

# Cost-Sensitive Active Learning for Computer-Assisted Translation

Jesús González-Rubio<sup>a,\*</sup>, Francisco Casacuberta<sup>b</sup>

<sup>a</sup>*Institut Tecnològic d'Informàtica, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain.*

<sup>b</sup>*D. Sistemes Informàtics i Computació, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain.*

---

## 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.

*Keywords:* computer-assisted translation, interactive machine translation, active learning, online learning

---

## 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

---

\*Corresponding author. Phone: (34) 96 387 70 69, Fax: (34) 96 387 72 39.

*Email addresses:* [jegonzalez@iti.upv.es](mailto:jegonzalez@iti.upv.es) (Jesús González-Rubio), [fcn@iti.upv.es](mailto:fcn@iti.upv.es) (Francisco Casacuberta)

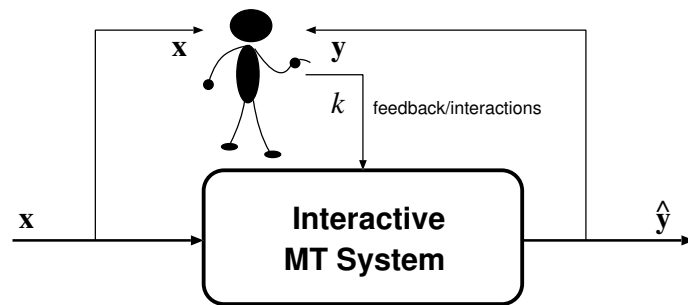


Figure 1: Diagram of an interactive-predictive MT system. To translate a source sentence  $x$ , the user interacts with the system accepting or correcting the proposed translations  $y$ . User feedback  $k$  is used by the system to improve its suggestions.

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 to produce complete translations, or portions thereof, which can be accepted or amended by a human expert, see Figure 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 transla-

tion 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 (Figure 2) is used to efficiently supervise automatic translations. Our goal is to make the translation process as efficient as possible. I.e., 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.

## 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)<sup>1</sup>:

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

---

<sup>1</sup>We use  $Pr(\cdot)$  to denote general probability distributions and  $P(\cdot)$  to denote model-based distributions.

**source** ( $\mathbf{x}$ ): Para ver la lista de recursos

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

<b>interaction-0</b>	$\mathbf{y}_p$ $\mathbf{y}_s$	 <i>To view the resources list</i>
<b>interaction-1</b>	$\mathbf{y}_p$ $k$ $\mathbf{y}_s$	 To view <span style="border: 1px solid black; padding: 0 2px;">a</span> <i>list of resources</i>
<b>interaction-2</b>	$\mathbf{y}_p$ $k$ $\mathbf{y}_s$	 To view a list <span style="border: 1px solid black; padding: 0 2px;">i</span> <i>ng resources</i>
<b>interaction-3</b>	$\mathbf{y}_p$ $k$ $\mathbf{y}_s$	 To view a listing <span style="border: 1px solid black; padding: 0 2px;">o</span> <i>f resources</i>
<b>accept</b>	$\mathbf{y}_p$	 To view a listing of resources

Figure 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.

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

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

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

67 However, despite a huge research effort, SMT systems are still not perfect. To obtain high-quality  
 68 translations, a human expert has to supervise the automatically generated translations. This supervision is  
 69 usually carried out as a separate post-edition step. The IMT framework (Barrachina et al., 2009) constitutes  
 70 an alternative to this serial procedure. In an IMT system, an SMT model and a human expert collaborate  
 71 to generate error-free translations. These translations are generated in a series of interactions between  
 72 the SMT model and the user. At each interaction, the SMT model generates a translation of the source

73 sentence which can be partially or completely accepted and corrected by the user. Each partially corrected  
 74 text segment, called prefix, is then used by the SMT model as additional information to generate better  
 75 translation suggestions. Figure 2 shows an example of a typical IMT session.

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

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

77 which can be straightforwardly rewritten as:

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

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

### 84 3. Cost-Sensitive Active Learning for Computer-Assisted Translation

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

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

99 We implement these ideas as a cost-sensitive AL scenario designed to minimize supervision effort, Sec-  
 100 tion 3.1. We define a new translation work-flow, Section 3.2, that focuses user-effort to only supervise the  
 101 subset of most “informative” translations. Section 3.3 describes the different ranking functions implemented

102 to measure the “informativeness” of each translation, and finally, Section 3.4 presents the incremental SMT  
103 model that is continually updated from user feedback.

### 104 3.1. Active Learning for Computer-Assisted Translation

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

110 Different alternatives to conventional supervised learning have been proposed to address these prob-  
111 lems. For example, semi-supervised learning methods use unlabeled data to help supervised learning  
112 tasks (Chapelle et al., 2006). These methods typically assume that the labeled data set is given and  
113 fixed. In practice, however, semi-supervised methods are allowed to pick a set of unlabeled examples to be  
114 annotated by an expert. In this case, rather than selecting the examples randomly, it may be attractive  
115 to let the learning algorithm to proactively tell us which of them to annotate. This approach is known as  
116 active learning (AL). The idea is to select which training examples to label and the order in which they are  
117 labeled to increase learning efficiency (Angluin, 1988; Atlas et al., 1990; Cohn et al., 1994; Lewis and Gale,  
118 1994). An active learner is considered successful if it obtains better performance than a traditional learner  
119 given the same *number* of training examples. Therefore, AL expedites annotation by reducing the number  
120 of labeled examples required to train an accurate model.

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

### 129 3.2. Translation Work-Flow and Supervision Protocol

130 The proposed AL framework for CAT implies a modification of the conventional IMT work-flow depicted  
131 in Figure 1. The user no longer supervises the translation of all sentences but only of those selected as  
132 “worthy of being supervised”. Since only the most informative sentences are supervised, we maximize the  
133 utility of each user interaction. Final translations however may not be error-free as for conventional IMT.  
134 In exchange, an important reduction in human effort is potentially achievable. Moreover, we can modify

```

input   :  $\mathcal{D}$  (stream of source sentences)
            $\mathbb{M}$  (initial SMT model)
            $\rho$  (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      $\mathcal{B} = \text{getBlockFromStream}(\mathcal{D});$ 
4      $\mathcal{S} = \text{sampling}(\mathcal{B}, \rho);$ 
5     foreach  $\mathbf{x} \in \mathcal{B}$  do
6        $\hat{\mathbf{y}} = \text{translate}(\mathbb{M}, \mathbf{x});$ 
7       if  $\mathbf{x} \in \mathcal{S}$  then
8          $\mathbf{y} = \hat{\mathbf{y}};$ 
9         repeat
10           $\mathbf{y}_p = \text{validPrefix}(\mathbf{y});$ 
11           $\hat{\mathbf{y}}_s = \text{genSuffix}(\mathbb{M}, \mathbf{x}, \mathbf{y}_p);$ 
12           $\mathbf{y} = \mathbf{y}_p \hat{\mathbf{y}}_s;$ 
13          until  $\text{validTranslation}(\mathbf{y});$ 
14           $\mathbb{M} = \text{update}(\mathbb{M}, (\mathbf{x}, \mathbf{y}));$ 
15           $\text{output}(\mathbf{y});$ 
16        else
17           $\text{output}(\hat{\mathbf{y}});$ 
18   until  $\mathcal{D} \neq \emptyset;$ 

```

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

135 the ratio of sentences to be supervised by the user thus modifying the behavior of our system between an  
136 automatic SMT system, and a fully-supervised IMT system. In other words, we can adapt the system’s  
137 behavior to the requirements of each particular translation task.

138 Conventional IMT technology is build over the implicit assumption that the inbound text to be translated  
139 behaves as a text stream (see Figure 1). Source sentences are translated separately and no information

140 is stored (or assumed) about the preceding (or following) sentences, e.g. how many sentences remain  
 141 untranslated. Since the IMT framework uses static SMT models and requires the user to supervise all  
 142 translations, this is not a strong assumption. However, we have to take it into account because information  
 143 about previously supervised translations, and particularly, about following sentences may have great impact  
 144 on the final user effort. We handle the inbound text stream by partitioning the data into blocks of consecutive  
 145 sentences. Within a block, all sentences are available, but once the algorithm moves to the next block, all  
 146 sentences in previous blocks become inaccessible. We use the sentences within a block to estimate the current  
 147 distribution of sentences in the stream, so that the estimation of the “informativeness” of supervising the  
 148 translation of a sentence can be done as accurately as possible.

149 Algorithm 1 shows the pseudo-code that implements the proposed cost-sensitive AL scenario for CAT.  
 150 The algorithm takes as input a stream of source sentences  $\mathcal{D}$ , a “base” SMT model  $\mathbb{M}$ , and an effort level  $\rho$   
 151 denoting the percentage of sentences of each block to be supervised. First, the next block of sentences  $\mathcal{B}$  is  
 152 read from the data stream (line 3). From this block, we sample the set of sentences  $\mathcal{S} \subseteq \mathcal{B}$  that are worthy  
 153 of being supervised by the human expert (line 4). For each sentence in  $\mathcal{B}$ , the current SMT model generates  
 154 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  
 155 collaborates with the system to translate the sentence (lines 8–13). Then, the new sentence pair  $(\mathbf{x}, \mathbf{y})$  is  
 156 used to update the SMT model  $\mathbb{M}$  (line 14), and the human-supervised translation is returned (line 15).  
 157 Otherwise, we directly return the automatic translation  $\hat{\mathbf{y}}$  as the final translation (line 17). Although both  
 158 automatic and user-supervised translations are available, preliminary experiments showed that using both  
 159 translations to update the SMT model resulted in reduced learning rates.

160 Although other translation supervision methods, e.g. post-edition, can be used<sup>2</sup>, we implement the IMT  
 161 user-machine interaction protocol (Figure 2) to supervise each individual translation. Functions between  
 162 lines 8–13 denote this supervision procedure:

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

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

167 **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  
 168 implements Equation (2.4).

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

---

<sup>2</sup>This will imply a modification of lines 8 to 13 in Algorithm 1.



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

### 178 3.3. Sentence Sampling Strategies

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

186 **Uncertainty:** A sentence is as worthy as uncertain is the SMT model of how to translate it.

187 **Representativeness:** A sentence is as worthy as it is “representative” of the sentences in  $\mathcal{B}$ .

188 **Unreliability:** A sentence is as worthy as the amount of unreliably modeled events that it contains.

189 Next sections describe different sampling strategies designed to measure one (or more) of these comple-  
190 mentary properties.

#### 191 3.3.1. Random Ranking ( $R$ )

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

200 *3.3.2. Uncertainty Ranking (U)*

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

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

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

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

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

217 where  $P(y_i|x_j)$  is a word-to-word probability model, and  $x_0$  is the empty source word. Following Ueffing  
 218 and Ney (2005), we use an SMT model 1 (Brown et al., 1993) although other bilingual lexicon models, e.g.,  
 219 model 2 (Brown et al., 1993), or hidden Markov model (Vogel et al., 1996), could also be used.

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

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

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

---

<sup>3</sup>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.

225 *3.3.3. Information Density Ranking (ID)*

226 Uncertainty sampling bases its decisions on individual instances which makes the technique prone to sam-  
 227 ple outliers. The least certain sentences may not be “representative” of other sentences in the distribution,  
 228 in this case, knowing its label is unlikely to improve accuracy on the data as a whole (Roy and McCallum,  
 229 2001). We can overcome this problem by modeling the input distribution explicitly when scoring a sentence.

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

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

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

239 We compute the similarity of two sentences as the geometric mean of the precision of  $n$ -grams (sequences  
 240 of  $n$  consecutive words in a sentence) up to size four<sup>4</sup> between them:

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

241 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  
 242 similarity score is closely related to the widespread translation evaluation score BLEU (Papineni et al., 2002)  
 243 that will be further discussed in section 4.2.1.

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

248 *3.3.4. Coverage Augmentation Ranking (CA)*

249 Sparse data problems are ubiquitous in natural language processing (Zipf, 1935). This implies that some  
 250 rare events will be missing completely from a training set, even when it is very large. Missing events result

---

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

251 in a loss of coverage, a situation when the structure of the model is not rich enough to cover all types of  
 252 input. As a result, words (or sequences thereof) that do not appear in the training set cannot be adequately  
 253 translated (Turchi et al., 2009; Haddow and Koehn, 2012).

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

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

260 The coverage augmentation for each sentence  $\mathbf{x}$  is given by the count of  $n$ -grams in  $\mathbf{x}$  missing in the  
 261 training set  $\mathcal{T}$  that appear in the rest of sentences in the block. I.e., we measure how many missing  $n$ -grams  
 262 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  
 263 length.

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

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

272 Finally, the coverage augmentation score measures the potential SMT model improvement per unit of  
 273 user effort<sup>5</sup>:

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

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

---

<sup>5</sup>We ignore the effort proportionality constant since it is equal for all sentences.

### 279 3.4. Online Training for SMT

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

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

289 Together with this log-linear SMT model, Ortiz-Martínez et al. (2010) present online learning techniques  
290 that, given a training pair, update the incremental features. In contrast to conventional batch learning  
291 techniques, the computational complexity of adding a new training pair is constant, i.e., it does not depend  
292 on the number of training samples. To do that, a set of sufficient statistics is maintained for each feature  
293 function. If the estimation of the feature function does not require the use of the EM algorithm (Dempster  
294 et al., 1977) then it is generally easy to incrementally update the feature given the new training sample. For  
295 example, to update a language model with the new translation we simply have to update the current count of  
296 each  $n$ -gram in  $\mathbf{y}$ . By contrast, if the EM algorithm is required (e.g. to estimate phrase-based SMT models)  
297 the estimation procedure has to be modified because EM is designed to be used in batch learning scenarios.  
298 For such feature functions, the incremental version of the EM algorithm (Neal and Hinton, 1999) is applied.  
299 For example, phrase-based models are estimated from an hidden Markov (HMM) model (Vogel et al., 1996).  
300 Since the HMM model is determined by a hidden alignment variable, the incremental version of the EM  
301 algorithm is required to update the model with the new training sample  $(\mathbf{x}, \mathbf{y})$ . A detailed description of  
302 the update algorithm for each feature function was presented in (Ortiz-Martínez et al., 2010).

## 303 4. Experiments

304 We carried out experiments to assess the performance of the proposed cost-sensitive AL framework for  
305 CAT. The idea is to simulate a real-world scenario where a translation agency is hired to translate a huge  
306 amount of text. The experimentation was divided into two parts. First, Section 4.3 describes a typical  
307 AL experimentation, such as the one in (Haffari et al., 2009), where we studied the learning curves of  
308 the SMT model as a function of the number of training sentence pairs. Then, Section 4.4 focuses on the  
309 productivity of the whole CAT system. There, we measured, for each ranking function, the quality of the

---

<sup>6</sup>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.

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

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

310 final translations generated by the system as a function of the supervision effort required from the user. With  
 311 this experimentation, we can observe how the improvements of the underlying SMT model are reflected in  
 312 the productivity of the whole cost-sensitive AL CAT system.

#### 313 4.1. Methodology and Data

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

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

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

334 is given by the first mismatched character between  $\mathbf{y}_s$  and  $\hat{\mathbf{y}}$ . The interaction continued until the longest  
335 common prefix is equal to the reference translation.

## 336 4.2. Evaluation Measures

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

### 341 4.2.1. Measuring Translation Quality

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

### 349 4.2.2. Measuring Supervision Effort

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

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

### 361 4.2.3. Statistical Significance

362 We apply statistical significance testing to establish that an observed performance difference between  
363 two methods is in fact significant, and has not just arisen by chance. We state a null hypothesis: “Methods

---

<sup>7</sup>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.

364 A and B do not differ with respect to the evaluation measure of interest” and determine the probability,  
 365 namely the p-value, that an observed difference has arisen by chance given the null hypothesis. If the p-value  
 366 is lower than a certain significance level (usually  $p < 0.01$ , or  $p < 0.05$ ) we can reject the null hypothesis. To  
 367 do that, we use randomization tests because they free us from worrying about parametric assumptions and  
 368 they are no less powerful than ordinary t-tests (Noreen, 1989). Specifically, we use a randomization version  
 369 of the paired t-test based on (Chinchor, 1992):

- 370 1. Collect the absolute difference in evaluation measure  $Q(\cdot)$  for methods A and B  
 371  $|Q(A) - Q(B)|$
- 372 2. Shuffle  $N$  times ( $N = 999$  in our experiments)
- 373 3. Count the number of times ( $N^{\geq}$ ) that  
 374  $|Q(A') - Q(B')| \geq |Q(A) - Q(B)|$
- 375 4. The estimate of the p-value is  $\frac{N^{\geq} + 1}{N + 1}$   
 376 (1 is added to achieve an unbiased estimate)

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

### 385 4.3. Active Learning Results

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

390 Figure 3 displays the learning rates observed for each ranking function in Section 3.3: random (R),  
 391 uncertainty (U), information density (ID) and coverage augmentation (CA). Additionally, we report the  
 392 significance level of the observed difference for some pairwise comparisons. Similarly as done in (Becker,  
 393 2008), we present p-values on a logarithmic scale. Note that  $p = 0.001$  is the smallest possible p-value that  
 394 can be computed with 999 shuffles in the randomized test; lower p-values will be displayed as a flat line.

395 Results in Figure 3 show that coverage augmentation ranking consistently outperformed the random  
 396 ranking baseline. Additionally, the observed difference was statistically significant as shown in the second



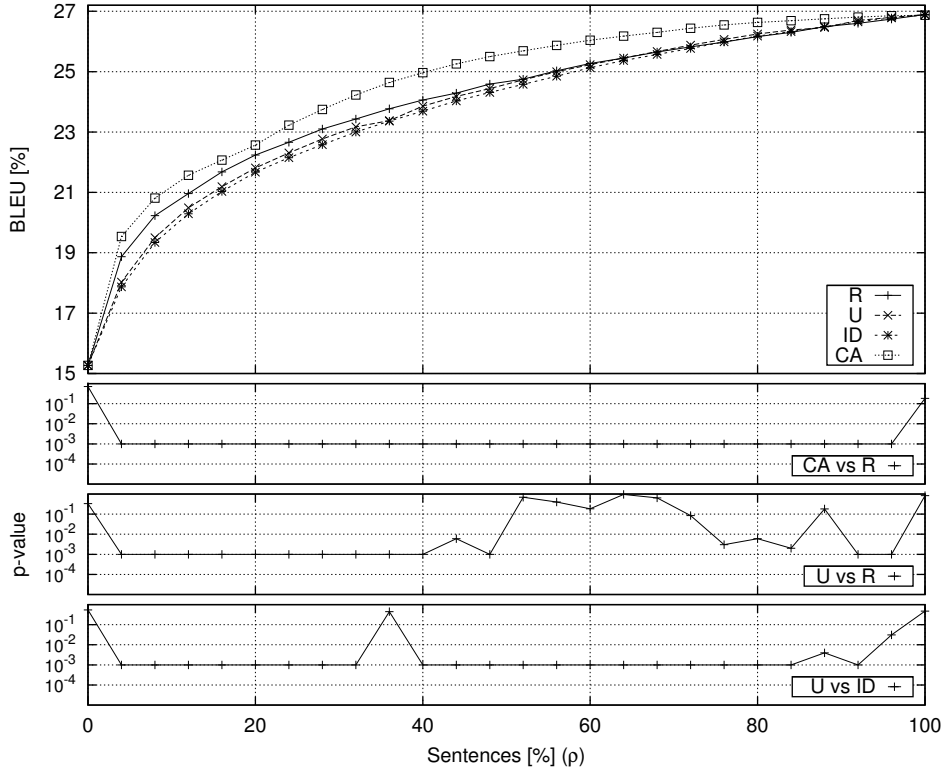


Figure 3: SMT model performance (BLEU) as a function of the percentage  $\rho$  of the corpus used to update it (first panel). We display results for random ranking (R), uncertainty ranking (U), information density ranking (ID), and coverage augmentation ranking (CA). Panels two to four display, on a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

397 panel of the figure. This result shows that coverage augmentation is the ranking function that more effectively  
 398 detected those sentences that improve most the performance of the SMT model.

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

407 In the previous experiment, we assumed that all translations were equally costly to supervise. However,  
 408 different sentences involve different translation costs. Therefore, we then focused on measuring user super-  
 409 vision effort. We studied the user effort required to supervise translations as a function of the percentage  
 410 of sentences  $\rho$  supervised. Figure 4 shows the KSMR scores obtained by each ranking function, and the

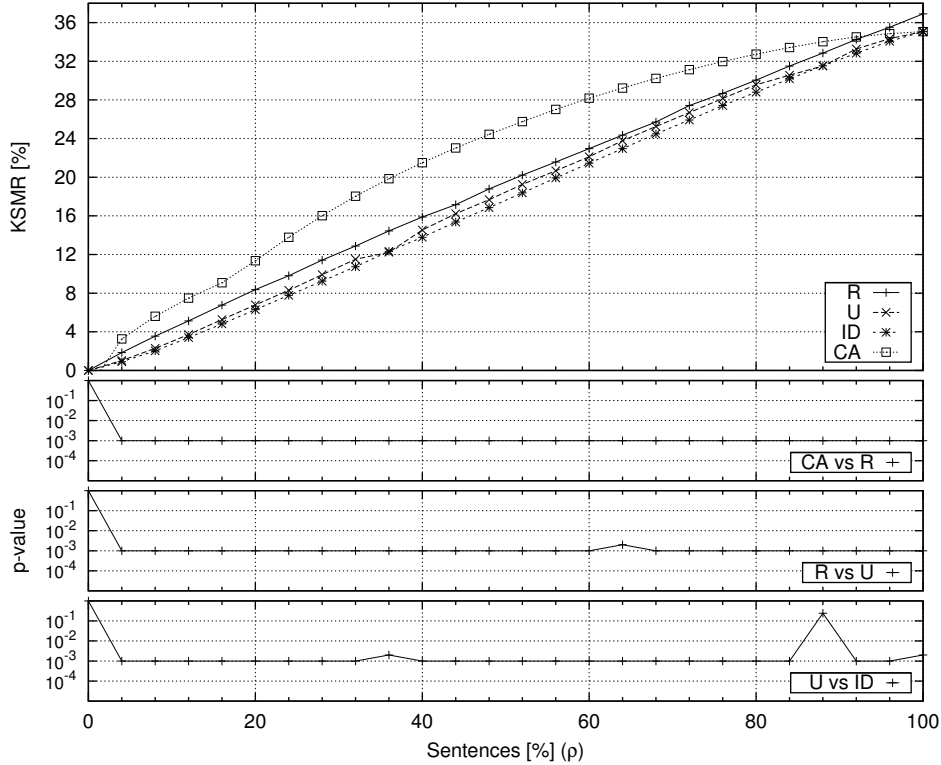


Figure 4: User effort (KSMR) as a function of the percentage  $\rho$  of the corpus used to update the SMT model (first panel). We display results for random ranking (R), uncertainty ranking (U), information density ranking (ID), and coverage augmentation ranking (CA). Panels two to four display, on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

411 significance level of some pairwise ranking function comparisons.

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

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

#### 421 4.4. Productivity Results

422 Results in the previous section show that those ranking functions that obtained better learning rates  
 423 are also those that required more supervision effort, and vice versa. However, from a point of view of

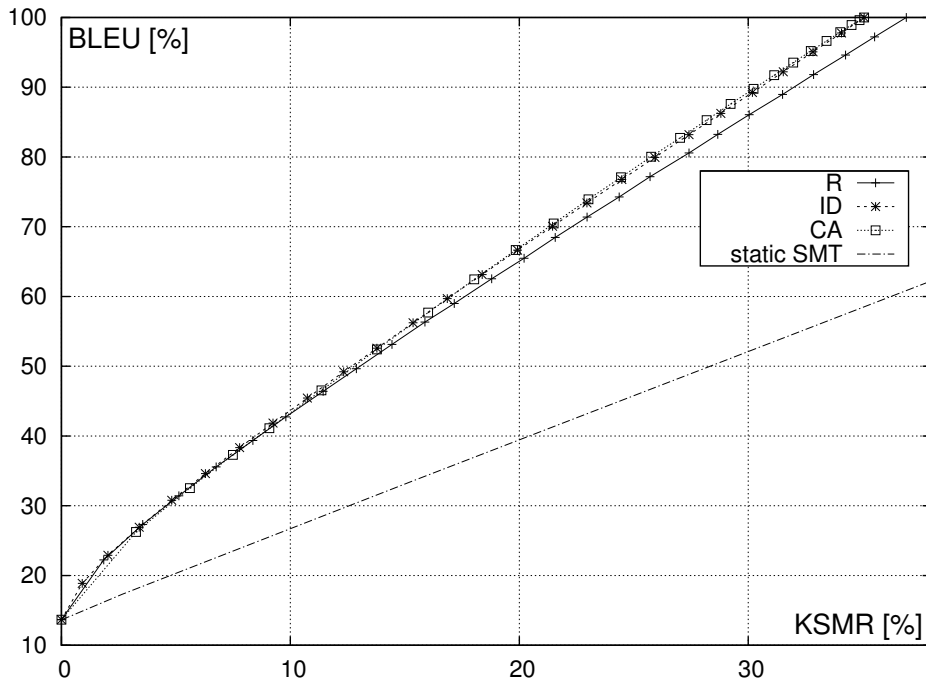


Figure 5: Final translation quality (BLEU) as a function of user effort (KSMR). We display results for random ranking (R), information density ranking (ID), coverage augmentation ranking (CA), and for a setup where the underlying SMT model is not updated (static SMT).

424 a translation agency that has to invest its limited resources, the key point is how to obtain the better  
 425 productivity. I.e., given a required translation quality, how to reduce supervision effort; or symmetrically,  
 426 given an effort level, how to maximize translation quality.

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

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

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

## 446 5. Conclusions and Future Work

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

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

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

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

## 467 6. Acknowledgments

468 Work supported by the European Union Seventh Framework Program (FP7/2007-2013) under the Cas-  
469 MaCat project (grants agreement n° 287576), by the Generalitat Valenciana under grant ALMPR (Prom-  
470 eteo/2009/014), and by the Spanish government under grant TIN2012-31723. The authors thank Daniel

471 Ortíz-Martínez for providing us with the log-linear SMT model with incremental features and the corre-  
472 sponding online learning algorithms. The authors also thank the anonymous reviewers for their criticisms  
473 and suggestions.

## 474 References

- 475 Angluin, D., April 1988. Queries and concept learning. *Machine Learning* 2, 319–342.
- 476 Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M. A., Marks, II, R. J., 1990. Advances in neural information processing systems  
477 2. Ch. Training connectionist networks with queries and selective sampling, pp. 566–573.
- 478 Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E.,  
479 Vilar, J.-M., March 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics* 35, 3–28.
- 480 Becker, M. A., 2008. Active learning - an explicit treatment of unreliable parameters. Ph.D. thesis, University of Edinburgh.
- 481 Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ueffing, N., 2004. Confidence  
482 estimation for machine translation. In: *Proceedings of the international conference on Computational Linguistics*. pp. 315–  
483 321.
- 484 Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., Mercer, R. L., 1993. The mathematics of statistical machine translation:  
485 parameter estimation. *Computational Linguistics* 19, 263–311.
- 486 Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J., 2007. (Meta-) evaluation of machine translation. In:  
487 *Proceedings of the Workshop on Statistical Machine Translation*. pp. 136–158.
- 488 Casacuberta, F., Vidal, E., 2007. Learning finite-state models for machine translation. *Machine Learning* 66 (1), 69–91.
- 489 Chapelle, O., Schölkopf, B., Zien, A. (Eds.), 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.  
490 URL <http://www.kyb.tuebingen.mpg.de/ssl-book>
- 491 Chinchor, N., 1992. The statistical significance of the muc-4 results. In: *Proceedings of the Conference on Message Under-*  
492 *standing*. pp. 30–50.
- 493 Cohn, D., Atlas, L., Ladner, R., 1994. Improving generalization with active learning. *Machine Learning* 15, 201–221.
- 494 Dasgupta, S., Hsu, D., 2008. Hierarchical sampling for active learning. In: *Proceedings of the international conference on*  
495 *Machine learning*. pp. 208–215.
- 496 Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the*  
497 *Royal Statistical Society*. 39 (1), 1–38.
- 498 EC, 2009. Translating for a multilingual community. [http://ec.europa.eu/dgs/translation/index\\_en.htm](http://ec.europa.eu/dgs/translation/index_en.htm), european Com-  
499 mission, directorate general for translation.
- 500 Foster, G., may 2002. Text prediction for translators. Ph.D. thesis, Université de Montréal.
- 501 Foster, G., Isabelle, P., Plamondon, P., January 1998. Target-text mediated interactive machine translation. *Machine Transla-*  
502 *tion* 12 (1/2), 175–194.
- 503 Gascó, G., Rocha, M.-A., Sanchis-Trilles, G., Andrés-Ferrer, J., Casacuberta, F., 2012. Does more data always yield better  
504 translations? In: *Proceedings of the European Chapter of the Association for Computational Linguistics*. pp. 152–161.
- 505 González-Rubio, J., Ortiz-Martínez, D., Casacuberta, F., 2010. Balancing user effort and translation error in interactive  
506 machine translation via confidence measures. In: *Proceedings of the Association for Computational Linguistics Conference*.  
507 pp. 173–177.
- 508 González-Rubio, J., Ortiz-Martínez, D., Casacuberta, F., 2012. Active learning for interactive machine translation. In: *Pro-*  
509 *ceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. pp. 245–254.
- 510 Haddow, B., Koehn, P., June 2012. Analysing the effect of out-of-domain data on smt systems. In: *Proceedings of the Workshop*  
511 *on Statistical Machine Translation*. Association for Computational Linguistics, Montreal, Canada, pp. 422–432.

512 Haffari, G., Roy, M., Sarkar, A., 2009. Active learning for statistical phrase-based machine translation. In: Proceedings of the  
513 North American Chapter of the Association for Computational Linguistics. pp. 415–423.

514 Isabelle, P., Church, K., January 1998. Special issue on: New tools for human translators. Vol. 12. Kluwer Academic Publishers.

515 Koehn, P., Monz, C., 2006. Manual and automatic evaluation of machine translation between european languages. In: Pro-  
516 ceedings of the Workshop on Statistical Machine Translation. pp. 102–121.

517 Koehn, P., Och, F. J., Marcu, D., 2003. Statistical phrase-based translation. In: Proceedings of the North American Chapter  
518 of the Association for Computational Linguistics on Human Language Technology. pp. 48–54.

519 Koponen, M., June 2012. Comparing human perceptions of post-editing effort with post-editing operations. In: Proceedings  
520 of the Workshop on Statistical Machine Translation. Association for Computational Linguistics, Montreal, Canada, pp.  
521 181–190.

522 Langlais, P., Foster, G., Lapalme, G., 2000. TransType: a computer-aided translation typing system. In: Proceedings of the  
523 Workshop of the North American Chapter of the Association for Computational Linguistics: Embedded Machine Translation  
524 Systems. Association for Computational Linguistics, pp. 46–51.

525 Langlais, P., Lapalme, G., September 2002. TransType: development-evaluation cycles to boost translator’s productivity.  
526 Machine Translation 17 (2), 77–98.

527 Lewis, D., Gale, W., 1994. A sequential algorithm for training text classifiers. In: Proceedings of the ACM SIGIR conference  
528 on Research and development in information retrieval. pp. 3–12.

529 Lopez, A., August 2008. Statistical machine translation. ACM Computational Survey 40, 8:1–8:49.

530 Macklovitch, E., 2006. TransType2: the last word. In: Proceedings of the conference on International Language Resources and  
531 Evaluation. pp. 167–17.

532 Neal, R., Hinton, G., 1999. A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in  
533 graphical models, 355–368.

534 NIST, November 2006. NIST 2006 machine translation evaluation official results. [http://www.itl.nist.gov/iad/mig/tests/  
535 mt/](http://www.itl.nist.gov/iad/mig/tests/mt/).

536 Noreen, E., 1989. Computer-intensive methods for testing hypotheses: an introduction. A Wiley Interscience publication. Wiley.

537 Och, F., 2003. Minimum error rate training in statistical machine translation. In: Proceedings of the Association for Compu-  
538 tational Linguistics. pp. 160–167.

539 Och, F., Ney, H., 2002. Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings  
540 of the Association for Computational Linguistics. pp. 295–302.

541 Och, F. J., Zens, R., Ney, H., 2003. Efficient search for interactive statistical machine translation. In: Proceedings of the  
542 European chapter of the Association for Computational Linguistics. pp. 387–393.

543 Ortiz-Martínez, D., García-Varea, I., Casacuberta, F., 2010. Online learning for interactive statistical machine translation. In:  
544 Proceedings of the North American Chapter of the Association for Computational Linguistics. pp. 546–554.

545 Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In:  
546 Proceedings of the Association for Computational Linguistics. pp. 311–318.

547 Roy, N., McCallum, A., 2001. Toward optimal active learning through sampling estimation of error reduction. In: Proceedings  
548 of the International Conference on Machine Learning. pp. 441–448.

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

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

553 Turchi, M., De Bie, T., Cristianini, N., 2009. Learning to translate: a statistical and computational analysis. Tech. rep.,  
554 University of Bristol.

555 URL <https://patterns.enm.bris.ac.uk/files/LearningCurveMain.pdf>  
556 Ueffing, N., Ney, H., 2005. Application of word-level confidence measures in interactive statistical machine translation. In:  
557 Proceedings of the European Association for Machine Translation conference. pp. 262–270.  
558 Ueffing, N., Ney, H., 2007. Word-level confidence estimation for machine translation. Computational Linguistics 33, 9–40.  
559 Vogel, S., Ney, H., Tillmann, C., 1996. HMM-based word alignment in statistical translation. In: Proceedings of the Association  
560 for Computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 836–841.  
561 Zipf, G. K., 1935. The Psychobiology of Language. Houghton-Mifflin.