Pattern Recognition Letters 37 (2014) 124–134

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Cost-sensitive active learning for computer-assisted translation

Jesús González-Rubio [a,*], Francisco Casacuberta [b]

[a] Institut Tecnológic d'Informatica, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain
[b] D. Sistemes Informàtics i Computació, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain

ARTICLE INFO

Article history:
Available online 19 June 2013

Keywords:
Computer-assisted translation
Interactive machine translation
Active learning
Online learning

ABSTRACT

Machine translation technology is not perfect. To be successfully embedded in real-world applications, it must compensate for its imperfections by interacting intelligently with the user within a computer-assisted translation framework. The interactive–predictive paradigm, where both a statistical translation model and a human expert collaborate to generate the translation, has been shown to be an effective computer-assisted translation approach. However, the exhaustive supervision of all translations and the use of non-incremental translation models penalizes the productivity of conventional interactive–predictive systems.

We propose a cost-sensitive active learning framework for computer-assisted translation whose goal is to make the translation process as painless as possible. In contrast to conventional active learning scenarios, the proposed active learning framework is designed to minimize not only how many translations the user must supervise but also how difficult each translation is to supervise. To do that, we address the two potential drawbacks of the interactive–predictive translation paradigm. On the one hand, user effort is focused to those translations whose user supervision is considered more "informative", thus, maximizing the utility of each user interaction. On the other hand, we use a dynamic machine translation model that is continually updated with user feedback after deployment. We empirically validated each of the technical components in simulation and quantify the user effort saved. We conclude that both selective translation supervision and translation model updating lead to important user-effort reductions, and consequently to improved translation productivity.

1. Introduction

Machine translation (MT) is a fundamental technology that is emerging as a core component of natural language processing systems. A good example of multilingualism with high translation needs can be found in the European Union (EU) political institutions. According to EC (2009), the EU employs 1,750 full-time translators. Additionally, to cope with demand fluctuations, the EU uses external translation providers which generate approximately one fourth of its translation output. As a result, in 2008 the EU translation services translated more than 1,800,000 pages and spent about one billion Euros on translation and interpreting.

Besides being an expensive and time-consuming task, the problem with translation by human experts is that the demand for high-quality translation has been steadily increasing, to the point where there are just not enough qualified translators available today to satisfy it. This poses a high pressure on translation agencies that must decide how to invest their limited resources (budget, manpower, time, etc.) to generate translations of the maximum quality in the most efficient way.

To address this challenge, many translation agencies have focused their interest on MT technology. However, current state-of-the-art MT systems are still far from generating error-free translations (NIST, 2006; Lopez, 2008). Indeed, they usually require human experts to post-edit their automatic translations. This serial process prevents MT systems from taking advantage of the knowledge of the human experts, and the users cannot take advantage of the adaptive ability of MT systems.

An alternative way to utilize the existing MT technologies is to use them in collaboration with human translators within a computer-assisted translation (CAT) framework (Isabelle and Church, 1998). An important contribution to CAT technology was carried out during the TransType project (Foster et al., 1998; Langlais et al., 2000; Foster, 2002; Langlais and Lapalme, 2002). They proposed the interactive–predictive machine translation (IMT) framework where data-driven MT technologies are embedded within the translation environment. Following these ideas, Barrachina et al. (2009) proposed an innovative embedding where a fully-fledged statistical MT (SMT) system is used

* Corresponding author. Tel.: +34 96 387 70 69; fax: +34 96 387 72 39.
E-mail addresses: jegonzalez@iti.upv.es (J. González-Rubio), fcn@iti.upv.es (F. Casacuberta).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.06.007

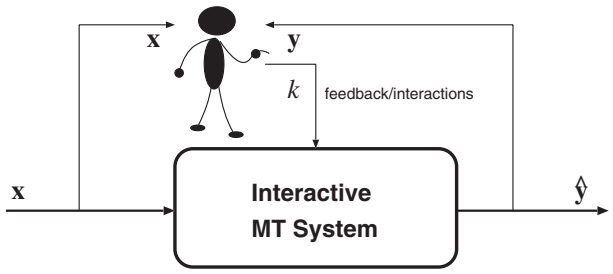

Fig. 1. Diagram of an interactive–predictive MT system. To translate a source sentence **x**, the user interacts with the system accepting or correcting the proposed translations **y**. User feedback k is used by the system to improve its suggestions.

to produce complete translations, or portions thereof, which can be accepted or amended by a human expert, see Fig. 1. Each corrected text segment is then used by the SMT system as additional information to achieve further, hopefully improved, translations.

Despite being an efficient CAT protocol, conventional IMT technology has two potential drawbacks. First, the user is required to supervise all the translations. Each translation supervision involves the user reading and understanding the proposed target language sentence, and deciding if it is an adequate translation of the source sentence. Even in the case of error-free translations, this process involves a non-negligible cognitive load. Second, conventional IMT systems consider static SMT models. This implies that after being corrected the system may repeat its errors, and the user will be justifiably disappointed.

We propose a cost-sensitive active learning (AL) (Angluin, 1988; Atlas et al., 1990; Cohn et al., 1994; Lewis and Gale, 1994) framework for CAT where the IMT user-machine interaction protocol (Fig. 2) is used to efficiently supervise automatic translations. Our goal is to make the translation process as efficient as possible. That is, we want to maximize the translation quality obtained per unit of user supervision effort. Note that this goal differs from the goal of traditional AL scenarios. While they minimize the number of manually-translated sentences to obtain a robust MT system, we aim at minimizing the number of corrective actions required to generate translations of a certain quality.

The proposed cost-sensitive AL framework boosts the productivity of IMT technology by addressing its two potential drawbacks. First, we do not require the user to exhaustively supervise all translations. Instead, we propose a selective interaction protocol where the user only supervises a subset of "informative" translations (González-Rubio et al., 2010). Additionally, we test several criteria to measure this "informativeness". Second, we replace the batch SMT model by an incremental SMT model (Ortiz-Martínez et al., 2010) that utilizes user feedback to continually update its parameters after deployment. The potential user effort reductions of our proposal are twofold. On the one hand, user effort is focused on those translations whose supervision is considered most "informative". Thus, we maximize the utility of each user interaction. On the other hand, the SMT model is continually updated with user feedback. Thus, the SMT model is able to learn new translations and to adapt its outputs to match the user's preferences which prevents the user from making repeatedly the same corrections.

The remainder of this article is organized as follows. First, we briefly describe the SMT approach to translation, and its application in the IMT framework (Section 2). Next, we present the proposed cost-sensitive AL framework for CAT (Section 3). Then, we show the results of experiments to evaluate our proposal (Section 4). Finally, we summarize the contributions of this article in Section 5.

source (**x**): Para ver la lista de recursos

desired translation ($\hat{\mathbf{y}}$): To view a listing of resources

interaction-0	$\mathbf{y}_p$	
	$\mathbf{y}_s$	*To view the resources list*
interaction-1	$\mathbf{y}_p$	To view
	k	**a**
	$\mathbf{y}_s$	*list of resources*
interaction-2	$\mathbf{y}_p$	To view a list
	k	**i**
	$\mathbf{y}_s$	*ng resources*
interaction-3	$\mathbf{y}_p$	To view a listing
	k	**o**
	$\mathbf{y}_s$	*f resources*
accept	$\mathbf{y}_p$	To view a listing of resources

Fig. 2. IMT session to translate a Spanish sentence into English. At interaction-0, the system suggests a translation ($\mathbf{y}_s$). At interaction-1, the user moves the mouse just before the first error and implicitly validates the first eight characters "To view " as a correct prefix ($\mathbf{y}_p$). Then, the user introduces a correction by pressing the **a** key (k). Lastly, the system suggests completing the translation from the user correction with "*list of resources*" (a new $\mathbf{y}_s$). At interaction 2, the user validates "*To view a list*" and introduces a correction **i** which is completed by the systems to form a new translation "*To view a listing of resources*". Interaction 3 is similar. Finally, the user accepts the current translation which is equal to the desired translation.

2. Interactive–predictive machine translation

The statistical machine translation (SMT) approach considers translation as a decision problem, where it is necessary to decide upon a translation $\mathbf{y}$ given a source language sentence $\mathbf{x}$. Statistical decision theory is used to select the correct translation among all the target language sentences. From the set of all possible target language sentences, we are interested in that with the highest probability according to the following equation (Brown et al., 1993)[1]:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} Pr(\mathbf{y}|\mathbf{x}) \tag{2.1}$$

where $Pr(\mathbf{y}|\mathbf{x})$ is usually modeled by a maximum entropy MT model (Och and Ney, 2002), also known as log-linear model. The decision rule for log-linear models is given by the expression:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} Pr(\mathbf{y}|\mathbf{x}) \approx \arg\max_{\mathbf{y}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{y}, \mathbf{x}) \right\} \tag{2.2}$$

where each $h_m(\mathbf{y}, \mathbf{x})$ is a feature function that describes a particular aspect of the translation process (e.g. the log-probability $\log(P(\mathbf{y}))$ of the translation), and λ_m is its associated weight. Phrase-based (Koehn et al., 2003) and finite state (Casacuberta and Vidal, 2007) models are two successful implementations of the log-linear approach.

However, despite a huge research effort, SMT systems are still not perfect. To obtain high-quality translations, a human expert has to supervise the automatically generated translations. This supervision is usually carried out as a separate post-edition step. The IMT framework (Barrachina et al., 2009) constitutes an alternative to this serial procedure. In an IMT system, an SMT model and a human expert collaborate to generate error-free translations. These translations are generated in a series of interactions between the SMT model and the user. At each interaction, the SMT model generates a translation of the source sentence which can be partially or completely accepted and corrected by the user. Each partially corrected text segment, called prefix, is then used by the SMT model as additional information to generate better translation suggestions. Fig. 2 shows an example of a typical IMT session.

The IMT decision rule searches for an extension $\mathbf{y}_s$ that completes a user-validated prefix $\mathbf{y}_p$ is given by:

$$\hat{\mathbf{y}}_s = \arg\max_{\mathbf{y}_s} Pr(\mathbf{y}_s|\mathbf{x}, \mathbf{y}_p) \tag{2.3}$$

which can be straightforwardly rewritten as:

$$\hat{\mathbf{y}}_s = \arg\max_{\mathbf{y}_s} Pr(\mathbf{y}_p, \mathbf{y}_s|\mathbf{x}) \tag{2.4}$$

Given that $\mathbf{y}_p\mathbf{y}_s = \mathbf{y}$, this equation is very similar to Eq. (2.1). The main difference is that the search now is performed over the set of suffixes $\mathbf{y}_s$ that complete $\mathbf{y}_p$ instead of complete sentences ($\mathbf{y}$ in Eq. (2.1)). This implies that we can use the same MT models whenever the search procedures are adequately modified (Och et al., 2003). It should be noted that SMT models are defined at word level while the IMT interface depicted in Fig. 2 works at character level. This is not an important issue since the transformations that are required in the SMT models for their use at character level are trivial.

[1] We use $Pr(\cdot)$ to denote general probability distributions and $P(\cdot)$ to denote model-based distributions.

3. Cost-sensitive active learning for computer-assisted translation

Although IMT have been successfully deployed in many practical applications, it still demands the human user to supervise all translations. This exhaustive supervision guarantees that the generated translations are error-free. However, it demands a large amount of cognitive effort by the user which penalizes translation productivity. A translation agency with limited resources, in terms of person-hours, may be willing to sacrifice some translation quality in exchange for improved productivity. Certainly, this is an unrealistic scenario in some cases, for example it is inconceivable not to fully-supervise the translation of a legal document such as a contract, but there are many other translation tasks, e.g. manuals for electronic devices, or twitter and blog postings, that match this productivity-focused scenario.

The goal of this section is to present a cost-efficient CAT framework that allows the user to supervise and correct automatic translations as effortlessly as possible. From the existing IMT technology, we import its user-machine interaction process (Fig. 2) to efficiently supervise individual translations. However, we implement a different work-flow to address its drawbacks. On the one hand, user effort will be focused to supervise only those translations considered most "informative". On the other hand, the translation model will be continually updated with the new sentence pairs $(\mathbf{x}, \mathbf{y})$ supervised by the user.

We implement these ideas as a cost-sensitive AL scenario designed to minimize supervision effort, Section 3.1. We define a new translation work-flow, Section 3.2, that focuses user-effort to only supervise the subset of most "informative" translations. Section 3.3 describes the different ranking functions implemented to measure the "informativeness" of each translation, and finally, Section 3.4 presents the incremental SMT model that is continually updated from user feedback.

3.1. Active learning for computer-assisted translation

Training an SMT model requires translation examples of source language sentences and its corresponding target language translations. Example annotation is difficult for structured prediction tasks, since each example may have multiple, interacting labels, all of which must be correctly annotated for the example to be of use to the learner. This is particularly true for translation where additionally there may be multiple correct translations for a source sentence.

Different alternatives to conventional supervised learning have been proposed to address these problems. For example, semi-supervised learning methods use unlabeled data to help supervised learning tasks (Chapelle et al., 2006). These methods typically assume that the labeled data set is given and fixed. In practice, however, semi-supervised methods are allowed to pick a set of unlabeled examples to be annotated by an expert. In this case, rather than selecting the examples randomly, it may be attractive to let the learning algorithm to proactively tell us which of them to annotate. This approach is known as active learning (AL). The idea is to select which training examples to label and the order in which they are labeled to increase learning efficiency (Angluin, 1988; Atlas et al., 1990; Cohn et al., 1994; Lewis and Gale, 1994). An active learner is considered successful if it obtains better performance than a traditional learner given the same *number* of training examples. Therefore, AL expedites annotation by reducing the number of labeled examples required to train an accurate model.

Algorithm 1. Pseudo-code of the proposed cost-sensitive AL framework for CAT. Functions `translate`$(\mathbb{M}, \mathbf{x})$, `validPrefix`$(\mathbf{y})$, `genSuffix`$(\mathbb{M}, \mathbf{x}, \mathbf{y}_p)$, and `validTranslation`$(\mathbf{y})$ (Section 3.2) denote the IMT user-machine interaction protocol, see Fig. 2. Function `sampling`$(\mathcal{B}, \rho)$ implements the strategy to sample the most "informative" sentences from $\mathcal{B}$ (Section 3.3), and function `update`$(\mathbb{M}, (\mathbf{x}, \mathbf{y}))$ returns translation model $\mathbb{M}$ updated with the new sentence pair $(\mathbf{x}, \mathbf{y})$ (Section 3.4)

input: $\mathcal{D}$ (stream of source sentences)
$\mathbb{M}$ (initial SMT model)
ρ (effort level, percentage of sentences to be supervised)
auxiliar: $\mathcal{B}$ (block of consecutive sentences)
$\mathcal{S} \subseteq \mathcal{B}$ (list of sentences to be supervised by the user)

```
1  begin
2    repeat
3      B = getBlockFromStream(D);
4      S = sampling(B, ρ));
5      for each x ∈ B do
6        ŷ = translate(M, x)
7        if x ∈ S then
8          y = ŷ;
9          repeat
10           y_p = validPrefix(y);
11           ŷ_s = genSuffix M, x, y_p;
12           y = y_p ŷ_s;
13         until validTranslation(y);
14         M = update(M, (x, y));
15         output(y);
16       else
17         output(ŷ);
18     until D ≠ ∅;
19 end
```

In contrast to previous applications of AL to structured prediction tasks, e.g. sequence labeling (Settles and Craven, 2008), natural language parsing and information extraction (Thompson et al., 1999), or machine translation (Haffari et al., 2009), that minimize the number of labeled samples required to train an accurate model, our goal is to reduce the user supervision effort required to generate high-quality translations. Clearly, the amount of work required to supervise a translation will vary between sentences, e.g. based on the size and the complexity of the source sentence. Thus, it is desirable to design an AL supervision scenario that considers not only *how many* translations the user is required to supervise, but also *how difficult* each translation is to supervise.

3.2. Translation work-flow and supervision protocol

The proposed AL framework for CAT implies a modification of the conventional IMT work-flow depicted in Fig. 1. The user no longer supervises the translation of all sentences but only of those selected as "worthy of being supervised". Since only the most informative sentences are supervised, we maximize the utility of each user interaction. Final translations however may not be error-free as for conventional IMT. In exchange, an important reduction in human effort is potentially achievable. Moreover, we can modify the ratio of sentences to be supervised by the user thus modifying the behavior of our system between an automatic SMT system, and a fully-supervised IMT system. In other words, we can adapt the system's behavior to the requirements of each particular translation task.

Conventional IMT technology is build over the implicit assumption that the inbound text to be translated behaves as a text stream (see Fig. 1). Source sentences are translated separately and no information is stored (or assumed) about the preceding (or following) sentences, e.g. how many sentences remain untranslated. Since the IMT framework uses static SMT models and requires the user to supervise all translations, this is not a strong assumption. However, we have to take it into account because information about previously supervised translations, and particularly, about following sentences may have great impact on the final user effort. We handle the inbound text stream by partitioning the data into blocks of consecutive sentences. Within a block, all sentences are available, but once the algorithm moves to the next block, all sentences in previous blocks become inaccessible. We use the sentences within a block to estimate the current distribution of sentences in the stream, so that the estimation of the "informativeness" of supervising the translation of a sentence can be done as accurately as possible.

Algorithm 1 shows the pseudo-code that implements the proposed cost-sensitive AL scenario for CAT. The algorithm takes as input a stream of source sentences $\mathcal{D}$, a "base" SMT model $\mathbb{M}$, and an effort level ρ denoting the percentage of sentences of each block to be supervised. First, the next block of sentences $\mathcal{B}$ is read from the data stream (line 3). From this block, we sample the set of sentences $\mathcal{S} \subseteq \mathcal{B}$ that are worthy of being supervised by the human expert (line 4). For each sentence in $\mathcal{B}$, the current SMT model generates an initial translation, $\hat{\mathbf{y}}$ (line 6). If the sentence has been sampled as worth of supervision, $\mathbf{x} \in \mathcal{S}$, the user collaborates with the system to translate the sentence (lines 8–13). Then, the new sentence pair $(\mathbf{x}, \mathbf{y})$ is used to update the SMT model $\mathbb{M}$ (line 14), and the human-supervised translation is returned (line 15). Otherwise, we directly return the automatic translation $\hat{\mathbf{y}}$ as the final translation (line 17). Although both automatic and user-supervised translations are available, preliminary experiments showed that using both translations to update the SMT model resulted in reduced learning rates.

Although other translation supervision methods, e.g. post-edition, can be used,[2] we implement the IMT user-machine interaction protocol (Fig. 2)) to supervise each individual translation. Functions between lines 8–13 denote this supervision procedure:

`translate`$(\mathbb{M}, \mathbf{x})$:
It returns the most probable automatic translation of $\mathbf{x}$ according to $\mathbb{M}$. If $\mathbb{M}$ is a log-linear SMT model, this function implements Eq. (2.2).

`validPrefix`$(\mathbf{y})$:
It denotes the user actions (positioning and correction of the first error) performed to amend $\mathbf{y}$. It returns the user-validated prefix $\mathbf{y}_p$ of translation $\mathbf{y}$, including the user correction k.

`genSuffix`$(\mathbb{M}, \mathbf{x}, \mathbf{y}_p)$:
It returns the suffix $\mathbf{y}_s$ of maximum probability that extends prefix $\mathbf{y}_p$. This function implements Eq. (2.4).

`validTranslation`$(\mathbf{y})$:
It denotes the user decision of whether translation $\mathbf{y}$ is a correct translation or not. It returns *True* if the user considers $\mathbf{y}$ to be correct and *False* otherwise.

[2] This will imply a modification of lines 8–13 in Algorithm 1.

In addition to the supervision procedure, the two elements that define the performance of Algorithm 1 are the sampling strategy $\texttt{sampling}(\mathcal{B}, \rho)$ and the SMT model update function $\texttt{update}(\mathbb{M}, (\mathbf{x}, \mathbf{y}))$. The sampling strategy decides which sentences of $\mathcal{B}$ are worthy of being supervised by the user. This is the main component of our framework and has a major impact on the final performance of the algorithm. Section 3.3 describes several strategies implemented to measure each sentence's "informativeness". In turn, the $\texttt{update}(\mathbb{M}, (\mathbf{x}, \mathbf{y}))$ function updates the SMT model $\mathbb{M}$ with a new training pair $(\mathbf{x}, \mathbf{y})$. Section 3.4 describes the implementation of this functionality.

3.3. Sentence sampling strategies

The goal of our AL framework for CAT is to generate high-quality translations as effortlessly as possible. Since good translations are less costly to supervise than bad ones, the aim of a sampling strategy $\texttt{sampling}(\mathcal{B}, \rho)$ should be to select those sentences $\mathcal{S} \subseteq \mathcal{B}$ for which knowing their correct translation allows to improve most the performance of the SMT model for future sentences. To do that, we first use a ranking function $\Phi(\mathbf{x})$ to score the sentences in $\mathcal{B}$. Then, the percentage ρ of the highest-scoring sentences are selected to be supervised by the user. We identify three properties that (partially) account for the "worth" of a given sentence:

Uncertainty: A sentence is as worthy as uncertain is the SMT model of how to translate it.
Representativeness: A sentence is as worthy as it is "representative" of the sentences in $\mathcal{B}$.
Unreliability: A sentence is as worthy as the amount of unreliably modeled events that it contains.

Next sections describe different sampling strategies designed to measure one (or more) of these complementary properties.

3.3.1. Random ranking (R)

Random ranking assigns a random score in the range $[0, 1]$ to each sentence. It is the baseline ranking function used in the experimentation. Although simple, random ranking performs surprisingly well in practice. Its success stems from the fact that it always selects sentences according to the underlying distribution. Using a typical AL heuristic, as training proceeds and sentences are sampled, the training set quickly diverges from the real data distribution. This difficulty known as *sampling bias* (Dasgupta and Hsu, 2008) is the fundamental characteristic that separates AL from other learning methods. However, since by definition random ranking selects sentences according to the underlying distribution, it does not suffer from sampling bias. This fact makes random ranking a very strong baseline to compare with.

3.3.2. Uncertainty ranking (U)

One of the most common AL methods is uncertainty sampling (Lewis and Gale, 1994). This method selects those samples about which the model is least certain how to label. The intuition is clear: much can be learned from the correct output if the model is uncertain of how to label the sample. Formally, a typical uncertainty sampling strategy scores each sample $\mathbf{x}$ with one minus the probability of its most probable prediction $\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$:

$$\Phi(\mathbf{x}) = 1 - P(\hat{\mathbf{y}}|\mathbf{x}) \tag{3.1}$$

However, due to the peculiarities of SMT models, uncertainty sampling has to be re-considered. Since the normalization term does not influence the decision on the highest-probability translation, it is usually ignored in the model formulation, see Eq. (2.2). As a result, instead of true probabilities these models generate simple scores that are not directly comparable between translations. Hence the conventional uncertainty technique cannot be implemented. Instead, under the assumption that the "certainty" of a model in a particular translation is correlated with the quality of that translation, we measure the uncertainty of a translation using an estimation of its quality. Specifically, we use confidence measures (Blatz et al., 2004; Ueffing and Ney, 2007) to estimate the quality of a translation from the confidence estimations of its individual words.

Given a translation $\mathbf{y} = y_1, \ldots, y_i, \ldots, y_{|\mathbf{y}|}$[3] generated from a source sentence $\mathbf{x} = x_1, \ldots, x_j, \ldots, x_{|\mathbf{x}|}$, the confidence of each target language word $C(y_i, \mathbf{x})$ is computed as described in (Ueffing and Ney, 2005):

$$C(y_i, \mathbf{x}) = \max_{0 \leqslant j \leqslant |\mathbf{x}|} P(y_i|x_j) \tag{3.2}$$

where $P(y_i|x_j)$ is a word-to-word probability model, and x_0 is the empty source word. Following Ueffing and Ney (2005), we use an SMT model 1 (Brown et al., 1993) although other bilingual lexicon models, e.g., model 2 (Brown et al., 1993), or hidden Markov model (Ueffing and Ney, 2007), could also be used.

The confidence-based uncertainty score is then computed as one minus the ratio of words in the most probable translation $\hat{\mathbf{y}} = y_1, \ldots, y_i, \ldots, y_{|\hat{\mathbf{y}}|}$ classified as incorrect according to a word-confidence threshold τ_w:

$$\Phi_U(\mathbf{x}) = 1 - \frac{|\{y_i | C(y_i, \mathbf{x}) > \tau_w\}|}{|\hat{\mathbf{y}}|} \tag{3.3}$$

In the experimentation, threshold value τ_w was tuned to minimize classification error in a separate development set. Additionally, we use the incremental version of the EM algorithm (Neal and Hinton, 1999) to update the word-to-word probability model $P(y_i|x_j)$ each time a new sentence pair is available.

3.3.3. Information density ranking (ID)

Uncertainty sampling bases its decisions on individual instances which makes the technique prone to sample outliers. The least certain sentences may not be "representative" of other sentences in the distribution, in this case, knowing its label is unlikely to improve accuracy on the data as a whole (Roy and McCallum, 2001). We can overcome this problem by modeling the input distribution explicitly when scoring a sentence.

The information density framework (Settles and Craven, 2008) is a general density-weighting technique. The main idea is that informative instances should not only be those which are uncertain, but also those which are "representative" of the underlying distribution (i.e., inhabit dense regions of the input space). To address this, we compute the information density score:

$$\Phi_{ID}(\mathbf{x}) = \Phi_U(\mathbf{x}) \cdot \left(\frac{1}{|\mathcal{B}|} \sum_{b=1}^{|\mathcal{B}|} S(\mathbf{x}, \mathbf{x}_b) \right)^{\gamma} \tag{3.4}$$

where the uncertainty of a given sentence $\mathbf{x}$ is weighted by its average similarity $S(\mathbf{x}, \cdot)$ to the rest of sentences in the distribution, subject to a parameter γ that controls the relative importance of the similarity term. Since the distribution is unknown, we use the block of sentences $\mathcal{B} = \{\mathbf{x}_1, \ldots, \mathbf{x}_b, \ldots, \mathbf{x}_{|\mathcal{B}|}\}$ to approximate it. We use uncertainty ranking $\Phi_U(\mathbf{x})$ to measure the "base" worth of a sentence, but we could use any other instance-level strategies presented in the literature (Settles and Craven, 2008; Haffari et al., 2009).

[3] We use the same symbol $|\cdot|$ to denote an absolute value $|a|$, the length of a sequence $|\mathbf{x}|$, and the cardinality of a set $|\mathcal{B}|$. The particular meaning will be clear depending on the context.

We compute the similarity of two sentences as the geometric mean of the precision of n-grams (sequences of n consecutive words in a sentence) up to size four[4] between them:

$$S(\mathbf{x},\mathbf{x}_b) = \left(\prod_{n=1}^{4} \frac{\sum_{\mathbf{w}\in\mathcal{W}_n(\mathbf{x})} \min(\#_{\mathbf{w}}(\mathbf{x}), \#_{\mathbf{w}}(\mathbf{x}_b))}{\sum_{\mathbf{w}\in\mathcal{W}_n(\mathbf{x})} \#_{\mathbf{w}}(\mathbf{x})}\right)^{\frac{1}{4}} \tag{3.5}$$

where $\mathcal{W}_n(\mathbf{x})$ is the set of n-grams of size n in $\mathbf{x}$, and $\#_{\mathbf{w}}(\mathbf{x})$ represents the count of n-gram $\mathbf{w}$ in $\mathbf{x}$. This similarity score is closely related to the widespread translation evaluation score BLEU (Papineni et al., 2002) that will be further discussed in Section 4.2.1.

One potential drawback of information density is that the number of similarity calculations grows quadratically with the number of instances in $\mathcal{B}$. However, similarities only need to be computed once for a given $\mathcal{B}$ and are independent of the base measure. Therefore, we can pre-compute and cache them for efficient look-up during the AL process.

3.3.4. Coverage augmentation ranking (CA)

Sparse data problems are ubiquitous in natural language processing (Zipf, 1935). This implies that some rare events will be missing completely from a training set, even when it is very large. Missing events result in a loss of coverage, a situation when the structure of the model is not rich enough to cover all types of input. As a result, words (or sequences thereof) that do not appear in the training set cannot be adequately translated (Turchi et al., 2009; Haddow and Koehn, 2012).

Uncertainty sampling assumes that the model structure is fixed in advance and focus upon improving parameters within that structure. However, this is not appropriate for SMT where the model structure and the associated parameters are determined from training data. The problem is that uncertainty-based methods fail at dealing with sentences with words not covered by the model. To efficiently reduce classification error in SMT, we should explicitly address unreliably trained model parameters. We do that by measuring the coverage augmentation $\Delta_{cov}(\mathbf{x},\mathcal{T})$ due to the incorporation of sentence $\mathbf{x}$ to the current training set $\mathcal{T}$:

$$\Delta_{cov}(\mathbf{x},\mathcal{T}) = \sum_{n=1}^{4} \sum_{\mathbf{w}\in(\mathcal{W}_n(\mathbf{x})-\mathcal{W}_n(\mathcal{T}))} \sum_{b=1}^{|\mathcal{B}|} \#_{\mathbf{w}}(\mathbf{x}_b) \tag{3.6}$$

The coverage augmentation for each sentence $\mathbf{x}$ is given by the count of n-grams in $\mathbf{x}$ missing in the training set $\mathcal{T}$ that appear in the rest of sentences in the block. That is, we measure how many missing n-grams in $\mathcal{B}$ would be covered if $\mathbf{x}$ is added to the training set. Again, we consider $n = 4$ as the maximum n-gram length.

This coverage augmentation score is biased towards longer sentences since longer sentences can contain a larger amount of unseen n-grams. This is one of the reasons for its successful application in conventional AL scenarios (Haffari et al., 2009) and bilingual sentence selection tasks (Gascó et al., 2012). However, longer sentences also imply a higher cognitive effort from the user (Koponen, 2012) which may penalize performance. We address this dilemma by normalizing the coverage augmentation score by an estimation of the user-effort $\mathrm{E}(\mathbf{x})$ required to supervise the translation. Since out-of-coverage words cannot be adequately translated and their translations will be corrected by the user, we assume user effort to be proportional to the number of out-coverage-words in the source sentence:

$$\mathrm{E}(\mathbf{x}) \propto \sum_{\mathbf{w}\in(\mathcal{W}_1(\mathbf{x})-\mathcal{W}_1(\mathcal{T}))} \#_{\mathbf{w}}(\mathbf{x}) \tag{3.7}$$

Finally, the coverage augmentation score measures the potential SMT model improvement per unit of user effort[5]:

$$\Phi_{CA}(\mathbf{x}) = \frac{\Delta_{cov}(\mathbf{x},\mathcal{T})}{\mathrm{E}(\mathbf{x})} \tag{3.8}$$

To avoid selecting several sentences with the same missing n-grams, we update the set of n-grams seen in training each time a new sentence is selected. First, sentences in $\mathcal{B}$ are scored using Eq. (3.8). Then, the highest-scoring sentence is selected and removed from $\mathcal{B}$. The set of training n-grams is updated with the n-grams present in the selected sentence and, hence, the scores of the rest of the sentences in the block are also updated. This process is repeated until we select the desired ratio ρ of sentences from $\mathcal{B}$.

3.4. Online training for SMT

After the translation supervision process, we have a new sentence pair $(\mathbf{x},\mathbf{y})$ at our disposal. We now briefly describe the incremental SMT model used in the experimentation, and the online learning techniques implemented to update the model with new sentence pairs in constant time.

We implement the online learning techniques proposed in Ortiz-Martínez et al. (2010). In that work, a state-of-the-art log-linear SMT model (Och and Ney, 2002) was presented. This model is composed of a set of incremental feature functions governing different aspects of the translation process, see Eq. (2.2), including a language model, a model of source sentences length, direct $P(\mathbf{y}|\mathbf{x})$ and inverse $P(\mathbf{x}|\mathbf{y})$ phrase-based[6] translation models (Koehn et al., 2003), models of the length of the source and target language phrases, and a reordering model.

Together with this log-linear SMT model, Ortiz-Martínez et al. (2010) present online learning techniques that, given a training pair, update the incremental features. In contrast to conventional batch learning techniques, the computational complexity of adding a new training pair is constant, i.e., it does not depend on the number of 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 EM algorithm (Dempster et al., 1977) then it is generally easy to incrementally update the feature given the new training sample. For example, to update a language model with the new translation we simply have to update the current count of each n-gram in $\mathbf{y}$. By contrast, if the EM algorithm is required (e.g. to estimate phrase-based SMT models) the estimation procedure has to be modified because EM is designed to be used in batch learning scenarios. For such feature functions, the incremental version of the EM algorithm (Neal and Hinton, 1999) is applied. For example, phrase-based models are estimated from an hidden Markov (HMM) model (Ueffing and Ney, 2007). Since the HMM model is determined by a hidden alignment variable, the incremental version of the EM algorithm is required to update the model with the new training sample $(\mathbf{x},\mathbf{y})$. A detailed description of the update algorithm for each feature function was presented in Ortiz-Martínez et al. (2010).

4. Experiments

We carried out experiments to assess the performance of the proposed cost-sensitive AL framework for CAT. The idea is to simulate a real-world scenario where a translation agency is hired to

[4] Papineni et al. (2002) obtained the best correlation with human judgments using n-grams of maximum size $n = 4$.

[5] We ignore the effort proportionality constant since it is equal for all sentences.

[6] In contrast with word-based translation models where the fundamental translation unit is the word, phrase-based models translate whole sequences of words. These sequences are called phrases although typically they are not linguistically motivated.

translate a huge amount of text. The experimentation was divided into two parts. First, Section 4.3 describes a typical AL experimentation, such as the one in Haffari et al. (2009), where we studied the learning curves of the SMT model as a function of the number of training sentence pairs. Then, Section 4.4 focuses on the productivity of the whole CAT system. There, we measured, for each ranking function, the quality of the final translations generated by the system as a function of the supervision effort required from the user. With this experimentation, we can observe how the improvements of the underlying SMT model are reflected in the productivity of the whole cost-sensitive AL CAT system.

4.1. Methodology and data

The experimentation carried out comprises the translation of a test corpus using different setups of the proposed cost-sensitive AL framework. Each setup was defined by the ranking function used. All experiments start with a "base" SMT model whose feature functions are trained on the training partition of the Europarl (Koehn and Monz, 2006) corpus, and its log-linear weights are tuned by minimum error-rate training (Och, 2003) to optimize BLEU (Papineni et al., 2002) in the development partition. Then, we run Algorithm 1 until all sentences of the News Commentary corpus (Callison-Burch et al., 2007) are translated into English. We use blocks of size $|\mathcal{B}| = 1000$ (González-Rubio et al., 2012 show that similar results were obtained with other block sizes), and for information density, we arbitrarily set $\gamma = 1$ (i.e., uncertainty and density terms had equal importance). The main figures of the training, development, and test corpora are shown in Table 1.

The reasons to choose the News Commentary corpus as test corpus are threefold: its size is large enough to test the proposed techniques in the long term, its sentences come from a different domain (news) than the sentences in the Europarl corpus (proceedings of the European parliament), and it contains sentences of different topics which allows us to test the robustness of our system against topic-changing data streams. Therefore, by translating the News Commentary corpus we simulate a realistic scenario where translation agencies must be ready to fulfill eclectic real-world translation requirements.

Since an evaluation involving human users is too expensive, we use the reference translations of the News Commentary corpus to simulate the target translations which a human user would want to obtain. At each interaction (see Fig. 2), the prefix validated by the user is computed as the longest common prefix between the translation suggested by the system ($\mathbf{y}_s$) and the reference translation ($\hat{\mathbf{y}}$), and the user correction (k) is given by the first mismatched character between $\mathbf{y}_s$ and $\hat{\mathbf{y}}$. The interaction continued until the longest common prefix is equal to the reference translation.

4.2. Evaluation measures

The goal of the proposed cost-sensitive AL framework is to obtain high translation quality with as few user effort as possible. Therefore, the evaluation is twofold: quality of the generated translations and amount of supervision effort required to generate them. Additionally, we describe how we compute the statistical significance of the results.

Table 1
Main figures of the Spanish–English corpora used, k and M stand for thousands and millions of elementsrespectively.

Corpus	Use	Sentences	Tokens (Spa/Eng)	Vocabulary (Spa/Eng)	Out-of-coverage tokens (Spa/Eng)
Europarl	Training	731k	15.7M/15.2M	103k/64k	–/–
	Development	2k	60k/58k	7k/6k	208/127
News commentary	Test	51k	1.5M/1.2M	48k/35k	13k/11k

4.2.1. Measuring translation quality

We evaluate translation quality using the well-established BLEU (Papineni et al., 2002) score. BLEU computes the geometric mean of the precision of n-grams of various lengths between a candidate translation and a reference translation. This geometric average is multiplied by a factor, namely the brevity penalty, that penalizes candidates shorter than the reference. Following the standard implementation, we consider $n = 4$ as the maximum n-gram length. BLEU is a percentage that measures to which extent the candidate translation contains the same information as the reference translation. Thus, a BLEU value of 100% denotes a perfect match between the candidate translation and the reference translation.

4.2.2. Measuring supervision effort

We estimate the user effort as the number of user actions required to supervise a translation which depend on the supervision method.[7] In the interaction protocol described in Section 3.2, the user can perform two different actions to interact with the system. The first action corresponds to the user looking for the next error and *moving the pointer* to the corresponding position of the translation hypothesis. The second action corresponds to the user replacing the first erroneous character with a *keystroke*.

Bearing this in mind, we compute the keystroke and mouse-action ratio (KSMR) (Barrachina et al., 2009) which has been extensively used to report user effort results in the IMT literature. KSMR is calculated as the number of keystrokes plus the number of movements (mouse actions) divided by the total number of characters of the reference translation. From a user point of view the two types of actions are different, and may require different types of effort (Macklovitch, 2006). A weighted measure could take this into account; however, in these experiments, we assume each action has unit cost.

4.2.3. Statistical significance

We apply statistical significance testing to establish that an observed performance difference between two methods is in fact significant, and has not just arisen by chance. We state a null hypothesis: "Methods A and B do not differ with respect to the evaluation measure of interest" and determine the probability, namely the p-value, that an observed difference has arisen by chance given the null hypothesis. If the p-value is lower than a certain significance level (usually $p < 0.01$, or $p < 0.05$) we can reject the null hypothesis. To do that, we use randomization tests because they free us from worrying about parametric assumptions and they are no less powerful than ordinary t-tests (Noreen, 1989). Specifically, we use a randomization version of the paired t-test based on (Chinchor, 1992):

1. Collect the absolute difference in evaluation measure $Q(\cdot)$ for methods A and B
$$|Q(A) - Q(B)|$$
2. Shuffle N times ($N = 999$ in our experiments) .
3. Count the number of times ($N^{\geqslant}$) that
$$|Q(A') - Q(B')| \geqslant |Q(A) - Q(B)|$$

[7] For example, if instead of using the IMT supervision protocol we ask the user to post-edit the translations, user actions are edit operations, and the natural effort measure is the word error rate, also known as Levenshtein distance.

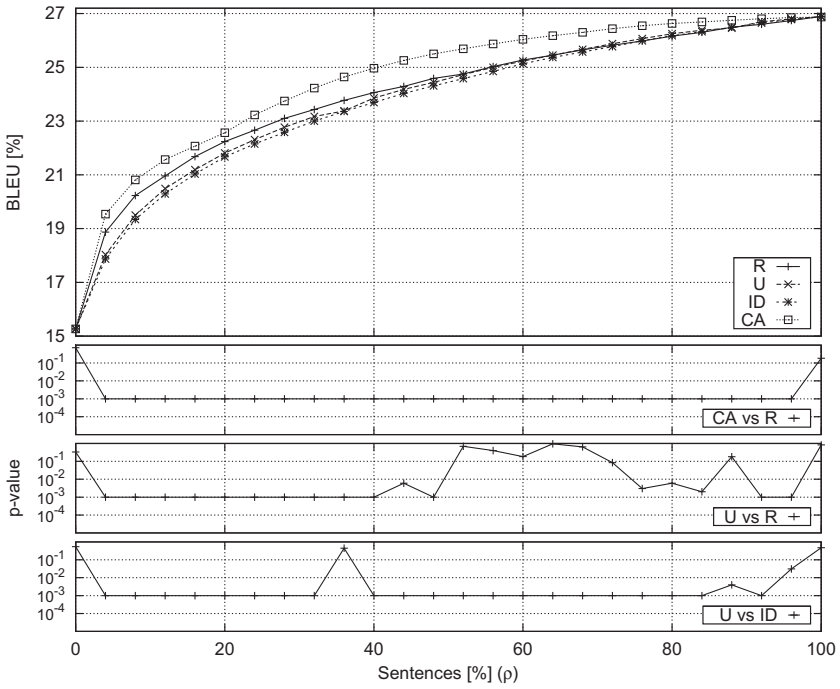

Fig. 3. SMT model performance (BLEU) as a function of the percentage ρ of the corpus used to update it (first panel). We display results for random ranking (R), uncertainty ranking (U), information density ranking (ID), and coverage augmentation ranking (CA). Panels two to four display, on a logarithmic scale, the significance levels (p-values) of the performance differences observed for various pairwise comparisons.

4. The estimate of the p-value is $\frac{N^{\geqslant}+1}{N+1}$ (1 is added to achieve an unbiased estimate).

Initially, we use an evaluation measure $Q(\cdot)$ (e.g. BLEU) to determine the absolute difference between the original outcomes of methods A and B. Then, we repeatedly create shuffled versions A' and B' of the original outcomes, determine the absolute difference between their evaluation metrics, and count the number of times $N^{\geqslant}$ that this difference is equal or larger than the original difference. To create the shuffled versions of the data sets, we iterate over each data point in the original outcomes and decide based on a simulated coin-flip whether data points should be exchanged between A and B. The p-value is the proportion of iterations in which the absolute difference in evaluation metric was indeed larger for the shuffled version (corrected to achieve an unbiased estimate).

4.3. Active learning results

We first studied the learning rates of the different ranking functions in a typical AL experimentation. Here, the performance of the SMT model is studied as a function of the percentage ρ of the corpus used to update it. SMT model performance was measured as the translation quality (BLEU) of the initial automatic translations generated during the interactive supervision process (line 6 in Algorithm 1).

Fig. 3 displays the learning rates observed for each ranking function in Section 3.3: random (R), uncertainty (U), information density (ID) and coverage augmentation (CA). Additionally, we report the significance level of the observed difference for some pairwise comparisons. Similarly as done in Becker (2008), we present p-values on a logarithmic scale. Note that $p = 0.001$ is the smallest possible p-value that can be computed with 999 shuffles in the randomized test; lower p-values will be displayed as a flat line.

Results in Fig. 3 show that coverage augmentation ranking consistently outperformed the random ranking baseline. Additionally, the observed difference was statistically significant as shown in the second panel of the figure. This result shows that coverage augmentation is the ranking function that more effectively detected those sentences that improve most the performance of the SMT model.

Both uncertainty ranking and information density ranking were outperformed by random ranking when supervising up to 50% of the corpus; after that, results for the three ranking functions were very similar and almost no statistical difference was observed (third panel). Additionally, uncertainty ranking and information density ranking obtained virtually the same results; however the slightly better results of uncertainty ranking were statistically significant (fourth panel). That is, the addition of the "representativeness" in information density deteriorated the performance of uncertainty ranking. This counter-intuitive result can be explained by the intrinsic sparse nature of natural language, and particularly by the eclectic topics, e.g. economic, science, or politics, of the sentences in the test corpus.

In the previous experiment, we assumed that all translations were equally costly to supervise. However, different sentences involve different translation costs. Therefore, we then focused on measuring user supervision effort. We studied the user effort required to supervise translations as a function of the percentage of sentences ρ supervised. Fig. 4 shows the KSMR scores obtained by each ranking function, and the significance level of some pairwise ranking function comparisons.

Results show that sentences selected by coverage augmentation required a statistically significant larger amount of effort than the ones selected by random; except when supervising almost all sentences $\rho > 96\%$ where coverage augmentation required a lower amount of effort (second panel in Fig. 4). This indicates that even when all sentences are supervised $\rho = 100\%$ the order in which they are supervised (depending on the ranking function) affects the efficiency of the supervision process.

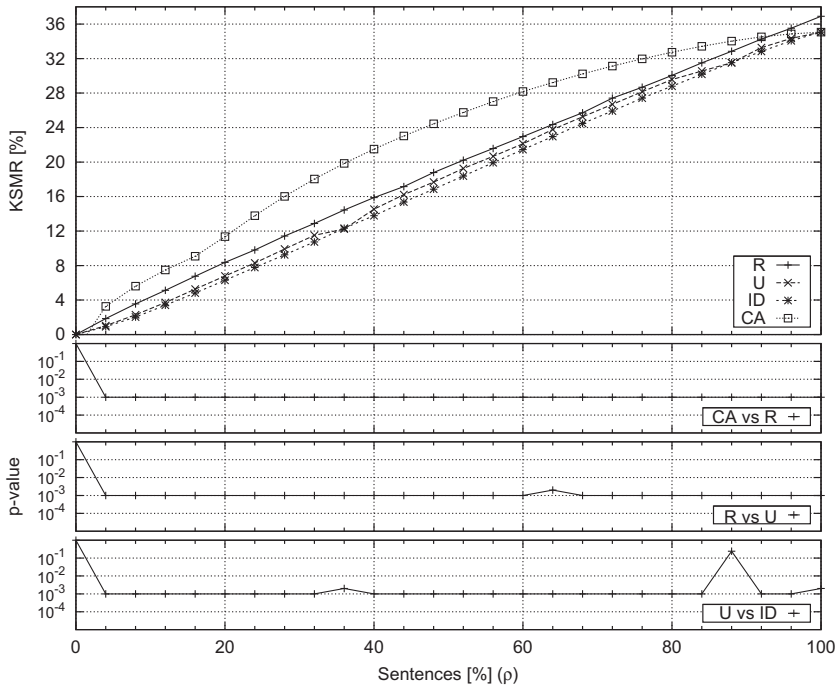

Fig. 4. User effort (KSMR) as a function of the percentage ρ of the corpus used to update the SMT model (first panel). We display results for random ranking (R), uncertainty ranking (U), information density ranking (ID), and coverage augmentation ranking (CA). Panels two to four display, on a logarithmic scale, the significance levels (p-values) of the effort differences observed for various pairwise comparisons.

Regarding uncertainty and information density, both ranking functions required a statistically lower amount of effort than random (third panel), and similarly to the results in Fig. 3, differences between uncertainty and information density were scarce but statistically significant (fourth panel). In this case, sentences selected by information density required a statistically lower amount of effort to be supervised.

4.4. Productivity results

Results in the previous section show that those ranking functions that obtained better learning rates are also those that required more supervision effort, and vice versa. However, from a point of view of a translation agency that has to invest its limited resources, the key point is how to obtain the better productivity. That is, given a required translation quality, how to reduce supervision effort; or symmetrically, given an effort level, how to maximize translation quality.

To answer these questions, we studied the relation between user effort and final translation quality. In contrast with the experimentation in Fig. 3 where we study the learning rates of the SMT model by measuring the quality of its automatic translations, we now are interested in the performance of the complete cost-sensitive AL system. We did that by measuring the translation quality of the translations generated by Algorithm 1 (lines 15 and 17) as a function of the required supervision effort. Note that this final translations are a mixture of automatic and user-supervised translations. The ratio between them is fixed by ρ which permits to adjust system's behavior between a fully automatic SMT system if none translation is supervised ($\rho = 0\%$), or a conventional IMT system where all translations are supervised ($\rho = 100\%$).

Since uncertainty and information density obtain so similar performance in the previous experiments, Fig. 5 compares the cperformance of only random (R), information density (ID), and coverage augmentation (CA) ranking functions. Additionally, we present results of the proposed cost-sensitive AL framework using a static SMT model. The objective was to test the influence of SMT model updating on translation productivity.

Results show a huge leap in productivity when the SMT model was updated with user feedback. This continuous model updating allowed to obtain twice the translation quality with the same level of supervision effort. Regarding the different ranking functions, both information density and coverage augmentation performed similarly yielding slight improvements in productivity with respect to random, particularly for high levels of effort. For example, if a translation quality of 60% BLEU is acceptable, then the human translator would need to modify only a 20% of the characters of the automatically generated translations.

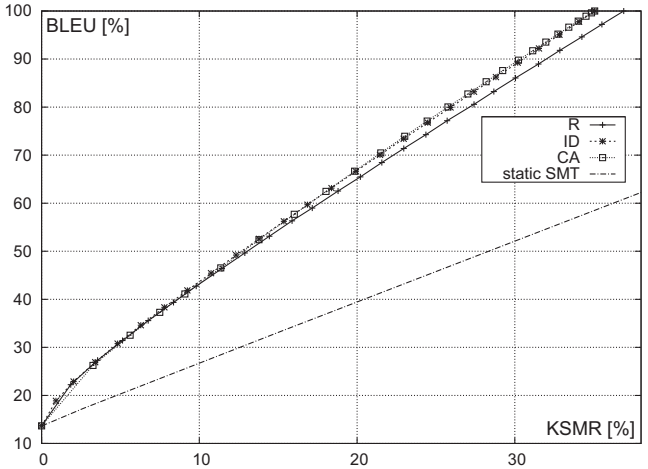

Fig. 5. Final translation quality (BLEU) as a function of user effort (KSMR). We display results for random ranking (R), information density ranking (ID), coverage augmentation ranking (CA), and for a setup where the underlying SMT model is not updated (static SMT).

5. Conclusions and future work

We have presented a cost-sensitive AL framework for CAT designed to boost translation productivity. The two cornerstones of our approach are the selective supervision protocol and the continual SMT model updating with user-supervised translations. Regarding selective supervision, we propose to focus user effort on a subset of sentences that are considered "worth of being supervised" according to a ranking function. The percentage of sentences to be supervised is defined by a tunable parameter which allows to adapt the system to meet task requirements in terms of translation quality, or resources availability. Whenever a new user-supervised translation pair is available, we use it to update a log-linear model. Different online learning techniques are implemented to incrementally update the model.

We evaluated the proposed cost-sensitive AL framework in a simulated translation of real data. Results showed that the use of user-supervised translations reduced to one half the effort required to translate the data. Additionally, the use of an adequate ranking function further improved translation productivity.

The experimental simulation carried out is effective for evaluation, but, to assess the obtained results, we plan to conduct a complete study involving real human users. Productivity could be measured by the actual time it takes a user to translate a test document. This evaluation additionally requires addressing issues of user interface design and user variability, but it is ultimately the most direct evaluation procedure.

An additional direction for further research is to study why random ranking performs so well. We have provided some insights of which are the reasons for this, but we hope that a further study will reveal new hints that may guide us towards the definition of sampling strategies that outperform random sampling. Moreover, the study of productivity-focused ranking functions is a wide research field that should also be explored.

Acknowledgments

Work supported by the European Union Seventh Framework Program (FP7/2007-2013) under the CasMaCat Project (Grants agreement No. 287576), by the Generalitat Valenciana under Grant ALMPR (Prometeo/2009/014), and by the Spanish Government under Grant TIN2012-31723. The authors thank Daniel Ortíz-Martínez for providing us with the log-linear SMT model with incremental features and the corresponding online learning algorithms. The authors also thank the anonymous reviewers for their criticisms and suggestions.

References

Angluin, D., 1988. Queries and concept learning. Machine Learning 2, 319–342.
Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M.A., Marks II, R.J., 1990. Training connectionist networks with queries and selective sampling. In: Touretzky, David S. (Ed.), Advances in Neural Information Processing Systems, vol. 2 Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 566–573.
Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., Vilar, J.-M., 2009. Statistical approaches to computer-assisted translation. Computational Linguistics 35, 3–28.
Becker, M.A., 2008. Active learning – an explicit treatment of unreliable parameters. Ph.D. Thesis, University of Edinburgh.
Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ueffing, N., 2004. Confidence estimation for machine translation. In: Proceedings of the International Conference on Computational Linguistics, pp. 315–321.
Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L., 1993. The mathematics of statistical machine translation: parameter estimation. Computational Linguistics 19, 263–311.
Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J., 2007. (Meta-) evaluation of machine translation. In: Proceedings of the Workshop on Statistical Machine Translation, pp. 136–158.
Casacuberta, F., Vidal, E., 2007. Learning finite-state models for machine translation. Machine Learning 66 (1), 69–91.
Chapelle, O., Schölkopf, B., Zien, A. (Eds.), 2006. Semi-Supervised Learning. MIT Press, Cambridge, MA, URL: <http://www.kyb.tuebingen.mpg.de/ssl-book>.
Chinchor, N., 1992. The statistical significance of the MUC-4 results. In: Proceedings of the Conference on Message Understanding, pp. 30–50.
Cohn, D., Atlas, L., Ladner, R., 1994. Improving generalization with active learning. Machine Learning 15, 201–221.
Dasgupta, S., Hsu, D., 2008. Hierarchical sampling for active learning. In: Proceedings of the International Conference on Machine Learning, pp. 208–215.
Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39 (1), 1–38.
EC, 2009. Translating for a multilingual community. European Commission, Directorate General for Translation, <http://ec.europa.eu/dgs/translation/indexen.htm>.
Foster, G., 2002. Text prediction for translators. Ph.D. Thesis, Université de Montréal.
Foster, G., Isabelle, P., Plamondon, P., 1998. Target-text mediated interactive machine translation. Machine Translation 12 (1/2), 175–194.
Gascó, G., Rocha, M.-A., Sanchis-Trilles, G., Andrés-Ferrer, J., Casacuberta, F., 2012. Does more data always yield better translations? In: Proceedings of the European Chapter of the Association for Computational Linguistics, pp. 152–161.
González-Rubio, J., Ortiz-Martínez, D., Casacuberta, F., 2010. Balancing user effort and translation error in interactive machine translation via confidence measures. In: Proceedings of the Association for Computational Linguistics Conference, pp. 173–177.
González-Rubio, J., Ortiz-Martínez, D., 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, pp. 245–254.
Haddow, B., Koehn, P., 2012. Analysing the effect of out-of-domain data on smt systems. In: Proceedings of the Workshop on Statistical Machine Translation. Association for Computational Linguistics, Montreal, Canada, pp. 422–432.
Haffari, G., Roy, M., Sarkar, A., 2009. Active learning for statistical phrase-based machine translation. In: Proceedings of the North American Chapter of the Association for Computational Linguistics, pp. 415–423.
Isabelle, P., Church, K., 1998. Special Issue on: New Tools for Human Translators, vol. 12. Kluwer Academic Publishers.
Koehn, P., Monz, C., 2006. Manual and automatic evaluation of machine translation between European languages. In: Proceedings of the Workshop on Statistical Machine Translation, pp. 102–121.
Koehn, P., Och, F.J., Marcu, D., 2003. Statistical phrase-based translation. In: Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 48–54.
Koponen, M., 2012. Comparing human perceptions of post-editing effort with post-editing operations. In: Proceedings of the Workshop on Statistical Machine Translation. Association for Computational Linguistics, Montreal, Canada, pp. 181–190.
Langlais, P., Foster, G., Lapalme, G., 2000. TransType: a computer-aided translation typing system. In: Proceedings of the Workshop of the North American Chapter of the Association for Computational Linguistics: Embedded Machine Translation Systems. Association for, Computational Linguistics, pp. 46–51.
Langlais, P., Lapalme, G., 2002. TransType: development-evaluation cycles to boost translator's productivity. Machine Translation 17 (2), 77–98.
Lewis, D., Gale, W., 1994. A sequential algorithm for training text classifiers. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–12.
Lopez, A., 2008. Statistical machine translation. ACM Computational Survey 40, 8:1–8:49.
Macklovitch, E., 2006. TransType2: the last word. In: Proceedings of the Conference on International Language Resources and Evaluation, pp. 167–17.
Neal, R., Hinton, G., 1999. A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in, Graphical Models, 355–368.
NIST, November 2006. NIST 2006 machine translation evaluation official results. <http://www.itl.nist.gov/iad/mig/tests/mt/>.
Noreen, E., 1989. Computer-Intensive Methods for Testing Hypotheses: An Introduction. A Wiley Interscience Publication. Wiley.
Och, F., 2003. Minimum error rate training in statistical machine translation. In: Proceedings of the Association for Computational Linguistics, pp. 160–167.
Och, F., Ney, H., 2002. Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of the Association for Computational Linguistics, pp. 295–302.
Och, F.J., Zens, R., Ney, H., 2003. Efficient search for interactive statistical machine translation. In: Proceedings of the European Chapter of the Association for Computational Linguistics, pp. 387–393.
Ortiz-Martínez, D., García-Varea, I., Casacuberta, F., 2010. Online learning for interactive statistical machine translation. In: Proceedings of the North American Chapter of the Association for Computational Linguistics. pp. 546–554.
Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the Association for Computational Linguistics, pp. 311–318.
Roy, N., McCallum, A., 2001. Toward optimal active learning through sampling estimation of error reduction In: Proceedings of the International Conference on Machine Learning, pp. 441–448.

Settles, B., Craven, M., 2008. An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1070–1079.

Thompson, C.A., Califf, M.E., Mooney, R.J., 1999. Active learning for natural language parsing and information extraction. In: Proceedings of the International Conference on Machine Learning, Bled, Slovenia, pp. 406–414.

Turchi, M., De Bie, T., Cristianini, N., 2009. Learning to translate: a statistical and computational analysis. Tech. Rep., University of Bristol, URL: <https://patterns.enm.bris.ac.uk/files/LearningCurveMain.pdf>.

Ueffing, N., Ney, H., 2005. Application of word-level confidence measures in interactive statistical machine translation. In: Proceedings of the European Association for Machine Translation Conference, pp. 262–270.

Ueffing, N., Ney, H., 2007. Word-level confidence estimation for machine translation. Computational Linguistics 33, 9–40.

Vogel, S., Ney, H., Tillmann, C., 1996. HMM-based word alignment in statistical translation. In: Proceedings of the Association for Computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 836–841.

Zipf, G.K., 1935. The Psychobiology of Language. Houghton-Mifflin.