*Title Page

Improving On-line Handwritten Recognition in Interactive Machine Translation

Vicent Alabau*, Alberto Sanchis, Francisco Casacuberta

Institut Tecnològic d'Informatica, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain

Abstract

On-line handwriting text recognition (HTR) could be used as a more natural way of interaction in many interactive applications. However, current HTR technology is far from developing error-free systems and, consequently, its use in many applications is limited. Despite this, there are many scenarios, as in the correction of the errors of fully-automatic systems using HTR in a post-editing step, in which the information from the specific task allows to constrain the search and therefore to improve the HTR accuracy. For example, in machine translation (MT), the on-line HTR system can also be used to correct translation errors. The HTR can take advantage of information from the translation problem such as the source sentence that is translated, the portion of the translated sentence that has been supervised by the human, or the translation error to be amended. Empirical experimentation suggests that this is a valuable information to improve the robustness of the on-line HTR system achieving remarkable results.

Keywords: interactive pattern recognition, on-line handwritten text recognition, interactive machine translation, human interaction

*Corresponding author: Phone: (34) 96 387 70 69, Fax: (34) 96 387 72 39.
Email addresses: valabau@iti.upv.es (Vicent Alabau), asanchis@iti.upv.es (Alberto Sanchis), fcn@iti.upv.es (Francisco Casacuberta)

1. Introduction

Since the breakout of tactile smartphones, the number of devices featuring a touch-screen has been increasing at a fast pace. The success of tactile smartphones has fostered a new kind of keyboardless technology: the tablet computers. They have been presented as a substitute of paper notebooks although the possibilities this new technology may provide are still to be unveiled.

In that context, on-line handwritten text recognition (HTR) can play a crucial role. First, because to input text in such devices using a virtual keyboard is far from the efficiency of regular keyboards. Secondly, handwriting is a natural way to communicate. Withal, an HTR interface can commit recognition errors. Thus, if the HTR system is not robust enough, user experience could be negatively affected hindering its use. In this regard, many works have tried to improve HTR accuracy. Primarily focusing on feature extraction and modeling [1, 2, 3]. Other authors have tackled the problem of automatically correcting errors from the system output in order to provide a more accurate input to higher-level applications. For instance, Quiniou et al. [4] propose a technique to improve the performance of a HTR system by obtaining a consensus hypothesis out of a n-best lists, and then, characterizing the errors and correcting them. Similarly, Farooq et al. [5] use a translation model to conduct an automatic post-editing. Additionally, Devlin et al. [6] used a machine translation system to rerank an OCR n-best list. The idea was that *easily translatable* options would have a better syntax, which in the end resulted in small accuracy improvements. Nevertheless, those works did not leverage any contextual information of the specific task at hand, a topic that, in our opinion, has received little attention. Following this line of research, Toselli et al. [7] explored the use of on-line HTR for interactive transcription of text images. In that work, the user was expected to correct erroneously recognized words by handwriting the correction using a tactile display. The authors took advantage of the erroneously predicted word and the previous one to improve HTR robustness.

Inspired by Toselli et al. [7, 8], we address the problem of using an on-line HTR system to correct the errors in a *machine translation* (MT) application. State-of-the-art MT systems usually cannot perform translations to fit quality demands by the translation industry. Hence, it is typical to have the automatically produced output documents revised by a professional translator. In this manual process, known as post-editing (PE), the human expert can spend hours of work to achieve high-quality translations. *Interactive machine translation* (IMT) [9, 10, 11] was developed to deal with this problem. In IMT, a human expert is introduced in the middle of the translation process. This way, she can amend errors from the system output and useful feedback is used by the system to automatically improve the part of the translation to be revised.

The usual way to introduce the corrections in IMT is by means of the keyboard where the mouse is used to fix the position [12]. However, other interaction modalities are also possible. For example, speech interaction was studied in [13, 14, 15]. There, several scenarios were proposed, in which the user was expected to utter aloud parts of the current hypothesis along with one or more corrections. Later, we proposed the use of on-line HTR to IMT in [16, 17]. To our knowledge, our work has been the first approach to on-line HTR in IMT so far. Nonetheless, those works presented very preliminary results explaining simple contextual models and HTR interaction restricted to isolated words.

In this paper we present relevant novelties with respect to previous work that can be summarized in two main improvements. First, we introduce a new HTR model that leverages state-of-the-art phrase-based models, whereas previous work was based only on word-based translation models. Second, we extend the interaction scheme to allow sequences of words (phrases) to be written and not just isolated words. In addition, we propose a method to recover efficiently from HTR errors using contextual menus. Finally, a new and exhaustive experimental study is presented to evaluate all those novel contributions and preliminary ideas.

The remainder of this paper is organized as follows. First, the process to produce high-quality translations is introduced in Sec. 2. Second, in Sec. 3 several alternatives to incorporate contextual information from the translation problem into the HTR decoding will be explored. Section 4 is devoted to the evaluation of the proposed models. Finally, conclusions and future work will be discussed in Sec. 6.

2. Producing High-Quality Translations

In the last years, *machine translation* (MT) has become a strategic asset in the translation industry. MT is used to speed up the translation process since it enables the automatic translation of large amounts of documents. In this context, MT is approached under a statistical framework, due to the fact that statistical MT allows companies to build customized, topic-specific MT systems very economically. Here, the problem consists in finding the most likely translation $\hat{t}$ in a target language given a source sentence s in a source language,

$$\hat{t} = \operatorname*{argmax}_{t} Pr(t \mid s) \tag{1}$$

which can be modeled in different ways [18].

2.1. Post-editing a Machine Translation Output

Although leveraging MT can be very convenient, it is usually the case that the translation quality does not meet the user requirements. Thus, the MT output must be revised. The process of revising and amending the system output, known as *post-editing* (PE), consists in deleting, inserting, substituting and swapping text from the MT output to achieve the desired quality in the translation. This is an expensive task, since the users should review the whole output and correct manually the translation errors. In the cases in which the automatically produced translations are of low quality, PE can eventually require more effort than manually translating the source input from the scratch. Moreover, in PE, the system does not take advantage of the human corrections.

2.2. Interactive Machine Translation

The MT paradigm is shifting slowly but steady towards an interactive MT scenario (IMT). In IMT [9, 10, 11] the system goal is not to produce translations in a completely automatic way and then perform a completely unassisted

PE. On the contrary, IMT aims at building the translation collaboratively with the user as a professional advisor, so that the effort to produce a satisfactory output is minimized.

A typical approach to IMT is shown in Fig. 1. A source sentence s is given to the IMT system. First, the system outputs a translation hypothesis $\hat{t}$ in the target language, that would correspond to the output of fully automated MT system (i.e., based on Eq. (1)). Next, the user analyzes the source sentence and the current hypothesis, and validates the longest error-free prefix p finding the first error. Then, the user amends the erroneous word by typing the correct word d. Based on this amendment, the system creates a new validated prefix $p \cdot d$, with $\cdot$ as a concatenation operator. With that information, the system is able to produce a new, hopefully improved, translation $\hat{t}$ that is coherent with the information provided, that is, $p \cdot d$ must be a prefix of the new $\hat{t}$. This process is repeated until the user agrees with the quality of the resulting translation. In this work we assume that this protocol is performed left-to-right, but other protocols are also possible.

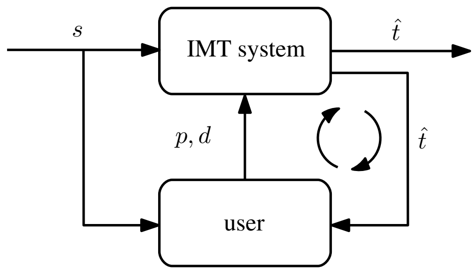


Figure 1: Diagram of a typical approach to IMT

The iterative nature of the process is emphasized by the loop in Fig. 1, which indicates that, for a source sentence to be translated, several interactions between the user and the system could be performed. In each interaction, the system produces the most probable translation $\hat{t}$ that is coherent with the prefix formed by concatenating the previous prefix p and the user correction d:

$$\hat{t} = \underset{t:\tau(t,p\cdot d)}{\operatorname{argmax}} Pr(t \mid p, d, s) \tag{2}$$

where $\tau(t, p \cdot d)$ is a function that is true if $p \cdot d$ is a prefix of t. It is worthy of note that the main difference between Eq. (1) and Eq. (2) is that, in the second case, $\hat{t}$ is must be coherent with the validated prefix $p \cdot d$. Since the probabilistic models in Eq. (1) and Eq. (2) are estimated in the same way, Eq. (2) can be considered as a constrained search problem of the classical MT problem. In fact, at the beginning, when the user has not validated any prefix, Eq. (1) and Eq. (2) are equivalent equations. In addition, adaptive approaches can also be assumed, where the system is able to learn from each user interaction to improve the underlying statistical models [19].

For the sake of a better understanding, a typical translation IMT session is exemplified in Fig. 2. First, the system starts with an empty prefix, so it proposes a full hypothesis. Then, the user corrects the first error, *not*, by typing ‘is’. Next, the system proposes a new suffix, in which the first word, *not*, has been automatically corrected. The user amends *at* by typing ‘in’. Finally, as the new proposed suffix is correct, the process ends. Note that 4 operations would have been needed in a PE scenario, whereas only 2 are needed in IMT. In this example, the user types the complete wrong word. Nevertheless, it is straightforward to extend this operation to the character level instead of word level.

SOURCE (s):		si alguna función no se encuentra disponible en su red
REFERENCE ($\boldsymbol{r}$):		if any feature is not available in your network
ITER-0	(p)	
ITER-1	$(\hat{t})$	if any feature not is available on the network
	(p)	*if any feature*
	(d)	**is**
ITER-2	$(\hat{t})$	if any feature is not available at the network
	(p)	if any feature is *not available*
	(d)	**in**
FINAL	$(\hat{t})$	if any feature is not available in your network
	$(\hat{t} \equiv \boldsymbol{r})$	if any feature is not available in your network

Figure 2: Example of an IMT session for translating a Spanish sentence s to an English sentence t. Initially, in iteration 0, the prefix is empty, i.e., the user has not performed any validation. In iteration 1, the system proposes a fully automatic translation $\hat{t}$. Then, the user finds the first error and amends it by introducing the correct word (d), which is shown in **boldface**. As a result, the user has implicitly validated a prefix (p), shown in *italics*. The concatenation of the prefix and the corrected word constitutes a new prefix for the next iteration (displayed in blue). The process continues until the user is satisfied with the solution. Note that 4 operations would have been needed in a PE scenario, whereas only 2 are needed in IMT.

3. Using On-Line HTR to Correct MT Output

Typically, the correction of MT output is performed using a keyboard and, occasionally, a mouse to position the cursor [12]. Professional translators agree that this approach has been proved to be efficient. However, the user needs to be in front of a desktop computer which imposes some restrictions regarding where and how the work is to be done. Laptop computers can also be used, although arguably performance could be diminished because of the use of uncomfortable laptop keyboards and track pads. Thus, although e-pen interaction may sound impractical for texts that need a large amounts of corrections, there is a number of circumstances where e-pen interaction can be more suitable. For example, it can be well suited for amending sentences with few errors, as the revision of human post-edited sentences, or translations where the system has a high confidence that the output is of good quality. Furthermore, it would allow to perform such tasks while commuting, traveling or sitting comfortably on the couch in the living room.

Now, imagine an application devised to translate documents. On the one hand, there is a text area with the output of an automatic machine translation system. As this output may contain errors, the user of the application reads the output to locate the first error. The reading is performed in a specific order, left-to-right in most western languages, for instance. Let us assume that when the user finds the first error, all the words before it have already been revised and validated. Thus, they can be regarded as correct. Once the error has been located, the user introduces the correction with a stylus. As a result, the system receives a position where the error is located, a word that is incorrect (the word pointed by the position) and a sequence of pen strokes that represent the correct word in that position. On the other hand, the source document to be transcribed is shown to the user. There is a strong relationship among the words in the source sentence and the words in the target sentence.

Figure 3 is a mock-up of a possible application on a tablet device for such scenario. The screen is divided in two sections. First, the upper part shows the source document, and probably the source sentence being currently translated, s, is highlighted appropriately. Second, the lower section contains the current state of the translation, t. Since we assume that post-editing is usually performed from left to right, the text which has already been revised and validated is highlighted. On the other hand, the text which is to be revised is displayed grayed out. From the sentence currently being translated we can identify three parts: the revised prefix of the sentence, p, the error committed by the system, e, and the correction proposed by the user introducing strokes with a stylus, x.

In a scenario as described above, the HTR subsystem should make few errors to make the application usable. The aim of this work is to devise a robust HTR system that allows a potential user to revise and correct the output of a machine translation system using an electronic pen. To this regard, we assume that the user will introduce the corrections by writing over the word or sequences of words (phrases) she judges to be incorrect. Thus, the problem of on-line HTR consists in converting a sequence of strokes, x, into a word or phrase in text format, d. The strokes can be acquired from a stylus, electronic pen or a touch-screen.

3.1. System Baseline

The baseline approach to the problem from a statistical point of view is to obtain the most likely decoding d given the strokes x,

$$\hat{d} = \operatorname*{argmax}_{d} Pr(d \mid x) = \operatorname*{argmax}_{d} Pr(d) Pr(x \mid d) \tag{3}$$

where $Pr(d)$ can be represented by a language model and $Pr(x \mid d)$ by morphological models.

The morphological models can be modeled by hidden Markov models [2] or neural networks [1]. On the other hand, a common and practical approach to model $Pr(d)$ is by means of n-grams [20]. The description of an on-line HTR system would end here for most applications. However, our purpose is to take advantage of the information available in the IMT application to make on-line HTR more robust. In the remainder of this section, we will introduce gradually the different kinds of information sources into the language model. With the addition of each of them, we aim to make the on-line HTR system more robust.

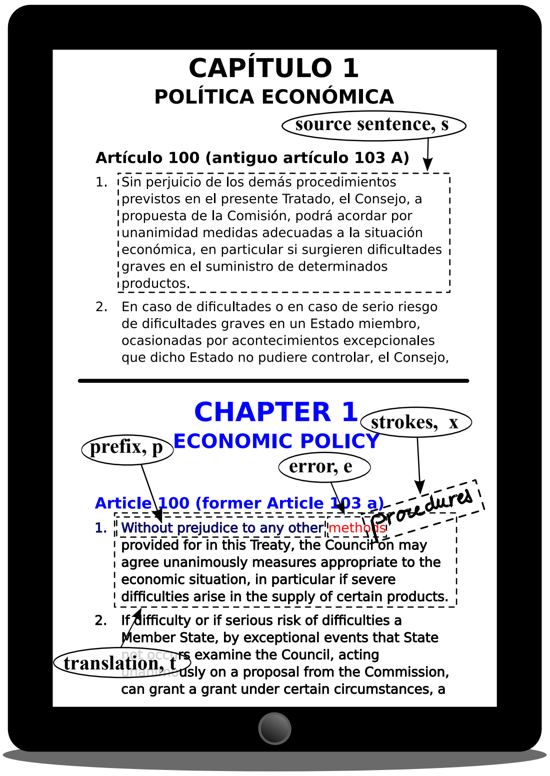


Figure 3: Mock-up of an interactive machine translation application on a tablet device.

3.2. *Discarding the produced error*

In the e-pen enabled IMT interface aforementioned, the user is expected to write the strokes over the erroneously translated word, and thus, the system knowns what word the user wants to replace. Therefore, the first and easiest approach is to remove the erroneous word e from the list of candidate hypotheses. This way, Eq. (3) becomes

$$\hat{d} = \underset{d \neq e}{\operatorname{argmax}}\, Pr(d) Pr(x \mid d) \tag{4}$$

3.3. *Exploiting information from the revised translation*

The second sensible approach to take is to add information regarding the revised translation prefix, p. Again, from Eq. (3) we can derive an HTR system that takes into account previously validated words:

$$\hat{d} = \underset{d}{\operatorname{argmax}}\, Pr(d \mid x, p) \approx \underset{d}{\operatorname{argmax}}\, Pr(d \mid p) Pr(x \mid d) \tag{5}$$

under the assumption that $Pr(x \mid d, p)$ does not depend on p if d is known. In addition, $Pr(d \mid p)$ is a prefix language model, i.e., the probability of d depends on the left-context. Of course, we can also discard the erroneous word from Eq. (5),

$$\hat{d} \approx \underset{d \neq e}{\operatorname{argmax}}\, Pr(d \mid p) Pr(x \mid d) \tag{6}$$

These techniques can be extrapolated to most post-editing tasks. In fact, Toselli et al. [7] used the erroneous word and a 2-gram model to improve the HTR performance for interactive transcription of text images. Next, we will show how the information regarding the translation process can be exploited for further improve HTR decoding.

3.4. Leveraging information from the source sentence

A specific source of information that can help to improve robustness in the MT scenario is, naturally, the sentence in the source language. Since the target sentence conveys the meaning of the source sentence, s, user corrections should be restricted somehow to the possible translations of it. Hence, we can formulate the problem as,

$$\hat{d} = \operatorname*{argmax}_{d} Pr(d \mid x, p, s) \approx \operatorname*{argmax}_{d} Pr(d \mid p, s) Pr(x \mid d) \tag{7}$$

assuming that $Pr(x \mid d, p, s)$ does not depend on p and s if d is known.

Nevertheless, the relationship between the target and the source sentence in $Pr(d \mid p, s)$ is not trivial to establish. Two possibilities are considered in this work. First, word-based models are the basis for modern statistical MT [21]. Although they cannot provide a good performance when translating complete sentences, they offer a smoothed and reliable probability distribution for word models. In addition, they serve as initialization for the second kind of models considered: phrase-based models [18]. These models improve word-based models since they are able to translate sequences of words (phrases) and constitute the state-of-the-art in MT.

3.4.1. Word-based translation models

Brown et al. [21] approached the problem of MT in Eq. (1) from a statistical point of view as a search problem of a translation t. In this approach a hidden variable a is introduced that represents the alignment between the words in the source and target sentence. Let a be a vector with the length of the target sentence $|t|$[1], where each element a_i represents an index in the source sentence to whom t_i is aligned, i.e., a_i means that t_i is aligned to s_{a_i}. In order to simplify the notation, from now on we will refer to a_i as j so that j indexes source words. Formally, we can model the posterior probability of the target sentence t being a translation of the source sentence s by marginalizing over the set of all possible alignments between the words in t and the words in s,

$$Pr(t \mid s) = \sum_{a} Pr(t, a \mid s) \tag{8}$$

Then, $Pr(t, a \mid s)$ can be decomposed using the chain rule. After taking some strong assumptions, two distributions are obtained. First, the alignment model, $Pr\,(j \mid i, |s|)$, represents the probability of the target word at position i to be aligned with the source word at position j for a source sentence of length $|s|$. Second, the word translation model, $Pr(t_i \mid s_j)$, models the probability of the target word at position i to be a translation of the source word at

[1] We define the length of a sentence as the number of elements in the sentence. The elements are typically words and symbols, but it depends on the tokenization.

position j. The above assumptions are necessary to make model estimation tractable and result in the so-called *model 2* (M2) [21].

In M2, the alignment probability, $Pr\,(j \mid i, |s|)$, can be approximated by the relative frequency of position j in the source sentence to be aligned with position i in the target sentence for a source sentence of length $|s|$. On the other hand, the translation probability, $Pr(t_i \mid s_j)$, can be approximated by a word-to-word statistical dictionary which essentially is the relative frequency of t_i being aligned with s_j. Nonetheless, these frequencies cannot be estimated directly since the real alignments are unknown. Thus, the EM algorithm is needed to reliably estimate these probabilities [21]. *Model 1* (M1) is a particular case of word-based models where the alignment probability is approximated by an uniform probability distribution, $Pr\,(j \mid i, |s|) \approx (|s| + 1)^{-1}$.

Returning to our original problem, we can approach $Pr(d \mid p, s)$ in Eq. (7) with word-based translation models with some assumptions. First, from the prefix p we can obtain the position of the erroneous word to be corrected, $i = |p| + 1$ ignoring the rest of the words in the prefix,

$$Pr(d \mid p, s) \approx Pr(d \mid i, s) \tag{9}$$

Then, we can introduce the alignment between d and the words from the source sentence by summing for every possible position j in s,

$$\begin{aligned} Pr(d \mid i, s) &= \sum_{j=1}^{|s|} Pr(d, j \mid i, s) \\ &= \sum_{j=1}^{|s|} Pr(j \mid i, s) Pr(d \mid j, i, s) \end{aligned} \tag{10}$$

Finally, if we assume, in a similar way to M2, that $Pr(j \mid i, s)$ does not depend on s but on $|s|$, and that $Pr(d \mid j, i, s)$ does not depend on the whole s but just the word aligned to d, s_j with j, then we can approximate Eq. (10) as

$$Pr(d \mid i, s) \approx \sum_{j=1}^{|s|} Pr(j \mid i, |s|) Pr(d \mid s_j) \tag{11}$$

where $Pr(j \mid i, |s|)$ is an M1 or M2 alignment model and $Pr(d \mid s_j)$ is a statistical dictionary.

To clarify the role of the alignments and the dictionary, observe Fig. 4. The source sentence is shown in the middle. Each word has its corresponding position, j, as a subscript. Above each word, there is a list of its most probable translations using the dictionary. Grey levels are proportional to the probability of the dictionary. On the other hand, in the bottom, there is a possible translation, which has an error in position $i = 4$. Below that, the user is trying to correct that mistake by introducing the word . Each link between a source word and the target word in position 4 represents the alignment probability. The stroke boldness is proportional to the M2 alignment probability. Note that for an M1 model, all alignments would have had the same thickness.

If we focus on the possible candidate transcriptions of , we realize that there are two possibilities that could create confusion to the decoder: 'if' as translation of 'si_1' and 'in' as translation of 'en_8' due to the fact that the

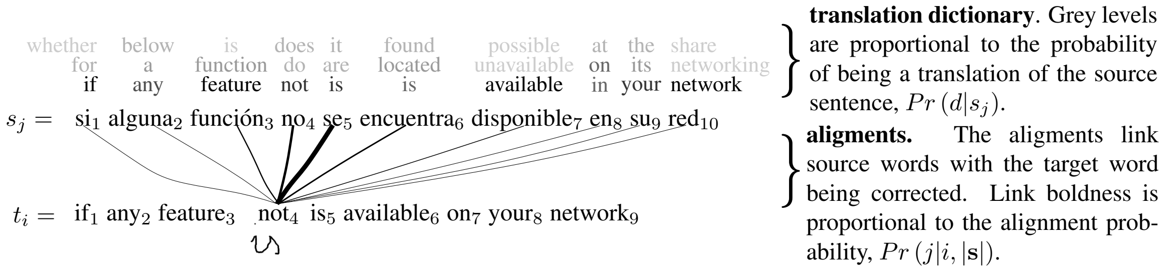


Figure 4: Visualization of alignments and translation dictionary.

strokes for 'is', 'if' and for 'in' can be very similar. Both can compete with the correct transcription 'is'. The first, has a high probability in the dictionary, $Pr(\text{if} \mid \text{si}_1) = 0.88$, whereas $Pr(\text{is} \mid \text{se}_5) = 0.46$, $Pr(\text{is} \mid \text{encuentra}_6) = 0.34$ and $Pr(\text{in} \mid \text{en}_8) = 0.40$. Then, since the M1 model has a uniform alignment probability, it would assign a higher probability to 'if' than to 'is'. However, 'si_1' actually has a lower probability of being aligned with 'not_4'. Therefore, the M2 model is able to solve this shortcoming thanks to the alignments with high probability to the correct words. In this case, $Pr(5 \mid 4, 10) = 0.38$ and $Pr(6 \mid 4, 10) = 0.12$, whereas $Pr(1 \mid 4, 10) = 0.04$.

3.4.2. Phrase-based translation models

Word-based translations provided a basis for MT. However, their performance regarding translation quality was not sufficient. Their limitation resides in that they cannot model properly context information [22]. Phrase-based models aim at reducing this problem by translating phrases (fragments of sentences) instead of single words. These models were popularized by Och and Ney [23], who established the state-of-the-art phrase-based log-linear models. Phrase-based models offer a great opportunity to estimate $Pr(d \mid p, s)$. However, we cannot use these models directly, as we did with word-based models. One limitation of phrase-based models is that their probabilities are 'peaky' and, usually, they cannot model all possible translations. As a result, it is possible that $Pr(d \mid p, s)$ is 0 for a user established prefix like it would be the case in IMT. Then, it is necessary to smooth theses probabilities. For instance, we can generate n-gram-like models from the hypotheses in a word graph (WG) of a MT system [24].

Word graphs contain a set of the most likely translations of the source sentence. They can encode a large number of translations in a more efficient way than n-best lists. Although one may think that the WG could be directly used, there are some details that must be taken into account. First, WGs do not contain all the possible translations since, in practice, many pruning techniques must be used to generate the translations efficiently. Second, phrase-based models are not good dealing with long distance alignments due to the introduction of heuristic length constrains, and thus, WGs do not present sentences with long distance reorderings. In those cases, a user validating a prefix p that is not contained in the WG would obtain a zero probability in $Pr(d \mid p, s)$. Hence, it is interesting to smooth the probability distribution encoded in the WGs. To do so, WGs can be simplified in the way that language modeling is typically approached: we make each word to depend only on the preceding $n - 1$ words instead of depending on the whole

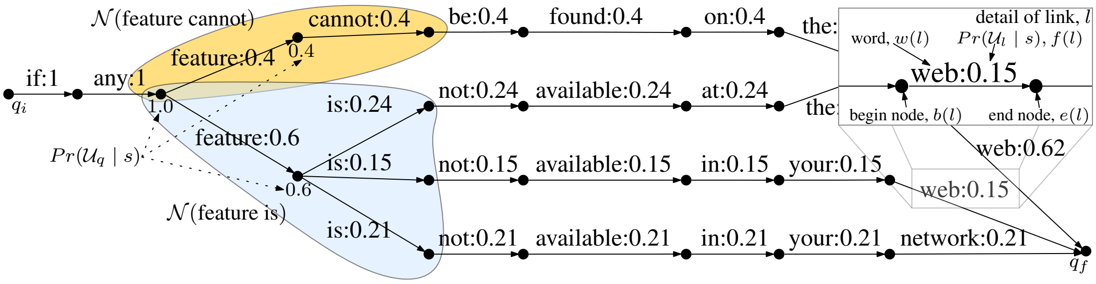


Figure 5: Word graph with posterior probabilities. It represents a subset of hypotheses of the hypothesis space of a state-of-the-art translation model for the source sentence 'si alguna función no se encuentra disponible en su red'. On the left, in blue the set of links considered when computing the average count of the bi-gram 'feature is' whereas in orange the link considered for the bi-gram 'feature cannot'.

prefix. As a result, Eq. (7) can be rewritten as

$$\hat{d} \approx \underset{d}{\operatorname{argmax}}\, Pr(d \mid p_{i-n+1}^{i-1}, s) Pr(x \mid d) \tag{12}$$

where p_{i-n+1}^{i-1} are the words in the prefix from position $i-n+1$ to position $i-1$, i.e., $Pr(d \mid p_{i-n+1}^{i-1}, s)$ only takes into account the latest $n-1$ words from the prefix. Note that $Pr(d \mid p_{i-n+1}^{i-1}, s)$ is very similar to a n-gram language model except for the dependency on s. We used a similar approach for dictation of handwritten historical documents [25, 26] and speech interaction to IMT [13]. Khadivi and Ney [14] presented a closely related approach for generating n-gram-like models from n-bests lists instead of WGs. The advantage of the n-gram-like prefix modeling assumption is that the models only take into account a limited size of the history, and thus, can provide a smoother probability distribution.

Formally speaking, a WG L is a directed, acyclic, weighted graph with an initial node q_i and a final node q_f. A link l is defined as any edge between two nodes; each link has associated a begin node $b(l)$, an end node $e(l)$, a hypothesized word $w(l)$, and a score $f(l)$; each link can be considered as a hypothesis $w(l)$ between the nodes $b(l)$ and $e(l)$ with score $f(l)$. Any path from q_i to q_f forms a translation hypothesis t. In MT, $f(l)$ is the score of the log-linear phrase based model for that particular link. An example of WG is displayed in Fig. 5.

Let $Pr(\mathcal{U}_q \mid s)$ be the posterior probability of all the paths that use the node q and let $Pr(\mathcal{U}_l \mid s)$ be the posterior probability of all the paths that use the link l. These probabilities can be efficiently computed with a *forward-backward*-based algorithm [27]. Then, the average counts of word sequences for a given source sentence can be estimated efficiently as in [28]. For a given n-gram length:

$$C^*(d_{i-n+1}^{i} \mid s) = \sum_{l_1^n \in \mathcal{N}(d_{i-n+1}^{i})} \frac{\prod_{j=1}^{n} Pr(\mathcal{U}_{l_j} \mid s)}{\prod_{j=2}^{n} Pr(\mathcal{U}_{b(l_j)} \mid s)} \tag{13}$$

where $\mathcal{N}(d_{i-n+1}^{i})$ is the set of all the sequences of concatenated links in L that produce the sequence of words d_{i-n+1}^{i}.

An example of such sets on a simplistic WG is shown in Fig. 5 for the 2-grams ‘feature cannot’ and ‘feature is’. Then, $C^*(\text{feature cannot} \mid s)$ and $C^*(\text{feature is} \mid s)$ can be computed as

$$C^*(\text{feature cannot} \mid s) = \frac{0.4 \cdot 0.4}{0.4} = 0.4$$
$$C^*(\text{feature is} \mid s) = \frac{0.6 \cdot 0.24}{0.6} + \frac{0.6 \cdot 0.15}{0.6} + \frac{0.6 \cdot 0.21}{0.6} = 0.6$$

That is, ‘feature is’ appears 0.6 times in average in the possible set of translation, whereas ‘feature cannot’ only appears 0.4 times. Note that if a sequence of words appears more than once in a sentence, the average counts might exceed 1.

Now, n-gram-like probabilities from the WG with posterior probabilities can be calculated after a proper normalization:

$$Pr(d_i \mid d_{i-n+1}^{i-1}, s) = \frac{C^*(d_{i-n+1}^{i} \mid s)}{C^*(d_{i-n+1}^{i-1} \mid s)} \tag{14}$$

Then, Eq. (14) can be used directly in Eq. (7) to approximate $Pr(d \mid p, s)$. In other words, given a sequence of words d_{i-n+1}^{i}, $Pr(d_i \mid d_{i-n+1}^{i-1}, s)$ can be estimated by summing up the posterior probabilities of all sentences containing the sequence d_{i-n+1}^{i}.

The estimation in Eq. (14) presents the problem that many n-grams are not seen in the WG. Then, they will have zero probability, and the HTR system will fail to recognize them. A common approach is to rely on simpler models to account for unseen events using back-off models [29]. As the estimated counts are not real counts (they vary from 0 to the number of times the n-gram occurs in a sentence), typical discount methods cannot be applied [30]. However, absolute discount can be used [31], which consists in subtracting a constant, ϵ, from C^*.

Furthermore, only words present in the WG are included into the model (which implies a high number of out-of-vocabulary words (OOV), since WGs only contain the words of the most likely hypotheses). The OOV problem is solved by distributing the discounted mass from the unigram among the remaining words of the vocabulary.

Finally, to improve the estimation of unseen events, n-grams from the WG can be interpolated linearly with the standard n-gram model:

$$Pr_\gamma(d \mid p, s) = \gamma Pr(d \mid p, s) + (1-\gamma) Pr(d \mid p) \tag{15}$$

This way, the words that were not used by the MT engine are assigned a meaningful probability.

3.5. *Integrated HTR and IMT decoding*

Previous models assume a two-step process, in which the strokes are first decoded into a word or phrase, and then, the decoded word is used to correct the output of the IMT system. However, this decoding can be performed in an integrated way by marginalizing over every possible decoding d in Eq. (2):

$$\hat{t} = \operatorname*{argmax}_t \sum_d Pr(t, d \mid p, x, s) \tag{16}$$

Note that Eq. (16) sums over all possible values of d, but we also are interested in the result of the decoding. Then, we can decompose Eq. (16) using the chain rule. Approximating the sum by the maximum, and assuming that $Pr(t \mid p, x, d, s)$ does not depend on x if d is known,

$$\hat{t} \approx \underset{t}{\operatorname{argmax}} \max_{d} Pr(d \mid p, x, s) Pr(t \mid p, d, s) \tag{17}$$

where $\hat{d}$ can be obtained as a byproduct of the decoding of $\hat{t}$.

The first term in Eq. (17) can be approximated as in Eq. (3),Eq. (4), Eq. (5), Eq. (11) or Eq. (12). The second term is a prefix conditioned translation model as in Eq. (2). This probability forces d not just to be a good translation of s but to form part of a sentence that is good translation of it. Hence, the decoding of d is benefiting from a new source of information.

4. Experiments

In this section, we present a set of experiments to assess the performance of the MT specific HTR systems described in the above sections. Two kinds of experiments were conducted. First, the word-based experiments assume that the user only writes one word at a time. Second, in the phrase-based experiments the user writes a set of consecutive erroneous words. Additionally, two corpora were generated from the Xerox corpus, one with Spanish phrases from translations of English sentences and the other one with English phrases from translations of Spanish sentences. The details of how the two corpora were generated are given in Sec. 4.3.

4.1. IMT corpus: Xerox

The Xerox corpus, created in the TT2 project [32], was used for the experiments, since it has been extensively used in the literature to evaluate IMT systems. It consists of a collection of technical manuals in English, Spanish, French, and German. The English version is the original document, while the others are professional translations of the original. The English and Spanish versions were used in the experiments. The training data was used to generate the translation models. Examples of sentence pairs are shown in Fig. 6. The corpus consists of $56k$ sentences of training and a development and test sets of $1.1k$ sentences. The development set was used to find the tuning parameters that were used in test. Test perplexities for Spanish and English are 35 and 51, respectively. In addition, the Spanish test set has 0.7% out-of-vocabulary running words, whereas the English test set has 0.6% out-of-vocabulary running words.

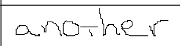
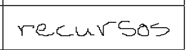
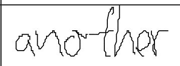
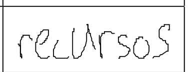
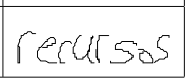
4.2. HTR corpus: Unipen

For on-line HTR, the UNIPEN corpus [33] was used. The training data was composed of symbols, digits and the 1000 most frequent English and Spanish words. The words were generated by concatenating different instances of characters from the same writer, with a total of 17 different writers. Overall, 68 character classes and a total of $23.5k$ unique character instances were used to generate all the $43.8k$ training samples. The feature extraction and modeling

Figure 6: Examples of paired sentences in Spanish and English extracted from the Xerox corpus.

Spanish	English
use este botón para ampliar la búsqueda de dispositivos xerox.	use this button to expand the search for xerox devices.
la búsqueda puede ampliarse para incluir otros nombres de comunidades de snmp que se han agregado a la red.	the search may be expanded to include additional snmp community names that have been added to your network.

Figure 7: Examples of pen strokes from the UNIPEN database used for the simulation of HTR. The words were obtained by concatenating random character instances from the corresponding user.

user	*another*	*recursos*
User 1		
User 2		
User 3		

we used was based on Pastor et al. [2]. Basically, the strokes were preprocessed by eliminating pen-up points and consecutively repeated points. Then, a low pass filter was applied to reduce noise by replacing each point with the mean of its neighbors [3]. From the resulting trajectory, 6 features were extracted:

- the vertical position is normalized by scaling and translating it to $[0, 100]$ keeping aspect ratio.
- the first and second derivatives for the vertical and horizontal position.
- the curvature, which is the inverse of the radius of the curve in each point.

Next, these feature vectors were used to train the morphological models, which were represented by left-to-right continuous density Hidden Markov Models (HMM) [34] with Gaussian mixtures and variable number of states per character. Three users were separated from the training process to produce the words from concatenated characters for the development sets, which were used to find the optimal tuning parameters, and test sets. Examples of generated word in Fig. 7.

4.3. Procedure

For the word-based experiments, the simulation of the user interaction was performed in the following way. First, the publicly available IMT decoder Thot [35] was used to run an off-line simulation for keyboard-based IMT. To do this, we translated each test source sentence. Then, we obtained the longest correct prefix comparing to the reference. Next, we took the word that followed that prefix as the word the user would introduce as a correction. Finally, we used the prefix, and the correct word to obtain a new translation. This was repeated until the reference was obtained. As a result, a list of words that the system failed to predict was obtained. Supposedly, this would be the list of words that the user would correct with handwriting.

Then, from UNIPEN corpus, three writers were selected to simulate the user interaction. For each writer and for each of the words in the list of corrections, the handwritten words were generated by concatenating random character instances from the user's data to form a single stroke. Finally, the generated handwritten words were decoded using the proposed systems with *iAtros* decoder [36]. The 3-gram perplexities for the generated words are 205 and 226 for development and test, respectively, in Spanish, and 242 and 336 for English. It is worthy of note these high perplexities, when for the whole dev and test sets the perplexities are 35 and 51. The word lists were extracted from the erroneous translations that were generated with a decoder using the very same n-grams models used to compute the perplexity. Hence, it is reasonable to assume that if the decoder failed to translate these words it was in part because the language probabilities were low enough, i.e., these probabilities were not well estimated, resulting in a high perplexity. Finally, the number of words in the development sets are 2767 for Spanish and 2398 for English, and in the test sets 2248 and 2102, respectively.

For the phrase-based experiments, the development and test sets were constructed in a similar way. In this case, from the word lists aforementioned, we concatenated the strokes of the words that were consecutive in the original text to form strokes of phrases. For instance, if the MT system had translated 'lista de impresoras' to 'list of printers' when the user preferred 'printer list', in the word-based scenario we would have generated the word 'printer' and the word 'list'. In the phrase-based scenario, as both errors are consecutive, we would have concatenated them in a single phrase as 'printer list'. Figure 8 illustrates a box-and-whisker diagram of the phrase lengths in the different sets. We can observe from the whiskers that the majority of the phrases are less than 3 words for Spanish and 6 for English, whereas for the outliers the lengths reach a maximum at 12 and 18, respectively. Note, however, that the interquartile range is between 2 and 3, meaning that half of the phrases are reasonably short. Finally, the number of phrases in the development sets are 941 for Spanish and 896 for English, and in the test sets 1268 and 1130, respectively.

4.4. Evaluation measures

The performance of the word-based HTR system has been assessed with the *classification error rate* (CER). CER is the ratio between the number of misrecognized words and the total number of words. On the other hand, the phrase-based HTR system has been assessed with the *word error rate* (WER), which can be computed as the number

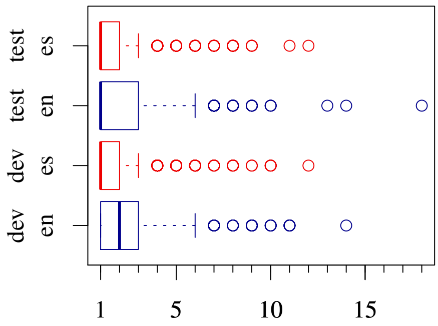


Figure 8: Box-and-whisker plots of the phrase lengths obtained from consecutive errors. In Spanish (es), the system commits typically between 1 and 3 consecutive errors although in rare cases it can commit up to 12 errors. In contrast, up to 6 consecutive errors can be considered normal in English (en). In this case, rare sentences contain at most 18 consecutive errors.

of substitutions, deletions and insertions needed to transform the hypothesis into the reference, normalized by the number of words in the reference. The results present the average error of the three users.

4.5. Results

In this section, we will compare the performance of the proposed systems. In order to make the references easier, we will name the different systems as follows:

HTR. The baseline HTR system as defined in Eq. (3).

ERR. The baseline HTR system after removing the erroneous word, Eq. (4).

n**PREF.** In Eq. (5), the latest n words of validated prefix in the target sentence are taken into account.

M1. In Eq. (11), information regarding the dictionary is used, but the alignment probabilities are uniform.

M2. In Eq. (11), the dictionary and the alignment probabilities are used.

n**WG.** In Eq. (12), the system uses an n-gram that has been extracted from the translation WG.

Furthermore, if the decoding is performed in an integrated way, the system will be marked with **+IMT**. Besides, several of the proposed systems can be combined by linear interpolation as in Eq. (15). In this case, we will use **+** symbol to mark which models were interpolated. The interpolation parameters were obtained in the development set to optimize the accuracy.

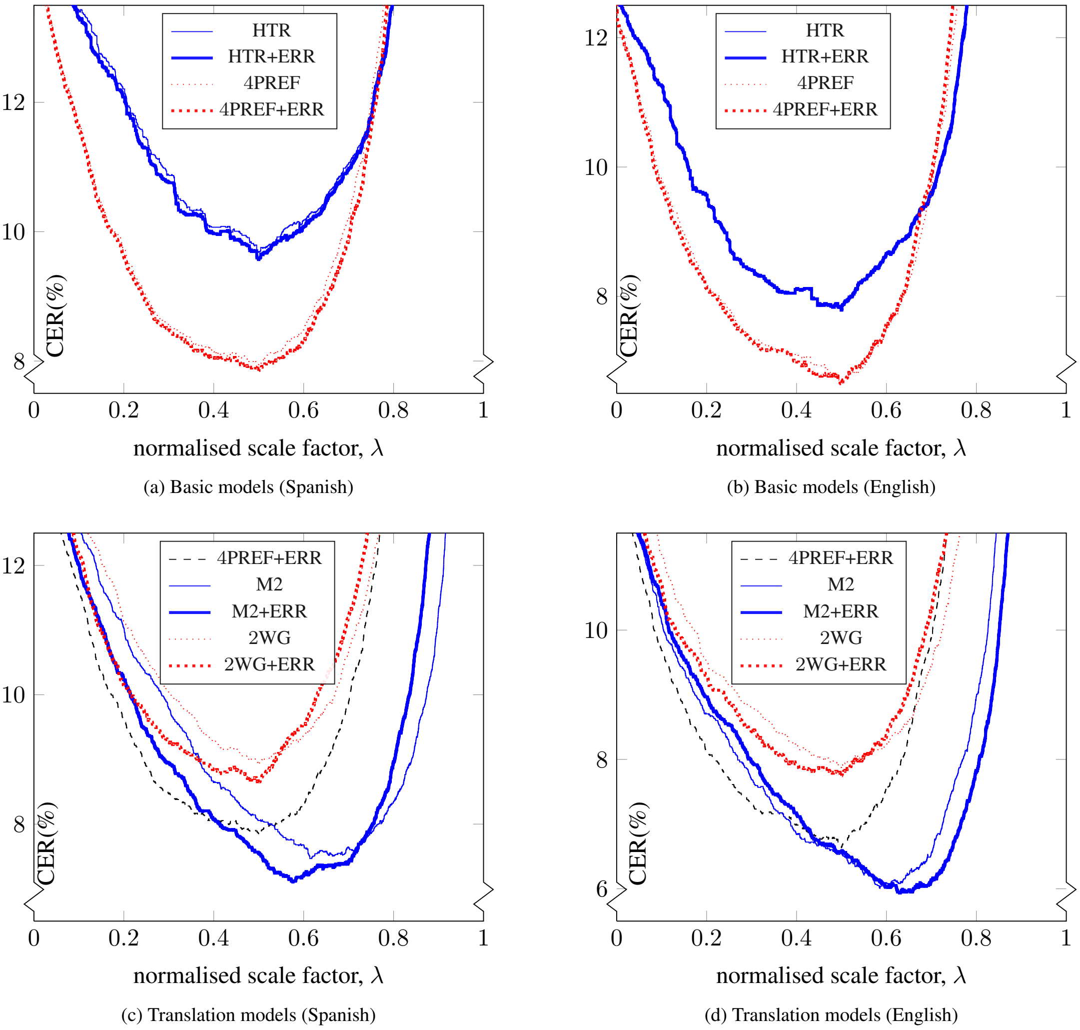
In addition, the proposed language models were encoded as n-grams. The aim of this is two-folded. First, we would like to leverage current HTR systems without custom software modifications. Second, since the new sources of information are added early in the HTR system, we expect to reduce the error cascade produced in post-processing error correcting systems. However, although all the proposed models can be trivially encoded as 1-grams for the case of word-based recognition, some of them cannot be encoded efficiently for n-grams as such and require special search algorithms. As these cases are out of the scope of the current paper, such models will not be evaluated for phrase recognition. Nevertheless, these models could also be applied in a post-processing rescoring stage. For instance, both **M1** and **M2** models can be easily encoded as a 1-gram for word-based recognition. As there is just one possible value for i and s, the 1-gram can be built by computing Eq. (11) for each word of the vocabulary. In contrast, **M2** models cannot be encoded as n-grams for phrase recognition since the probability depends on the position i of the hypothesized word, and then, i should be stored in the search algorithm for every word hypothesis. Luckily, **M1** models assume independence of the position i so they can be encoded as a 1-gram even for the case of phrase recognition.

Finally, as it is typical in modern HTR and IMT models, the different probability distributions must be scaled, particularly the language model. Here, the optimum language model scaling factor, λ, was chosen to optimize the average CER or WER in the development set of the three writers with the downhill simplex method [37]. There were not significant differences in the optimum parameters obtained separately for each writer. Therefore, the estimation of these parameters seems rather robust to the variability of writers.

Regarding the results for the word-based experiments, Fig. 9 shows the test CER for different values of λ for the most relevant systems. First, it must be pointed out that the optimum λ from the development set approximated quite well the test optimum, i.e., the estimation of λ does not present much overfitting. The only exception was the 2**WG** system for which an extra error reduction of 0.5% absolute points could have been achieved.

Second, we should note the effect of adding **ERR** to the system on the error rate. A small improvement can be noticed in Spanish. However, the curves in English overlap. The explanation for this is a bit involving. Note that Spanish is a more inflected language than English. For example, 'both' (in English) can be translated by 'ambos' or 'ambas' (in Spanish), depending on the gender, and having very similar writings. In contrast, 'añade' (in Spanish) can be translated by 'adds' (in English). Thus, we can see how translating from a less inflected language to a more inflected language introduces extra ambiguity. Furthermore, the possible translations of 'both' present also a similar spelling. Conversely, the ambiguity is reduced in the opposite direction. Table 1 shows the 5-best list of the HTR scores for the words 'ambos' and 'adds'. In the first case, 'ambas' and 'ambos' are the two most likely words in the HTR system, which differ in just one character and have similar HTR scores. Now, imagine that the IMT engine mistranslates 'both' to 'ambas', by changing the gender of the word. Then, by saying that *ambas* is not correct with the **ERR** model, we give the system the opportunity to amend the error himself. However, in the English case, none of the words are synonyms of the word to recognize, and thus is more difficult to find the mistranslated word at the top of the n-best list. As a consequence, it is very unlikely that **ERR** achieves much improvement when translating from Spanish to English.

With respect to the n**PREF** models, only 4**PREF** has been displayed in the plots. The improvement over the baseline is consistent and significant. The experiments were run on 2**PREF**, 3**PREF** and 5**PREF** as well. However, only



(a) Basic models (Spanish)

(b) Basic models (English)

(c) Translation models (Spanish)

(d) Translation models (English)

Figure 9: Development CER when modifying the λ scale factor. The x axis represents the variation of the normalized scale factor λ. The y axis shows the classification error rate (CER). At top, the comparison of the basic models described in Sec. 3.1, Sec. 3.2 and Sec. 3.3. At bottom, the most relevant translation models described in Sec. 3.4.

2**PREF** for English performed slightly worse than 4**PREF**. Longer prefixes achieved almost the same performance.

With respect to the systems using the translation models in Fig. 9c and Fig. 9d, we can see that these systems usually outperform the best basic system, 4**PREF+ERR**. The exception for this is 2**WG** for English, which shows a small performance degradation with respect to 4**PREF+ERR**. Still, 2**WG** systems do not seem to improve the basic systems significantly. Although several nWG systems were tested, any of them showed improvements over 2WG. On the other hand, **M2** systems achieve good improvements, although they are simpler than 2**WG**. A reason for that is

both → *ambos*		*añade* → *adds*	
word	HTR score	word	HTR score
ambas	*651.6*	*aids*	*137.9*
ambos	**646.9**	cities	105.6
cambios	390.7	cycles	91.6
amplias	384.1	**adds**	**90.7**
campos	344.4	circles	85.8

Table 1: 5-best list for the words **ambos** and **adds**, which have been misrecognized. The *cursive* word is the word the IMT system mistranslated and the user is amending.

that **M2** models have a smoother distribution probability and n**WG** systems need some sort of hypothesis pruning. In fact, the average number of candidates with probability greater than zero is 292 for **M2** while it is 38 for 4**WG**. **IMT** suffer even more from this problem with 2 candidates average.

A summary of the different alternatives studied for the word-based experiments is shown in Table 2. First, with only the basic information, 4**PREF+ERR** clearly outperforms **HTR**. Second, using translation models we can achieve further improvements. Since **M2** performs much better than **M1** we can deduce that alignment information is crucial for the translation models. On the other hand, n**WG** performance is worse than word-based translation models. As it has been explained before, that might be due to the poorly smoothed probability distribution. Another reason might be that, in the process of obtaining n-gram models, information regarding alignments is lost as a result of the n-gram assumptions. When interpolating with 4**PREF**, **M2** models do not show significant improvements. In fact, for Spanish, the system presents over-fitting, since performance in development improves but in test decreases. However, 4**PREF** smooths 2**WG** distribution achieving close results to word-based models. Next, by introducing **IMT**, small improvements can be obtained. Not surprisingly, **IMT** suffers from the same problems than n**WG**, but even more prominent. Finally, including all systems we can observe the best results overall, except for the over-fitting in the Spanish test set. Thus, 2**WG** seems to contribute slightly to improve the final model accuracy.

Table 3 shows the WER for phrase-based recognition. First, it must be noted that the results for **ERR**, **M2**, and **IMT** are not shown, since they would require a different search engine. In addition, it is worth of mention that the baselines for phrase-based HTR have almost the double error rate than the word-based baselines. This is caused primarily because the segmentation for the words in the phrases are unknown. Then, it is the search algorithm that must find the most likely segmentation. As a result, segmentation errors are propagated to word errors. If we look at the results regarding the n**WG** models, they perform unexpectedly bad when used alone. However, when interpolated with 3**PREF** they show a good improvement. As in word-based recognition, word-based translation models show the

System	Spanish	English
HTR	11.1	9.9
4**PREF+ERR**	9.9	9.5
2**WG+ERR**	9.8	9.4
M1+ERR	9.4	9.0
M2+ERR	**8.6**	7.7
2**WG+**4**PREF+ERR**	9.2	7.9
M2+4**PREF+ERR**	9.0	7.5
2**WG+**4**PREF+ERR+IMT**	9.2	7.9
M2+4**PREF+ERR+IMT**	8.9	7.5
ALL	8.9	**7.4**

Table 2: Summary of the CER results for word-based recognition. The results show various language modeling approaches for the test sets. In **boldface** the best systems.

System	Spanish	English
HTR	16.8	18.6
3**PREF**	16.3	18.0
2**WG**	18.9	19.7
M1	17.0	17.4
2**WG+**3**PREF**	16.2	16.6
M1+3**PREF**	**15.2**	**15.5**
M1+2**WG+**3**PREF**	**15.2**	**15.5**

Table 3: Summary of the WER results for phrase-based recognition. The results show various language modeling approaches for the test sets. In **boldface** the best systems.

best results, especially when interpolated with other models.

To sum up, all the proposed systems significantly outperform the baseline recognizer. Basic models obtain a good improvement over the baseline. However, adding information from the translation may achieve remarkable results. Although more complex translation models suffer from smoothing problems, they can also contribute when interpolated with the rest of the models.

4.6. Error Analysis

An analysis (Table 4) of the results for the best word-based model shows that 49.2% to 54.4% of the recognition errors were produced by punctuation and other symbols. To circumvent this problem, we proposed a contextual menu in [16]. With such menu, errors would have been reduced (best test result) to 4.4% in Spanish and 3.5% in English. Out-of-vocabulary (OOV) words plus zero probability (P0) words (the words for which the decoder assigned zero probability or were pruned out) also summed up a big percentage of the error (40.3% and 28.9%, respectively). Finally, the rest of the errors were mostly due to one-to-three letter words, which can be basically a problem of handwriting morphological modeling.

On the other hand, phrase recognition presents a different error distribution. First, note that two new classes of errors have been introduced: deletions and insertions. The former account for the words in the reference that have been omitted, whereas the latter account for words inserted in the output hypothesis but do not correspond to any word in the reference. Both contribute to generate hypotheses with lengths different to their respective references, since the HMM models is not able to perform an accurate segmentation. Then, as a result, the proportion of recognition errors from the 'others' category increases from 3 to 20. In contrast, the proportion of errors regarding punctuation symbols decreases. Finally, it is to be remarked how the errors for short words have increased, probably because of small insertions or deletions.

4.7. Reducing Effort Correcting HTR errors

In case an HTR error is committed, the user may fall back to the virtual keyboard and type the correct word. The problem with this kind of keyboards is that typing is slow. To minimize this problem, we propose a contextual menu with a list of the n-best candidates (excluding the erroneous word). The aim is to reduce the number of clicks needed to obtain the correct word with respect to a conventional virtual keyboard. As a baseline, for each HTR mistake, we count the number of clicks needed to input the correct word as: one click to pop up the keyboard, plus the number of characters in the word, plus on click to close the keyboard. For the Spanish test set, the average number of clicks per word amounts to 9.3, while for English it is 9.1 for the best word-based models in Table 2. This values can be surprisingly high, since it is known that the average word length is 4.5, i.e. the average number of clicks per word 6.5. However, it must be noticed that longer words are also more difficult to recognize. Thus, the average word length in the erroneous words is higher.

If the contextual menu is used, we count: one click for opening the menu plus one for choosing a word. If the correct word cannot be found in the n-best list, then we add: one count for the keyboard, plus the number of characters, plus a closing click. In Fig. 10, we can see, on the left axis, the CER for a given size of the n-best list. Clearly, the error almost reduces to a quarter, around $n = 5$, with respect to the baseline. Between 10 and 15, the error stabilizes. Note that from 5 to 10 is still a reasonable amount of candidates to be shown in a circular menu. For more than 15,

		word-based		phrase-based	
class	words	es (%)	en (%)	es (%)	en (%)
punct.	., ,, :, ;, *, (,), —	49.2	54.4	14.0	18.6
1-char	a, e, y, o, u	4.1	0.9	8.3	2.3
2-char	of, if, la, by, on, is, …	1.8	7.1	4.4	3.4
3-char	for, off, los, may, …	0.0	4.3	2.1	4.9
numbers	xxvii, xxvi, xxiii, …	2.3	0.9	2.1	2.3
OOV + P0	termina, luz, …	40.3	28.9	20.2	13.6
others	latin, flash, fsma, …	2.3	3.4	20.3	18.6
substitutions		100	100	71.5	63.8
insertions		–	–	3.0	4.6
deletions		–	–	25.5	31.6

Table 4: Detailed analysis of the word-based and phrase-based recognition errors. Five classes have been identified to produce the most amount of recognition errors. The second column shows samples of misrecognized words for these classes. Columns three and four are the percentage of these classes among the total number of misrecognized words for Spanish (es) and English (en), respectively. Columns five and six are the percentages for the phrase-based experiments. In this case, the percentage of substitutions, insertions and deletions is also shown.

the CER almost equals the error for OOV+P0, since they cannot be found in n-best lists. On the right axis, we can observe the average number of clicks per word necessary to correct the mistakes. For $n = 1$ the number of clicks is reduced to 2.0. A trade-off can be found at $n = 7$ with 1.83 (80% relative improvement w.r.t. the baseline) and 1.82 (78% relative improvement), for Spanish and English, whereas the lower bounds are 1.75 and 1.73, respectively.

5. Final Thoughts and Recommendations

While the techniques addressed in this paper have been focused on correcting machine translation output, in reality some of them can be generalized to the correction of other automatically generated outputs. In particular, **ERR** and n**PREF** can be used to improve HTR accuracy for any tasks in which n-grams can be used for language modeling, e.g., [7]. Obviously, **M1** and **M2** are MT specific, but n**WG** can be used for many other structured prediction problems where a word graph can be generated as an output. In fact in a similar way to this work, n**WG** has been successfully used for speech-enabled user interfaces for IMT [13] and for dictation of historical documents [25, 26]. In the same way, integrating HTR with interactive systems is possible for other applications as far as n**WG** is available. Nonetheless, using more specific techniques, such as **M2**, although less general, have proven to be more effective.

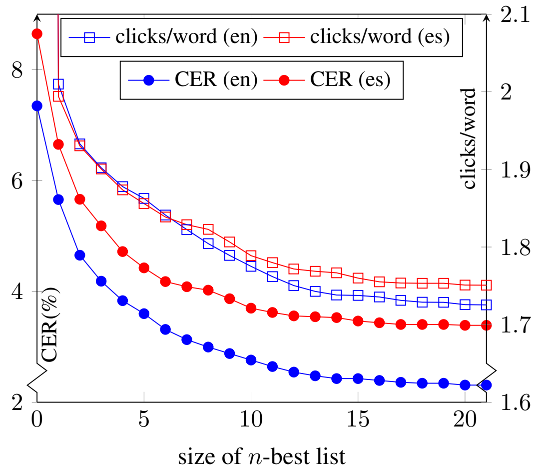


Figure 10: Reduction of CER and number of clicks as a function of the n-best list size.

Finally, we recommend that the integration of contextual information in the decoding is performed in an early stage of the decoding process, avoiding cascade errors. More importantly, if the techniques can be encoded as n-grams, as the techniques presented here, it will allow practitioners to improve their HTR systems without modifying their preferred HTR engines.

6. Conclusions and Future Work

In this paper we have described a task specific on-line HTR system to operate with an IMT application. We have shown that a tight integration of the HTR and IMT decoding process can produce significant HTR error reductions. It is worth of note that all the proposed systems significantly outperform the baseline recognizer. Basic models obtain a good improvement over the baseline. However, translation models achieve remarkable results. Although more complex translation models suffer from smoothing problems, they also contribute when interpolated with the rest of the models. We also have introduced a new method for correcting HTR mistakes that consists on a contextual menu with the n-best candidates. The results show that a list with as few as 7 candidates allows to correct the HTR mistakes with just 1.83 clicks per word.

On the other hand, the analysis of the results has shown two important issues to be tackled. First, the system should be able to decode unknown words since they are a clear limitation to system performance. A solution for this might be to use character language models instead of word language models, a technique that has achieved promising results in other areas. Second, phrase-based models could benefit from better smoothing methods. Alignment information should be also taken into account more explicitly in these models. Furthermore, other alternatives could also be explored, as more advanced word-based translation models (such as HMM, M3, M4 or M5) that cannot be used as n-grams in phrase-based decoding. These models could be used instead in the rescoring of the HTR WGs. Finally, if the rescoring of WGs shows promising results, it would be interesting to directly implement the more advanced MT models into the HTR search algorithm.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o} 287576 (CasMaCat), from the EC (FEDER/FSE), and from the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018) and iTrans2 (TIN2009-14511) project. It is also supported by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/01) and GV/2010/067.

References

[1] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 31:855–868, 2009.

[2] M. Pastor, A. Toselli, and E. Vidal. Writing speed normalization for on-line handwritten text recognition. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition, 2005*, page 1131–1135, 2005.

[3] Stefan Jaeger, Stefan Manke, Jrgen Reichert, and Alex Waibel. Online handwriting recognition: the NPen++ recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–180, 2001.

[4] S. Quiniou, M. Cheriet, and E. Anquetil. Error handling approach using characterization and correction steps for handwritten document analysis. *International Journal on Document Analysis and Recognition*, 15:1–17, 2011.

[5] F. Farooq, D. Jose, and V. Govindaraju. Phrase-based correction model for improving handwriting recognition accuracies. *Pattern Recognition*, 42(12):3271–3277, 2009.

[6] J. Devlin, M. Kamali, K. Subramanian, R. Prasad, and P. Natarajan. Statistical Machine Translation as a Language Model for Handwriting Recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, page 291–296, 2012.

[7] A.H. Toselli, V. Romero, M. Pastor, and E. Vidal. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.

[8] A. H. Toselli, E. Vidal, and F. Casacuberta, editors. *Multimodal Interactive Pattern Recognition and Applications*. Springer, 1st edition edition, 2011.

[9] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. L. Lagarda, H. Ney, J. Tomás, E. Vidal, and J. M. Vilar. Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1):3–28, 2009.

[10] P. Koehn and B. Haddow. Interactive Assistance to Human Translators using Statistical Machine Translation Methods. In *Proceedings of the MT Summit XII*, page 73–80, 2009.

[11] G. Foster, P. Isabelle, and P. Plamondon. Target-Text Mediated Interactive Machine Translation. *Machine Translation*, 12:175–194, 1998.

[12] G. Sanchis-Trilles, D. Ortiz-Martínez, J. Civera, F. Casacuberta, E. Vidal, and H. Hoang. Improving Interactive Machine Translation via Mouse Actions. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, page 485–494, 2008.

[13] V. Alabau, L. Rodríguez-Ruiz, A. Sanchis, P. Martínez-Gómez, and F. Casacuberta. On multimodal interactive machine translation using speech recognition. In *Proceedings of the 13th international conference on multimodal interfaces (ICMI '11)*, page 129–136, 2011.

[14] S. Khadivi and H. Ney. Integration of speech recognition and machine translation in computer-assisted translation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1551–1564, 2008.

[15] E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-Assisted Translation Using Speech Recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941–951, 2006.

[16] V. Alabau, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*, page 389–394, 2011.

[17] V. Alabau, D. Ortiz-Martínez, A. Sanchis, and F. Casacuberta. Multimodal interactive machine translation. In *Proceedings of the International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI '10)*, page 46:1–46:4, 2010.

[18] P. Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

[19] Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Online Learning for Interactive Statistical Machine Translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, page 546–554, 2010.

[20] Frederick Jelinek. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1997.

[21] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The Mathematics of Machine Translation. 19(2):263–311, 1993.

[22] R. Zens, F. Och, and H. Ney. Phrase-based statistical machine translation. page 35–56. 2002.

[23] F. J. Och and H. Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics 2002 (ACL'02)*, page 295–302, July 2002.

[24] N. Ueffing, F.J. Och, and H. Ney. Generation of word graphs in statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP'02)*, page 156–163, 2002.

[25] Vicent Alabau, Carlos D. Martínez-Hinarejos, Verónica Romero, and Antonio L. Lagarda. An iterative multimodal framework for the transcription of handwritten historical documents. *Pattern Recognition Letters*, in press:–, 2012.

[26] V. Alabau, V. Romero, A. L. Lagarda, and C. D. Martínez-Hinarejos. A Multimodal Approach to Dictation of Handwritten Historical Documents. In *Proceedings of the Interspeech 2011*, page 2245–2248, 2011.

[27] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.

[28] W. Campbell and F. Richardson. Discriminative Keyword Selection Using Support Vector Machines. In *Proceedings of the Neural Information Processing Systems 2007 (NIPS'07)*, page 209–216, 2008.

[29] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[30] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, volume 1, page 181–184. IEEE, 1995.

[31] H. Ney, U. Essen, and R. Kneser. On the estimation of 'small' probabilities by leaving-one-out. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 17(12):1202–1212, 1995.

[32] SchulmbergerSema S.A., Celer Soluciones, Instituto Técnico de Informática, R.W.T.H. Aachen - Lehrstuhl für Informatik VI, R.A.L.I. Laboratory - University of Montreal, Société Gamma, and Xerox Research Centre Europe. X.R.C.: TT2. TransType2 - Computer assisted translation. Project technical annex, 2001.

[33] Isabelle Guyon, Lambert Schomaker, Réjean Plamondon, Mark Liberman, and Stan Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Proceedings of the 12th Int'l Conference on Pattern Recognition (ICPR-94)*, page 29–33, 1994.

[34] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[35] D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. Thot: a Toolkit To Train Phrase-based Statistical Translation Models. In *Proceedings of the MT Summit X*, Phuket, Thailand, 2005.

[36] M. Luján-Mares, V. Tamarit, V. Alabau, C.D. Martınez-Hinarejos, M.P. i Gadea, A. Sanchis, and A.H. Toselli. iATROS: A speech and handwritting recognition system. In *Proceedings of the V Jornadas en Tecnologías del Habla (VJTH'2008)*, page 75–78, 2008.

[37] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965.

Vicent Alabau received the Master degree in computer science from the Universidad Politécnica de Valencia, Valencia, Spain, in 2003. Since that moment he has been working in several projects for the “Pattern Recognition and Human Language Technology Group” on speech recognition and translation, machine translation, and multi-modal interactive tools for pattern recognition.

Alberto Sanchis received the Diploma and the Ph.D. degrees in computer science from the Universitat Politècnica de València, Valencia, Spain, in 1997 and 2004, respectively. In 2000, he joined the Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, where he is currently serving as an Associate Professor. His main research interests include pattern recognition and its application to speech and handwriting recognition and machine translation. He is now leading a project on handwriting and speech transcription.

Francisco Casacuberta is a full professor in the Universitat Politècnica de València (UPV), Spain. He is also one of the leaders of the Pattern Recognition and Human Language Technology research group in the Instituto Tecnológico de Informatica. He received his Ph.D. degree from the Universidad de Valencia in 1981. His research interests include the areas of syntactic pattern recognition, statistical pattern recognition, machine translation, speech recognition and machine learning. He is a member of the Spanish AERFAI Society and the IEEE Computer Society.

***Author Biography**

Vicent Alabau received the Master degree in computer science from the Universidad Politécnica de Valencia, Valencia, Spain, in 2003. Since that moment he has been working in several projects for the “Pattern Recognition and Human Language Technology Group” on speech recognition and translation, machine translation, and multimodal interactive tools for pattern recognition.

Alberto Sanchis received the Diploma and the Ph.D. degrees in computer science from the Universitat Politècnica de València, Valencia, Spain, in 1997 and 2004, respectively. In 2000, he joined the Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, where he is currently serving as an Associate Professor. His main research interests include pattern recognition and its application to speech and handwriting recognition and machine translation. He is now leading a project on handwriting and speech transcription.

Francisco Casacuberta is a full professor in the Universitat Politècnica de València (UPV), Spain. He is also one of the leaders of the Pattern Recognition and Human Language Technology research group in the Instituto Tecnológico de Informatica. He received his Ph.D. degree from the Universidad de Valencia in 1981. His research interests include the areas of syntactic pattern recognition, statistical pattern recognition, machine translation, speech recognition and machine learning. He is a member of the Spanish AERFAI Society and the IEEE Computer Society.

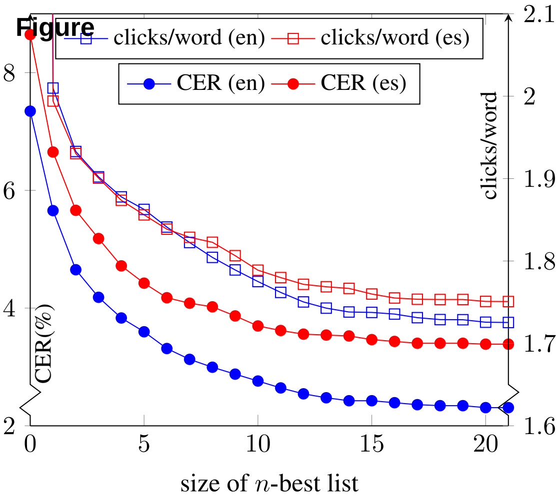


Figure
clicks/word (en)
clicks/word (es)
CER (en)
CER (es)
CER(%)
clicks/word
size of n-best list
8
6
4
2
0
5
10
15
20
2.1
2
1.9
1.8
1.7
1.6