. Pursuing the same goal for SMT, Levenberg et al., (2010) study how to bound the space when processing (potentially) unbounded streams of parallel data and propose a method to incrementally retrain SMT models. Another method to efficiently retrain a SMT model with new data was presented in (Ortiz-Martínez et al., 2010). In this work, the authors describe an application of the online learning paradigm to the IMT framework.

To the best of our knowledge, the only previous work on AL for IMT is (González-Rubio et al., 2011). There, the authors present a naïve application of the AL paradigm for IMT that do not take into account the dynamic change in probability distribution of the stream. Nevertheless, results show that even that simple AL framework halves the required human effort to obtain a certain translation quality.

In this work, the AL framework presented in (González-Rubio et al., 2011) is extended in an effort to address all the above described challenges. In short, we propose an AL framework for IMT that splits the data stream into blocks. This approach allows us to have more context to model the changing probability distribution of the stream (challenge 2) and results in a more accurate sampling of the changing pool of sentences (challenge 1). In contrast to the proposal described in (González-Rubio et al., 2011), we define sentence sampling strategies whose underlying models can be updated with the newly available data. This way, the sentences to be supervised by the user are chosen taking into account previously supervised sentences. To efficiently retrain the underlying SMT models of the IMT system (challenge 3), we follow the online learning technique described in (Ortiz-Martínez et al., 2010). Finally, we integrate all these elements to define an AL framework for IMT with an objective of obtaining an optimum balance between translation quality and human user effort.

3 Interactive machine translation

IMT can be seen as an evolution of the SMT framework. Given a sentence $\mathbf{f}$ from a source language to be translated into a sentence $\mathbf{e}$ of a target language, the fundamental equation of SMT (Brown et al., 1993) is defined as follows:

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

where $Pr(\mathbf{e} \mid \mathbf{f})$ is usually approximated by a log linear translation model (Koehn et al., 2003). In this case, the decision rule is given by the expression:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f}) \right\} \qquad (2)$$

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

In the IMT framework, a human translator is introduced in the translation process to collaborate with an SMT model. For a given source sentence, the SMT model fully automatically generates an initial translation. The human user checks this translation, from left to right, correcting the first

source ($\mathbf{f}$): Para ver la lista de recursos
desired translation ($\hat{\mathbf{e}}$): To view a listing of resources

inter.-0	$\mathbf{e}_\mathrm{p}$	
	$\mathbf{e}_\mathrm{s}$	*To view the resources list*
inter.-1	$\mathbf{e}_\mathrm{p}$	To view
	k	**a**
	$\mathbf{e}_\mathrm{s}$	*list of resources*
inter.-2	$\mathbf{e}_\mathrm{p}$	To view a list
	k	**i**
	$\mathbf{e}_\mathrm{s}$	*ng resources*
inter.-3	$\mathbf{e}_\mathrm{p}$	To view a listing
	k	**o**
	$\mathbf{e}_\mathrm{s}$	*f resources*
accept	$\mathbf{e}_\mathrm{p}$	To view a listing of resources

Figure 1: IMT session to translate a Spanish sentence into English. The desired translation is the translation the human user have in mind. At interaction-0, the system suggests a translation ($\mathbf{e}_\mathrm{s}$). At interaction-1, the user moves the mouse to accept the first eight characters "To view " and presses the **a** key (k), then the system suggests completing the sentence with "*list of resources*" (a new $\mathbf{e}_\mathrm{s}$). Interactions 2 and 3 are similar. In the final interaction, the user accepts the current translation.

error. Then, the SMT model proposes a new extension taking the correct prefix, $\mathbf{e}_\mathrm{p}$, into account. These steps are repeated until the user accepts the translation. Figure 1 illustrates a typical IMT session. In the resulting decision rule, we have to find an extension $\mathbf{e}_\mathrm{s}$ for a given prefix $\mathbf{e}_\mathrm{p}$. To do this we reformulate equation (1) as follows, where the term $Pr(\mathbf{e}_\mathrm{p} \,|\, \mathbf{f})$ has been dropped since it does not depend on $\mathbf{e}_\mathrm{s}$:

$$\hat{\mathbf{e}}_\mathrm{s} = \arg\max_{\mathbf{e}_\mathrm{s}} Pr(\mathbf{e}_\mathrm{p}, \mathbf{e}_\mathrm{s} \,|\, \mathbf{f}) \quad (3)$$

$$\approx \arg\max_{\mathbf{e}_\mathrm{s}} p(\mathbf{e}_\mathrm{s} \,|\, \mathbf{f}, \mathbf{e}_\mathrm{p}) \quad (4)$$

The search is restricted to those sentences $\mathbf{e}$ which contain $\mathbf{e}_\mathrm{p}$ as prefix. Since $\mathbf{e} \equiv \mathbf{e}_\mathrm{p}\,\mathbf{e}_\mathrm{s}$, we can use the same log-linear SMT model, equation (2), whenever the search procedures are adequately modified (Barrachina et al., 2009).

4 Active learning for IMT

The aim of the IMT framework is to obtain high-quality translations while minimizing the required human effort. Despite the fact that IMT may reduce the required effort with respect to post-editing, it still requires the user to supervise all the translations. To address this problem, we propose to use AL techniques to select only a small number of sentences whose translations are worth to be supervised by the human expert.

This approach implies a modification of the user-machine interaction protocol. For a given source sentence, the SMT model generates an initial translation. Then, if this initial translation is classified as incorrect or "worth of supervision", we perform a conventional IMT procedure as in Figure 1. If not, we directly return the initial automatic translation and no effort is required from the user. At the end of the process, we use the new sentence pair $(\mathbf{f}, \mathbf{e})$ available to refine the SMT models used by the IMT system.

In this scenario, the user only checks a small number of sentences, thus, final translations are not error-free as in conventional IMT. However, results in previous works (González-Rubio et al., 2011) show that this approach yields important reduction in human effort. Moreover, depending on the definition of the sampling strategy, we can modify the ratio of sentences that are interactively translated to adapt our system to the requirements of a specific translation task. For example, if the main priority is to minimize human effort, our system can be configured to translate all the sentences without user intervention.

Algorithm 1 describes the basic algorithm to implement AL for IMT. The algorithm receives as input an initial SMT model, M, a sampling strategy, S, a stream of source sentences, F, and the block size, B. First, a block of B sentences, X, is extracted from the data stream (line 3). From this block, we sample those sentences, Y, that are worth to be supervised by the human expert (line 4). For each of the sentences in X, the current SMT model generates an initial translation, $\hat{\mathbf{e}}$, (line 6). If the sentence has been sampled as worthy of supervision, $\mathbf{f} \in Y$, the user is required to interactively translate it (lines 8–13) as exemplified in Figure 1. The source sentence $\mathbf{f}$ and its human-supervised translation, $\mathbf{e}$, are then used to retrain the SMT model (line 14). Otherwise, we directly output the automatic translation $\hat{\mathbf{e}}$ as our final translation (line 17).

Most of the functions in the algorithm denote different steps in the interaction between the human user and the machine:

- `translate(`$M, \mathbf{f}$`)`: returns the most probable automatic translation of $\mathbf{f}$ given by M.
- `validPrefix(`$\mathbf{e}$`)`: returns the prefix of $\mathbf{e}$

```
input  : M (initial SMT model)
         S (sampling strategy)
         F (stream of source sentences)
         B (block size)
auxiliar : X (block of sentences)
         Y (sentences worth of supervision)
1  begin
2    repeat
3      X = getSentsFromStream(B, F);
4      Y = S(X, M);
5      foreach f ∈ X do
6        ê = translate(M, f);
7        if f ∈ Y then
8          e = ê;
9          repeat
10           e_p = validPrefix(e);
11           ê_s = genSuffix(M, f, e_p);
12           e = e_p ê_s;
13         until validTranslation(e);
14         M = retrain(M, (f, e));
15         output(e);
16       else
17         output(ê);
18   until True ;
19 end
```

Algorithm 1: Pseudo-code of the proposed algorithm to implement AL for IMT from unbounded data streams.

validated by the user as correct. This prefix includes the correction k.

- `genSuffix`$(M, \mathbf{f}, \mathbf{e}_\mathrm{p})$: returns the suffix of maximum probability that extends prefix $\mathbf{e}_\mathrm{p}$.
- `validTranslation`$(\mathbf{e})$: returns *True* if the user considers the current translation to be correct and *False* otherwise.

Apart from these, the two elements that define the performance of our algorithm are the sampling strategy $\mathrm{S}(X, M)$ and the `retrain`$(M, (\mathbf{f}, \mathbf{e}))$ function. On the one hand, the sampling strategy decides which sentences should be supervised by the user, which defines the human effort required by the algorithm. Section 5 describes our implementation of the sentence sampling to deal with the dynamic nature of data streams. On the other hand, the `retrain`$(\cdot)$ function incrementally trains the SMT model with each new training pair $(\mathbf{f}, \mathbf{e})$. Section 6 describes the implementation of this function.

5 Sentence sampling strategies

A good sentence sampling strategy must be able to select those sentences that along with their correct translations improve most the performance of the SMT model. To do that, the sampling strategy have to correctly discriminate "informative" sentences from those that are not. We can make different approximations to measure the informativeness of a given sentence. In the following sections, we describe the three different sampling strategies tested in our experimentation.

5.1 Random sampling

Arguably, the simplest sampling approach is random sampling, where the sentences are randomly selected to be interactively translated. Although simple, it turns out that random sampling perform surprisingly well in practice. The success of random sampling stem from the fact that in data stream environments the translation probability distributions may vary significantly through time. While general AL algorithms ask the user to translate informative sentences, they may significantly change probability distributions by favoring certain translations, consequently, the previously human-translated sentences may no longer reveal the genuine translation distribution in the current point of the data stream (Zhu et al., 2007). This problem is less severe for static data where the candidate pool is fixed and AL algorithms are able to survey all instances. Random sampling avoids this problem by randomly selecting sentences for human supervision. As a result, it always selects those sentences with the most similar distribution to the current sentence distribution in the data stream.

5.2 n-gram coverage sampling

One technique to measure the informativeness of a sentence is to directly measure the amount of new information that it will add to the SMT model. This sampling strategy considers that sentences with rare n-grams are more informative. The intuition for this approach is that rare n-grams need to be seen several times in order to accurately estimate their probability.

To do that, we store the counts for each n-gram present in the sentences used to train the SMT model. We assume that an n-gram is accurately represented when it appears A or more times in

the training samples. Therefore, the score for a given sentence $\mathbf{f}$ is computed as:

$$C(\mathbf{f}) = \frac{\sum_{n=1}^{N} |\mathcal{N}_n^{<A}(\mathbf{f})|}{\sum_{n=1}^{N} |\mathcal{N}_n(\mathbf{f})|} \quad (5)$$

where $\mathcal{N}_n(\mathbf{f})$ is the set of n-grams of size n in $\mathbf{f}$, $\mathcal{N}_n^{<A}(\mathbf{f})$ is the set of n-grams of size n in $\mathbf{f}$ that are inaccurately represented in the training data and N is the maximum n-gram order. In the experimentation, we assume $N = 4$ as the maximum n-gram order and a value of 10 for the threshold A. This sampling strategy works by selecting a given percentage of the highest scoring sentences.

We update the counts of the n-grams seen by the SMT model with each new sentence pair. Hence, the sampling strategy is always up-to-date with the last training data.

5.3 Dynamic confidence sampling

Another technique is to consider that the most informative sentence is the one the current SMT model translates worst. The intuition behind this approach is that an SMT model can not generate good translations unless it has enough information to translate the sentence.

The usual approach to compute the quality of a translation hypothesis is to compare it to a reference translation, but, in this case, it is not a valid option since reference translations are not available. Hence, we use confidence estimation (Gandrabur and Foster, 2003; Blatz et al., 2004; Ueffing and Ney, 2007) to estimate the probability of correctness of the translations. Specifically, we estimate the quality of a translation from the confidence scores of their individual words.

The confidence score of a word e_i of the translation $\mathbf{e} = e_1 \dots e_i \dots e_I$ generated from the source sentence $\mathbf{f} = f_1 \dots f_j \dots f_J$ is computed as described in (Ueffing and Ney, 2005):

$$C_w(e_i, \mathbf{f}) = \max_{0 \leq j \leq |\mathbf{f}|} p(e_i|f_j) \quad (6)$$

where $p(e_i|f_j)$ is an IBM model 1 (Brown et al., 1993) bilingual lexicon probability and f_0 is the empty source word. The confidence score for the full translation $\mathbf{e}$ is computed as the ratio of its words classified as correct by the word confidence measure. Therefore, we define the confidence-based informativeness score as:

$$C(\mathbf{e}, \mathbf{f}) = 1 - \frac{|\{e_i \mid C_w(e_i, \mathbf{f}) > \tau_w\}|}{|\mathbf{e}|} \quad (7)$$

Finally, this sampling strategy works by selecting a given percentage of the highest scoring sentences.

We dynamically update the confidence sampler each time a new sentence pair is added to the SMT model. The incremental version of the EM algorithm (Neal and Hinton, 1999) is used to incrementally train the IBM model 1.

6 Retraining of the SMT model

To retrain the SMT model, we implement the online learning techniques proposed in (Ortiz-Martínez et al., 2010). In that work, a state-of-the-art log-linear model (Och and Ney, 2002) and a set of techniques to incrementally train this model were defined. The log-linear model is composed of a set of feature functions governing different aspects of the translation process, including a language model, a source sentence–length model, inverse and direct translation models, a target phrase–length model, a source phrase–length model and a distortion model.

The incremental learning algorithm allows us to process each new training sample in constant time (i.e. the computational complexity of training a new sample does not depend on the number of previously seen training samples). To do that, a set of sufficient statistics is maintained for each feature function. If the estimation of the feature function does not require the use of the well-known expectation–maximization (EM) algorithm (Dempster et al., 1977) (e.g. n-gram language models), then it is generally easy to incrementally extend 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 such models, the incremental version of the EM algorithm (Neal and Hinton, 1999) is applied. A detailed description of the update algorithm for each of the models in the log-linear combination is presented in (Ortiz-Martínez et al., 2010).

7 Experiments

We carried out experiments to assess the performance of the proposed AL implementation for IMT. In each experiments, we started with an initial SMT model that is incrementally updated

corpus	use	sentences	words (Spa/Eng)
Europarl	train	731K	15M/15M
	devel.	2K	60K/58K
News Commentary	test	51K	1.5M/1.2M

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

with the sentences selected by the current sampling strategy. Due to the unavailability of public benchmark data streams, we selected a relatively large corpus and treated it as a data stream for AL. To simulate the interaction with the user, we used the reference translations in the data stream corpus as the translation the human user would like to obtain. Since each experiment is carried out under the same conditions, if one sampling strategy outperforms its peers, then we can safely conclude that this is because the sentences selected to be translated are more informative.

7.1 Training corpus and data stream

The training data comes from the Europarl corpus as distributed for the shared task in the NAACL 2006 workshop on statistical machine translation (Koehn and Monz, 2006). We used this data to estimate the initial log-linear model used by our IMT system (see Section 6). The weights of the different feature functions were tuned by means of minimum error–rate training (Och, 2003) executed on the Europarl development corpus. Once the SMT model was trained, we use the News Commentary corpus (Callison-Burch et al., 2007) to simulate the data stream. The size of these corpora is shown in Table 1. The reasons to choose the News Commentary corpus to carry out our experiments are threefold: first, its size is large enough to simulate a data stream and test our AL techniques in the long term; second, it is out-of-domain data which allows us to simulate a real-world situation that may occur in a translation company, and, finally, it consists in editorials from eclectic domain: general politics, economics and science, which effectively represents the variations in the sentence distributions of the simulated data stream.

7.2 Assessment criteria

We want to measure both the quality of the generated translations and the human effort required to obtain them.

We measure translation quality with the well-known BLEU (Papineni et al., 2002) score.

To estimate human user effort, we simulate the actions taken by a human user in its interaction with the IMT system. The first translation hypothesis for each given source sentence is compared with a single reference translation and the longest common character prefix (LCP) is obtained. The first non-matching character is replaced by the corresponding reference character and then a new translation hypothesis is produced (see Figure 1). This process is iterated until a full match with the reference is obtained. Each computation of the LCP would correspond to the user looking for the next error and *moving the pointer* to the corresponding position of the translation hypothesis. Each character replacement, on the other hand, would correspond to a *keystroke* of the user.

Bearing this in mind, we measure the user effort by means of the keystroke and mouse-action ratio (KSMR) (Barrachina et al., 2009). This measure has been extensively used to report results in the IMT literature. KSMR is calculated as the number of keystrokes plus the number of mouse movements divided by the total number of reference characters. From a user point of view the two types of actions are different and require different types of effort (Macklovitch, 2006). In any case, as an approximation, KSMR assumes that both actions require a similar effort.

7.3 Experimental results

In this section, we report results for three different experiments. First, we studied the performance of the sampling strategies when dealing with the sampling bias problem. In the second experiment, we carried out a typical AL experiment measuring the performance of the sampling strategies as a function of the percentage of the corpus used to retrain the SMT model. Finally, we tested our AL implementation for IMT in order to study the tradeoff between required human effort and final translation quality.

7.3.1 Dealing with the sampling bias

In this experiment, we want to study the performance of the different sampling strategies when

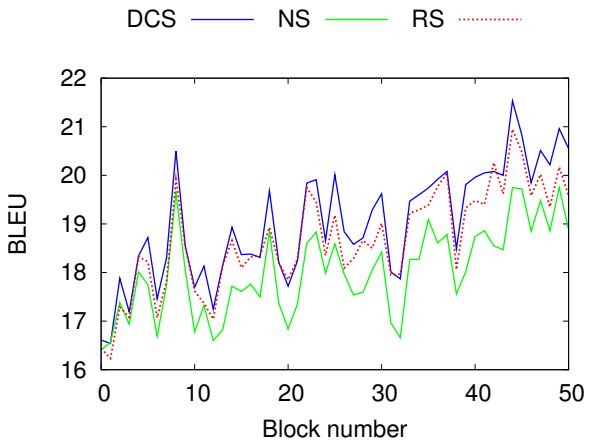

Figure 2: Performance of the AL methods across different data blocks. Block size 500. Human supervision 10% of the corpus.

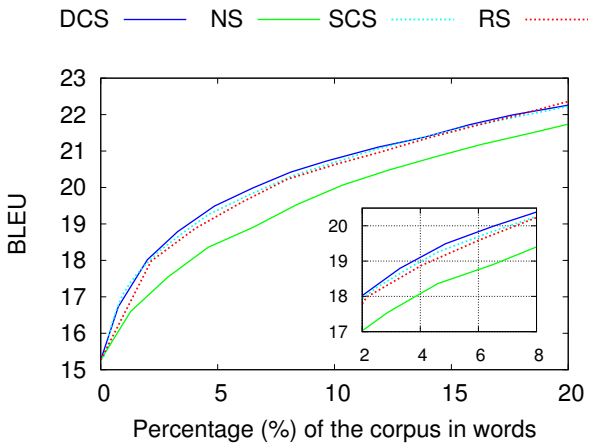

Figure 3: BLEU of the initial automatic translations as a function of the percentage of the corpus used to retrain the model.

dealing with the sampling bias problem. Figure 2 shows the evolution of the translation quality, in terms of BLEU, across different data blocks for the three sampling strategies described in section 5, namely, dynamic confidence sampling (DCS), n-gram coverage sampling (NS) and random sampling (RS). On the one hand, the x-axis represents the data blocks number in their temporal order. On the other hand, the y-axis represents the BLEU score when automatically translating a block. Such translation is obtained by the SMT model trained with translations supervised by the user up to that point of the data stream. To fairly compare the different methods, we fixed the percentage of words supervised by the human user (10%). In addition to this, we used a block size of 500 sentences. Similar results were obtained for other block sizes.

Results in Figure 2 indicate that the performances for the data blocks fluctuate and fluctuations are quite significant. This phenomenon is due to the eclectic domain of the sentences in the data stream. Additionally, the steady increase in performance is caused by the increasing amount of data used to retrain the SMT model.

Regarding the results for the different sampling strategies, DCS consistently outperformed RS and NS. This observation asserts that for concept drifting data streams with constant changing translation distributions, DCS can adaptively ask the user to translate sentences to build a superior SMT model. On the other hand, NS obtains worse results that RS. This result can be explained by the fact that NS is independent of the target language and just looks into the source language, while DCS takes into account both the source sentence and its automatic translation. Similar phenomena has been reported in a previous work on AL for SMT (Haffari et al., 2009).

7.3.2 AL performance

We carried out experiments to study the performance of the different sampling strategies. To this end, we compare the quality of the initial automatic translations generated in our AL implementation for IMT (line 6 in Algorithm 1). Figure 3 shows the BLEU score of these initial translations represented as a function of the percentage of the corpus used to retrain the SMT model. The percentage of the corpus is measured in number of running words.

In Figure 3, we present results for the three sampling strategies described in section 5. Additionally, we also compare our techniques with the AL technique for IMT proposed in (González-Rubio et al., 2011). Such technique is similar to DCS but it does not update the IBM model 1 used by the confidence sampler with the newly available human-translated sentences. This technique is referred to as static confidence sampler (SCS).

Results in Figure 3 indicate that the performance of the retrained SMT models increased as more data was incorporated. Regarding the sampling strategies, DCS improved the results obtained by the other sampling strategies. NS obtained by far the worst results, which confirms the results shown in the previous experiment. Finally,

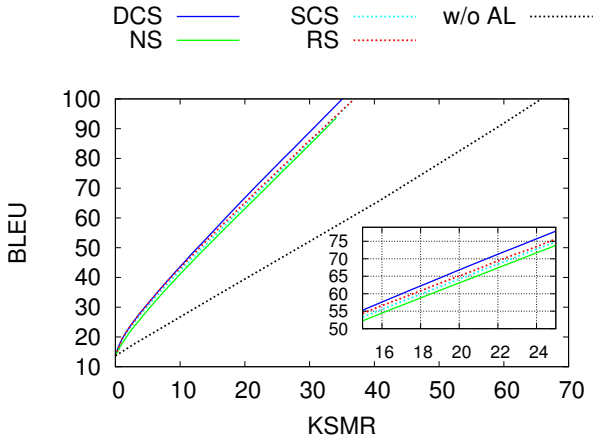

Figure 4: Quality of the data stream translation (BLEU) as a function of the required human effort (KSMR). w/o AL denotes a system with no retraining.

as it can be seen, SCS obtained slightly worst results than DCS showing the importance of dynamically adapting the underlying model used by the sampling strategy.

7.3.3 Balancing human effort and translation quality

Finally, we studied the balance between required human effort and final translation error. This can be useful in a real-world scenario where a translation company is hired to translate a stream of sentences. Under these circumstances, it would be important to be able to predict the effort required from the human translators to obtain a certain translation quality.

The experiment simulate this situation using our proposed IMT system with AL to translate the stream of sentences. To have a broad view of the behavior of our system, we repeated this translation process multiple times requiring an increasing human effort each time. Experiments range from a fully-automatic translation system with no need of human intervention to a system where the human is required to supervise all the sentences. Figure 4 presents results for SCS (see section 7.3.2) and the sentence selection strategies presented in section 5. In addition, we also present results for a static system without AL (w/o AL). This system is equal to SCS but it do not perform any SMT retraining.

Results in Figure 4 show a consistent reduction in required user effort when using AL. For a given human effort the use of AL methods allowed to obtain twice the translation quality. Regarding the different AL sampling strategies, DCS obtains the better results but differences with other methods are slight.

Varying the sentence classifier, we can achieve a balance between final translation quality and required human effort. This feature allows us to adapt the system to suit the requirements of the particular translation task or to the available economic or human resources. For example, if a translation quality of 60 BLEU points is satisfactory, then the human translators would need to modify only a 20% of the characters of the automatically generated translations.

Finally, it should be noted that our IMT systems with AL are able to generate new suffixes and retrain with new sentence pairs in tenths of a second. Thus, it can be applied in real time scenarios.

8 Conclusions and future work

In this work, we have presented an AL framework for IMT specially designed to process data streams with massive volumes of data. Our proposal splits the data stream in blocks of sentences of a certain size and applies AL techniques individually for each block. For this purpose, we implemented different sampling strategies that measure the informativeness of a sentence according to different criteria.

To evaluate the performance of our proposed sampling strategies, we carried out experiments comparing them with random sampling and the only previously proposed AL technique for IMT described in (González-Rubio et al., 2011). According to the results, one of the proposed sampling strategies, specifically the dynamic confidence sampling strategy, consistently outperformed all the other strategies.

The results in the experimentation show that the use of AL techniques allows us to make a tradeoff between required human effort and final translation quality. In other words, we can adapt our system to meet the translation quality requirements of the translation task or the available human resources.

As future work, we plan to investigate on more sophisticated sampling strategies such as those based in information density or query-by-committee. Additionally, we will conduct experiments with real users to confirm the results obtained by our user simulation.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 287576. Work also supported by the EC (FEDER/FSE) and the Spanish MEC under the MIPRCV Consolider Ingenio 2010 program (CSD2007-00018) and iTrans2 (TIN2009-14511) project and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/01).

References

Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. 2010. Active learning and crowd-sourcing for machine translation. In *Proc. of the conference on International Language Resources and Evaluation*, pages 2169–2174.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35:3–28.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proc. of the international conference on Computational Linguistics*, pages 315–321.

Michael Bloodgood and Chris Callison-Burch. 2010. Bucking the trend: large-scale cost-focused active learning for statistical machine translation. In *Proc. of the Association for Computational Linguistics*, pages 854–864.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proc. of the Workshop on Statistical Machine Translation*, pages 136–158.

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society.*, 39(1):1–38.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1998. Target-text mediated interactive machine translation. *Machine Translation*, 12:175–194.

Simona Gandrabur and George Foster. 2003. Confidence estimation for text prediction. In *Proc. of the Conference on Computational Natural Language Learning*, pages 315–321.

Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco casacuberta. 2011. An active learning scenario for interactive machine translation. In *Proc. of the 13thInternational Conference on Multimodal Interaction*. ACM.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Proc. of the North American Chapter of the Association for Computational Linguistics*, pages 415–423.

Pierre Isabelle and Kenneth Ward Church. 1997. Special issue on new tools for human translators. *Machine Translation*, 12(1-2):1–2.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proc. of the Workshop on Statistical Machine Translation*, pages 102–121.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 48–54.

Philippe Langlais and Guy Lapalme. 2002. Trans Type: development-evaluation cycles to boost translator's productivity. *Machine Translation*, 17:77–98.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Proc. of the North American Chapter of the Association for Computational Linguistics*, pages 394–402, Los Angeles, California, June.

Elliott Macklovitch. 2006. TransType2: the last word. In *Proc. of the conference on International Language Resources and Evaluation*, pages 167–17.

Radford Neal and Geoffrey Hinton. 1999. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, pages 355–368.

Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George Foster. 2004. Adaptive language and translation models for interactive machine translation. In *Proc, of EMNLP*, pages 190–197, Barcelona, Spain, July.

Franz Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the Association for Computational Linguistics*, pages 295–302.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, pages 160–167.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *Proc. of the North American Chapter of the Association for Computational Linguistics*, pages 546–554.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the Association for Computational Linguistics*, pages 311–318.

Nicola Ueffing and Hermann Ney. 2005. Application of word-level confidence measures in interactive statistical machine translation. In *Proc. of the European Association for Machine Translation conference*, pages 262–270.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33:9–40.

Xingquan Zhu, Peng Zhang, Xiaodong Lin, and Yong Shi. 2007. Active learning from data streams. In *Proc. of the 7th IEEE International Conference on Data Mining*, pages 757–762. IEEE Computer Society.

Xingquan Zhu, Peng Zhang, Xiaodong Lin, and Yong Shi. 2010. Active learning from stream data using optimal weight classifier ensemble. *Transactions on Systems, Man and Cybernetics Part B*, 40:1607–1621, December.