Statistical Approaches to Computer-Assisted Translation

Sergio Barrachina*
Universitat Jaume I

Oliver Bender**
RWTH Aachen

Francisco Casacuberta†
Universitat Politècnica de València

Jorge Civera†
Universitat Politècnica de València

Elsa Cubel†
Universitat Politècnica de València

Shahram Khadivi**
RWTH Aachen

Antonio Lagarda†
Universitat Politècnica de València

Hermann Ney**
RWTH Aachen

Jesús Tomás‡
Universitat Politècnica de València

Enrique Vidal†
Universitat Politècnica de València

Juan-Miguel Vilar§
Universitat Jaume I

Current machine translation (MT) systems are still not perfect. In practice, the output from these systems needs to be edited to correct errors. A way of increasing the productivity of the whole translation process (MT plus human work) is to incorporate the human correction activities within the translation process itself, thereby shifting the MT paradigm to that of computer-assisted translation. This model entails an iterative process in which the human translator activity is included in the loop: In each iteration, a prefix of the translation is validated (accepted or amended) by the human and the system computes its best (or n*-best) translation suffix hypothesis to complete this prefix. A successful framework for MT is the so-called statistical (or pattern recognition) framework. Interestingly, within this framework, the adaptation of MT systems to the interactive scenario affects mainly the search process, allowing a great reuse of successful techniques and models. In this article, alignment templates, phrase-based models, and stochastic finite-state transducers are used to develop computer-assisted translation systems. These systems were assessed in a European project (TransType2) in two real tasks: The translation of printer manuals; manuals and the translation of the* Bulletin of the European Union*. In each task, the following three pairs of languages were involved (in both translation directions): English–Spanish, English–German, and English–French.*

* Departament d'Enginyeria i Ciències dels Computadors, Universitat Jaume I, 12071 Castelló de la Plana, Spain.
** Lehrstuhl für Informatik VI, RWTH Aachen University of Technology, D-52056 Aachen, Germany.
† Institut Tecnològic d'Informàtica, Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, 46071 València, Spain.
‡ Institut Tecnològic d'Informàtica, Departament de Comunicacions, Universitat Politècnica de València, 46071 València, Spain.
§ Departament de Llenguatges i Sistemes Informàtics, Universitat Jaume I, 12071 Castelló de la Plana, Spain.

Submission received: 1 June 2006; revised submission received: 20 September 2007; accepted for publication: 19 December 2007.

1. Introduction to Computer-Assisted Translation

Research in the field of machine translation (MT) aims to develop computer systems which are able to translate text or speech without human intervention. However, present translation technology has not been able to deliver fully automated high-quality translations. Typical solutions to improving the quality of the translations supplied by an MT system require manual post-editing. This serial process prevents the MT system from taking advantage of the knowledge of the human translator, and the human translator cannot take advantage of the adaptive ability of the MT system.

An alternative way to take advantage of the existing MT technologies is to use them in *collaboration* with human translators within a computer-assisted translation (CAT) or *interactive* framework (Isabelle and Church 1997). Historically, CAT and MT have been considered different but close technologies (Kay 1997) and more so for one of the most popular CAT technologies, namely, translation memories (Bowker 2002; Somers 2003). Interactivity in CAT has been explored for a long time. Systems have been designed to interact with human translators in order to solve different types of (lexical, syntactic, or semantic) ambiguities (Slocum 1985; Whitelock et al. 1986). Other interaction strategies have been considered for updating user dictionaries or for searching through dictionaries (Slocum 1985; Whitelock et al. 1986). Specific proposals can be found in Tomita (1985), Zajac (1988), Yamron et al. (1993), and Sen, Zhaoxiong, and Heyan (1997), among others.

An important contribution to CAT technology, carried out within the TransType project, is worth mentioning (Foster, Isabelle, and Plamondon 1997; Langlais, Foster, and Lapalme 2000; Foster 2002; Langlais, Lapalme, and Loranger 2002). It entailed an interesting focus shift in which interaction is directly aimed at the production of the target text, rather than at the disambiguation of the source text, as in earlier interactive systems. The idea proposed in that work was to embed data-driven MT techniques within the interactive translation environment. The hope was to combine the best of both paradigms: CAT, in which the human translator ensures high-quality output, and MT, in which the machine ensures a significant gain in productivity.

Following these TransType ideas, the innovative embedding proposed here consists in using a complete MT system to produce full target sentence hypotheses, or portions thereof, which can be accepted or amended by a human translator. Each correct text segment is then used by the MT system as additional information to achieve further, hopefully improved, suggestions. More specifically, in each iteration, a prefix of the target sentence is somehow fixed by the human translator and, in the next iteration, the system predicts a best (or n-best) translation suffix(es)[1] to complete this prefix. We will refer to this process as **interactive-predictive machine translation** (IPMT).

This approach introduces two important requirements: First, the models have to provide *adequate completions* and, second, this has to happen *efficiently*. Taking these requirements into account, **stochastic finite-state transducers** (SFSTs), **alignment templates** (ATs), and **phrase-based models** (PBMs) are compared in this work. In previous works these models have proven adequate for conventional MT (Vidal 1997; Amengual et al. 2000; Ney et al. 2000; Tomás and Casacuberta 2001; Och and Ney 2003; Casacuberta and Vidal 2004; Och and Ney 2004; Vidal and Casacuberta 2004). This article shows that

1 The terms prefix and suffix are used here to denote any substring at the beginning and end (respectively) of a string of characters (including spaces and punctuation), with no implication of morphological significance as is usually implied by these terms in linguistics.

existing efficient searching algorithms can be adapted in order to provide completions (rather than full translations) also in a very efficient way.

The work presented here has been carried out in the *TransType2* (TT2) project (SchlumbergerSema S.A. et al. 2001), which is considered as a follow-up to the interactive MT concepts introduced in the precursory TransType project cited previously.

We should emphasize the novel contributions of the present work with respect to TransType. First, we show how fully fledged **statistical MT** (SMT) systems can be extended to handle IPMT. In particular, the TT2 systems always produce complete sentence hypotheses on which the human translator can work. This is an important difference to previous work, in which the use of basic MT techniques only allowed the prediction of single tokens (c.f., Section 2.2). Second, using fully fledged SMT systems, we have performed systematic offline experiments to simulate the specific conditions of interactive translation and we report and study the results of these experiments. Thirdly, the IPMT systems presented in this article were successfully used in several field trials with professional translators (Macklovitch, Nguyen, and Silva 2005; Macklovitch 2006).

We should finally mention that the work developed in TT2 has gone beyond conventional keyboard-and-mouse interaction, leading to the development of advanced *multi-modal interfaces*. Speech is the most natural form of human communication and its use as feedback in the IPMT framework has been explored by Vidal et al. (2006). On the other hand, human translators can be faster dictating the translation text rather than typing it, thus it has also been investigated how to improve system performance and usability when the user dictates the translation first and then edits the recognized text (Khadivi, Zolnay, and Ney 2005; Khadivi, Zens, and Ney 2006).

The rest of the article is structured as follows. The next section introduces the general setting for SMT and IPMT. In Section 3, AT, PBM, and SFST are briefly surveyed along with the corresponding learning procedures. In Section 4, general search procedures for the previous models are outlined and a detailed description of the extension of these procedures to IPMT scenarios is presented. Section 5 is devoted to introducing the tasks used for the assessment of the proposal presented in the previous sections: the pairs of languages, corpora, and assessment procedures. The results are reported in Section 6. A discussion of these results and the conclusions which can be drawn from this work are presented in the final section.

2. Statistical Framework

The statistical or pattern recognition framework constitutes a very successful framework for MT. As we will see here, this framework also proves adequate for IPMT.

2.1 Statistical Machine Translation

Assuming that we are given a sentence $\mathbf{s}$ in a source language, the *text-to-text translation* problem can be stated as finding its translation $\mathbf{t}$ in a target language. Using statistical decision theory, the best translation is given by the equation[2]

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

2 We follow the common notation of $\Pr(x)$ for $\Pr(X = x)$ and $\Pr(x|y)$ for $\Pr(X = x|Y = y)$, for any random variables X and Y. Similarly, $\Pr()$ will be used to denote "true" probability functions, and $p()$ or $q()$ will denote model approximations.

Using Bayes's Theorem, we arrive at

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s}|\mathbf{t}) \tag{2}$$

This equation is generally interpreted as follows. The best translation must be a correct sentence in the target language that conveys the meaning of the source sentence. The probability $\Pr(\mathbf{t})$ represents the well-formedness of $\mathbf{t}$ and it is generally called the **language model** probability (n-gram models are usually adopted [Jelinek 1998]). On the other hand, $\Pr(\mathbf{s}|\mathbf{t})$ represents the relationship between the two sentences (the source and its translation). It should be of a high value if the source is a good translation of the target and of a low value otherwise. Note that the translation direction is inverted from what would be normally expected; correspondingly the models built around this equation are often called **inverted translation models** (Brown et al. 1990, 1993). As we will see in Section 3, these models are based on the notion of *alignment*. It is interesting to note that if we had perfect models, the use of Equation (1) would suffice. Given that we have only approximations, the use of Equation (2) allows the language model to correct deficiencies in the translation model.

In practice all of these models (and possibly others) are often combined into a *log-linear model* for $\Pr(\mathbf{t} \mid \mathbf{s})$ (Och and Ney 2004):

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N} \lambda_i \cdot \log f_i(\mathbf{t}, \mathbf{s}) \right\} \tag{3}$$

where $f_i(\mathbf{t}, \mathbf{s})$ can be a model for $\Pr(\mathbf{s}|\mathbf{t})$, a model for $\Pr(\mathbf{t}|\mathbf{s})$, a target language model for $\Pr(\mathbf{t})$, or any model that represents an important feature for the translation. N is the number of models (or features) and λ_i are the weights of the log-linear combination.

When using SFSTs, a different transformation can be used. These transducers have an implicit target language model (which can be obtained from the finite-state transducer by dropping the source symbols of each transition (Vidal et al. 2005)). Therefore, this separation is no longer needed. SFSTs model joint probability distributions; therefore, Equation (1) has to be rewritten as

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{s}, \mathbf{t}) \tag{4}$$

This is the approach followed in GIATI (Casacuberta et al. 2004a; Casacuberta and Vidal 2004), but other models for the joint probability can be adopted.

If the input is a spoken sentence, instead of a written one, the problem becomes more complex; we will not deal with this here. The interested reader may consult Amengual et al. (2000), Ney et al. (2000), or Casacuberta et al. (2004a, 2004b), for instance.

2.2 Statistical Interactive-Predictive Machine Translation

Unfortunately, current models and therefore the systems which can be built from them are still far from perfect. This implies that, in order to achieve good, or even acceptable, translations, manual post-editing is needed. An alternative to this serial approach (first MT, then manual correction) is given by the IPMT paradigm. Under this paradigm, translation is considered as an iterative process where human and computer activity

ITER-0	$(\mathbf{t}_p)$	( )
ITER-1	$(\hat{\mathbf{t}}_s)$	(*Haga clic para cerrar el diálogo de impresión*)
	$(\mathbf{a})$	(Haga clic)
	$(\mathbf{k})$	(**en**)
	$(\mathbf{t}_p)$	(Haga clic en)
ITER-2	$(\hat{\mathbf{t}}_s)$	(*ACEPTAR para cerrar el diálogo de impresión*)
	$(\mathbf{a})$	(*ACEPTAR para cerrar el*)
	$(\mathbf{k})$	(**cuadro**)
	$(\mathbf{t}_p)$	(Haga clic en ACEPTAR para cerrar el cuadro)
FINAL	$(\hat{\mathbf{t}}_s)$	(*de diálogo de impresión*)
	$(\mathbf{a})$	(*de diálogo de impresión*)
	$(\mathbf{k})$	(**#**)
	$(\mathbf{t}_p \equiv \mathbf{t})$	(Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión)

Figure 1
Typical example of IPMT with keyboard interaction. The aim is to translate the English sentence *Click OK to close the print dialog* into Spanish. Each step starts with a previously fixed target language prefix $\mathbf{t}_p$, from which the system suggests a suffix $\hat{\mathbf{t}}_s$. Then the user accepts a part of this suffix (**a**) and types some keystrokes (**k**), possibly in order to amend the remaining part of $\mathbf{t}_s$. This produces a new prefix, composed by the prefix from the previous iteration and the accepted and typed text, (**a**) (**k**), to be used as $\mathbf{t}_p$ in the next step. The process ends when the user enters the special keystroke "**#**". System suggestions are printed in italics and user input in boldface typewriter font. In the final translation **t**, text that has been typed by the user is underlined.

are interwoven. This way, the models take into account both the input sentence and the corrections of the user.

As previously mentioned, this idea was originally proposed in the TransType project (Foster, Isabelle, and Plamondon 1997; Langlais, Foster, and Lapalme 2000; Langlais, Lapalme, and Loranger 2002). In that project, the parts proposed by the systems were produced using a linear combination of a target language model (trigrams) and a lexicon model (so-called IBM-1 or -2) (Langlais, Lapalme, and Loranger 2002). As a result, TransType allowed only single-*token* completions, where a token could be either a word or a short sequence of words from a predefined set of sequences. This proposal was extended to complete full target sentences in the TT2 project, as discussed hereafter.

The approach taken in TT2 is exemplified in Figure 1. Initially, the system provides a possible translation. From this translation, the user marks a prefix as correct and provides, as a hint, the beginning of the rest of the translation. Depending on the system or the user preferences, the hint can be the next word or some letters from it (in the figure, hints are assumed to be words and are referred to as **k**). Let us use $\mathbf{t}_p$ for the prefix validated by the user together with the hint. The system now has to produce (*predict*) a suffix $\mathbf{t}_s$ to complete the translation. The cycle continues with a new validation and hint from the user until the translation is completed. This justifies our choice of the term "interactive-predictive machine translation" for this approach.

The crucial step of the process is the production of the suffix. Again, decision theory tells us to maximize the probability of the suffix given the available information. That is, the best suffix will be

$$\hat{\mathbf{t}}_s = \operatorname*{argmax}_{\mathbf{t}_s} \Pr(\mathbf{t}_s | \mathbf{s}, \mathbf{t}_p) \tag{5}$$

which can be straightforwardly rewritten as

$$\hat{\mathbf{t}}_s = \operatorname*{argmax}_{\mathbf{t}_s} \Pr(\mathbf{t}_p, \mathbf{t}_s | \mathbf{s}) \tag{6}$$

Note that, because $\mathbf{t}_p\mathbf{t}_s = \mathbf{t}$, this equation is very similar to Equation (1). The main difference is that the *argmax* search now is performed over the set of suffixes $\mathbf{t}_s$ that complete $\mathbf{t}_p$ instead of complete sentences ($\mathbf{t}$ in Equation (1)). This implies that we can use the same models if the search procedures are adequately modified (Och, Zens, and Ney 2003).

The situation with respect to finite-state models is similar. Now, Equation (5) is rewritten as

$$\hat{\mathbf{t}}_s = \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_p, \mathbf{t}_s, \mathbf{s}) \tag{7}$$

which allows the use of the same models as in Equation (4) as long as the search procedure is changed appropriately (Cubel et al. 2003, 2004; Civera et al. 2004a, 2004b).

3. Statistical and Finite-State Models

The models used are presented in the following subsections: Section 3.1 for the conditional distribution $\Pr(\mathbf{s}|\mathbf{t})$ in Equation (2) and Section 3.2 for the joint distribution $\Pr(\mathbf{s}, \mathbf{t})$ in Equation (4).

3.1 Statistical Alignment Models

The translation models which Brown et al. (1993) introduced to deal with $\Pr(\mathbf{s}|\mathbf{t})$ in Equation (2) are based on the concept of *alignment* between the components of a pair $(\mathbf{s}, \mathbf{t})$ (thus they are called **statistical alignment models**). Formally, if the number of the source words in $\mathbf{s}$ is J and the number of target words in $\mathbf{t}$ is I, an *alignment* is a function $\mathbf{a} : \{1, ..., J\} \rightarrow \{0, ..., I\}$. The image of j by $\mathbf{a}$ will be denoted as $\mathbf{a}_j$, in which the particular case $\mathbf{a}_j = 0$ means that the position j in $\mathbf{s}$ is not aligned with any position of $\mathbf{t}$. By introducing the alignment as a hidden variable in $\Pr(\mathbf{s}|\mathbf{t})$,

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

The alignment that maximizes $\Pr(\mathbf{s}, \mathbf{a}|\mathbf{t})$ is shown to be very useful in practice for training and for searching.

Different approaches have been proposed for modeling $\Pr(\mathbf{s}, \mathbf{a}|\mathbf{t})$ in Equation (8): Zero-order models such as *model 1*, *model 2*, and *model 3* (Brown et al. 1993) and the first-order models such as *model 4*, *model 5* (Brown et al. 1993), *hidden Markov model* (Ney et al. 2000), and *model 6* (Och and Ney 2003).

In all these models, single words are taken into account. Moreover, in practice the summation operator is replaced with the maximization operator, which in turn reduces the contribution of each individual source word in generating a target word. On the other hand, modeling word sequences rather than single words in both the alignment and lexicon models cause significant improvement in translation quality (Och and Ney

2004). In this work, we use two closely related models: ATs (Och and Ney 2004) and PBMs (Tomás and Casacuberta 2001; Koehn, Och, and Marcu 2003; Zens and Ney 2004). Both models are based on bilingual phrases[3] (pairs of segments or word sequences) in which all words within the source-language phrase are aligned only to words of the target-language phrase and vice versa. Note that at least one word in the source-language phrase must be aligned to one word of the target-language phrase, that is, there are no empty phrases similar to the empty word of the word-based models. In addition, no gaps and no overlaps between phrases are allowed.

We introduce some notation to deal with phrases. As before, $\mathbf{s}$ denotes a source-language sentence; $\widetilde{\mathbf{s}}$ denotes a generic phrase in $\mathbf{s}$, and $\widetilde{\mathbf{s}}_k$ the kth phrase in $\mathbf{s}$. $\mathbf{s}_j$ denotes the jth source word in $\mathbf{s}$; $\mathbf{s}_j^{j'}$ denotes the contiguous sequence of words in s beginning at position j and ending at position j' (inclusive); obviously, if $\mathbf{s}$ has J words, $\mathbf{s}_1^J$ denotes the whole sentence $\mathbf{s}$. An analogous notation is used for target words, phrases, and sequences in target sentence $\mathbf{t}$.

3.1.1 Alignment Templates. The ATs are based on the bilingual phrases but they are generalized by replacing words with word classes and by storing the alignment information for each phrase pair. Formally, an AT Z is a triple $(S, T, \widetilde{\mathbf{a}})$, where S and T are a source class sequence and a target class sequence, respectively, and $\widetilde{\mathbf{a}}$ is an alignment from the set of positions in S to the set of positions in T.[4] Mapping of source and target words to bilingual word classes is automatically trained using the method described by Och (1999). The method is actually an unsupervised clustering method which partitions the source and target vocabularies, so that assigning words to classes is a deterministic operation. It is also possible to employ parts-of-speech or semantic categories instead of the unsupervised clustering method used here. More details can be found in Och (1999) and Och and Ney (2004). However, it should be mentioned that the whole AT approach (and similar PBM approaches as they are now called) is independent of the word clustering concept. In particular, for large training corpora, omitting the word clustering in the AT system does not much affect the translation accuracy.

To arrive at our translation model, we first perform a segmentation of the source and target sentences into K "blocks" $d_k \equiv (i_k; b_k, j_k)$ ($i_k \in \{1, \ldots, I\}$ and $j_k, b_k \in \{1, \ldots, J\}$ for $1 \leq k \leq K$). For a given sentence pair $(\mathbf{s}_1^J, \mathbf{t}_1^I)$, the kth bilingual segment $(\widetilde{\mathbf{s}}_k, \widetilde{\mathbf{t}}_k)$ is $(\mathbf{s}_{b_{k-1}+1}^{j_k}, \mathbf{t}_{i_{k-1}+1}^{i_k})$ (Och and Ney 2003). The AT $Z_k = (S_k, T_k, \widetilde{\mathbf{a}}_k)$ associated with the kth bilingual segment is: S_k the sequence of word classes in $\widetilde{\mathbf{s}}_k$; T_k the sequence of word classes in $\widetilde{\mathbf{t}}_k$, and $\widetilde{\mathbf{a}}_k$ the alignment between positions in a source class sequence S and positions in a target class sequence T.

For translating a given source sentence $\mathbf{s}$ we use the following decision rule as an approximation to Equation (1):

$$(\hat{I}, \hat{\mathbf{t}}_1^{\hat{I}}) = \underset{I, \mathbf{t}_1^I}{\operatorname{argmax}} \left\{ \max_{K, d_1^K, \widetilde{\mathbf{a}}_1^K} \log P_{AT}(\mathbf{s}_1^J, \mathbf{t}_1^I; d_1^K, \widetilde{\mathbf{a}}_1^K) \right\} \tag{9}$$

3 Although the term "phrase" has a more restricted meaning, in this article it refers to a word sequence.

4 Note that the phrases in an AT are sequences of word classes rather than words, which motivates the use of a different notation.

We use a log-linear model combination:

$$\begin{aligned}
&\log P_{AT}(\mathbf{s}_1^J, \mathbf{t}_1^I; d_1^K, \widetilde{\mathbf{a}}_1^K) = \\
&\sum_{i=1}^{I} \left[\lambda_1 + \lambda_2 \cdot \log p(\mathbf{t}_i | \mathbf{t}_{i-2}^{i-1}) + \lambda_3 \cdot \log p(T_i | T_{i-4}^{i-1}) \right] + \\
&\sum_{k=1}^{K} [\ \lambda_4 + \lambda_5 \cdot \log q(b_k | j_{k-1}) + \lambda_6 \cdot \log p(T_k, \widetilde{\mathbf{a}}_k | S_k) + \\
&\qquad \sum_{i=i_{k-1}+1}^{i_k} \lambda_7 \cdot \log p(\mathbf{t}_i | \widetilde{\mathbf{s}}_k, \widetilde{\mathbf{a}}_k)\ ]
\end{aligned} \tag{10}$$

with weights $\lambda_i, i = 1, \cdots, 7$. The weights λ_1 and λ_4 play a special role and are used to control the number I of words and number K of segments for the target sentence to be generated, respectively. The log-linear combination uses the following set of models:

- $p(\mathbf{t}_i | \mathbf{t}_{i-2}^{i-1})$: Word-based trigram language model
- $p(T_i | T_{i-4}^{i-1})$: Class-based five-gram language model
- $p(T_k, \widetilde{\mathbf{a}}_k | S_k)$: AT at class level, model parameters are estimated directly from frequency counts in a training corpus
- $p(\mathbf{t}_i | \widetilde{\mathbf{s}}_k, \widetilde{\mathbf{a}}_k)$: Single word model based on a statistical dictionary and $\widetilde{\mathbf{a}}_k$. As in the preceding model, the model parameters are estimated by using frequency counts
- $q(b_k | j_{k-1}) = e^{|b_k - j_{k-1} + 1|}$: Re-ordering model using absolute j distance of the phrases.

As can be observed, all models are implemented as feature functions which depend on the source and the target language sentences, as well as on the two hidden variables $(\widetilde{\mathbf{a}}_1^K, b_1^K)$. Other feature functions can be added to this sort of model as needed. For a more detailed description the reader is referred to Och and Ney (2004).

Learning alignment templates. To learn the probability of applying an AT, $p(Z = (S, T, \widetilde{\mathbf{a}}) | \widetilde{\mathbf{s}})$, all bilingual phrases that are consistent with the segmentation are extracted from the training corpus together with the alignment within these phrases. Thus, we obtain a count $N(Z)$ of how often an AT occurred in the aligned training corpus. Using the relative frequency

$$p(Z) = (S, T, \widetilde{\mathbf{a}}) | \widetilde{\mathbf{s}}) = \frac{N(Z) \cdot \delta(S, C(\widetilde{\mathbf{s}}))}{N(C(\widetilde{\mathbf{s}}))} \tag{11}$$

we estimate the probability of applying an AT Z to translate the source language phrase $\widetilde{\mathbf{s}}$, in which δ is Kronecker's delta function. The class function C maps words onto their

classes. To reduce the memory requirements, only probabilities for phrases up to a maximal length are estimated, and phrases with a probability estimate below a certain threshold are discarded.

The weights λ_i in Equation (10) are usually estimated using held-out data with respect to the automatic evaluation metric employed using the downhill simplex algorithm from Press et al. (2002).

3.1.2 Phrase-Based Models. A simple alternative to AT has been introduced in recent works: The PBM approach (Tomás and Casacuberta 2001; Marcu and Wong 2002; Zens, Och, and Ney 2002; Tomás and Casacuberta 2003; Zens and Ney 2004). These methods learn the probability that a sequence of contiguous words—the *source phrase*—(as a whole unit) in a source sentence is a translation of another sequence of contiguous words—the *target phrase*—(as a whole unit) in the target sentence. In this case, the statistical dictionaries of single word pairs are substituted by statistical dictionaries of *bilingual phrases* or *bilingual segments*. These models are simpler than ATs, because no alignments are assumed between word positions inside a bilingual segment and word classes are not used in the definition of a bilingual phrase.

The simplest formulation is for *monotone* PBMs (Tomás and Casacuberta 2007), assuming a uniform distribution of the possible segmentations of the source and of the target sentences. In this case, the approximation to Equation (1) is:

$$(\hat{I}, \hat{\mathbf{t}}_1^{\hat{I}}) = \underset{I, \mathbf{t}_1^I}{\operatorname{argmax}} \left\{ \max_{K, d_1^K} \log P_{PBM}(\mathbf{s}_1^J, \mathbf{t}_1^I; d_1^K) \right\} \tag{12}$$

In our implementation of this approach, we have also adopted a log-linear model

$$\begin{aligned} &\log P_{PBM}(\mathbf{s}_1^J, \mathbf{t}_1^I; d_1^K) = \\ &\sum_{i=1}^{I} \left[\lambda_1 + \lambda_2 \cdot \log p(\mathbf{t}_i | \mathbf{t}_{i-2}^{i-1}) + \lambda_3 \cdot \log p(T_i | T_{i-4}^{i-1}) \right] + \\ &\sum_{k=1}^{K} \left[\lambda_4 + \lambda_5 \cdot \log p(\widetilde{\mathbf{t}}_k | \widetilde{\mathbf{s}}_k) \right] \end{aligned} \tag{13}$$

with weights $\lambda_i, i = 1, \cdots, 5$. The weights λ_1 and λ_4 play a special role and are used to control the number I of words and number K of segments for the target sentence to be generated, respectively. The log-linear combination uses the following set of models:

- $p(\mathbf{t}_i | \mathbf{t}_{i-2}^{i-1})$: Word-based trigram language model
- $p(T_i | T_{i-4}^{i-1})$: Class-based five-gram language model
- $p(\widetilde{\mathbf{t}}_k | \widetilde{\mathbf{s}}_k)$: Statistical dictionary of bilingual phrases.

If segment re-ordering is desired (*non-monotone* models), the probability of phrase-alignment q can be introduced (a first-order distortion model is assumed):

$$\log P_{PBM}(\mathbf{s}_1^J, \mathbf{t}_1^I; d_1^K) = \sum_{i=1}^{I} \left[\lambda_1 + \lambda_2 \cdot \log p(\mathbf{t}_i|\mathbf{t}_{i-2}^{i-1}) + \lambda_3 \cdot \log p(T_i|T_{i-4}^{i-1})\right] + \sum_{k=1}^{K} \left[\lambda_4 + \lambda_5 \cdot \log p(\widetilde{\mathbf{t}}_k|\widetilde{\mathbf{s}}_k) + \lambda_6 \cdot \log q(b_k|j_{k-1})\right] \quad (14)$$

with the additional model q, similar to the one used for AT.

Learning phrase-based alignment models. The parameters of each model and the weights λ_i in Equations (13) and (14) have to be estimated. There are different approaches to estimating the parameters of each model (Tomás and Casacuberta 2007). Some of these techniques correspond to a direct learning of the parameters from a sentence-aligned corpus using a maximum likelihood approach (Tomás and Casacuberta 2001; Marcu and Wong 2002). Other techniques are heuristics based on the previous computation of word alignments in the training corpus (Zens, Och, and Ney 2002; Koehn, Och, and Marcu 2003). On the other hand, as for AT, the weights λ_i in Equation (13) are usually optimized using held-out data.

3.2 Stochastic Finite-State Transducers

SFSTs constitute an important framework in syntactic pattern recognition and natural language processing. The simplicity of finite-state models has given rise to some concerns about their applicability to real tasks. Specifically in the field of language translation, it is often argued that *natural languages* are so complex that these simple models are never able to cope with the required source-target mappings. However, one should take into account that the complexity of the *mapping* between the source and target domains of a transducer is not always directly related to the complexity of the domains themselves. Instead, a key factor is the degree of *monotonicity* or *sequentiality* between source and target subsequences of these domains (Casacuberta, Vidal, and Picó 2005). Finite-state transducers have been shown to be adequate to handle complex mappings efficiently (Berstel 1979) and SFSTs are closely related to monotone PBMs.

In Equation (4), $\Pr(\mathbf{s}, \mathbf{t})$ can be modeled by an SFST T, which is defined as a tuple $\langle \Sigma, \Delta, Q, q_0, p, f \rangle$, where Σ is a finite set of *source symbols*, Δ is a finite set of *target symbols* ($\Sigma \cap \Delta = \emptyset$), Q is a finite set of *states*, q_0 is the initial state, p and f are two functions $p : Q \times \Sigma \times \Delta^\star \times Q \to [0, 1]$ (for the *probabilities of transitions*) and $f : Q \to [0, 1]$ (for the *probabilities of final states*) that satisfy $\forall q \in Q$:

$$f(q) \quad + \sum_{(s,\widetilde{\mathbf{t}},q') \in \Sigma \times \Delta^\star \times Q} p(q, s, \widetilde{\mathbf{t}}, q') = 1 \quad (15)$$

Given T, a *path* with J transitions associated with the *translation pair* $(\mathbf{s}, \mathbf{t}) \in \Sigma^* \times \Delta^*$ is a sequence of transitions $\phi = (q_0, \mathrm{s}_1, \tilde{\mathrm{t}}_1, q_1)\ (q_1, \mathrm{s}_2, \tilde{\mathrm{t}}_2, q_2)\ (q_2, \mathrm{s}_3, \tilde{\mathrm{t}}_3, q_3) \ldots (q_{J-1}, \mathrm{s}_J, \tilde{\mathrm{t}}_J, q_J)$, such that $\mathrm{s}_1\ \mathrm{s}_2 \ldots \mathrm{s}_J = \mathbf{s}$ and $\tilde{\mathrm{t}}_1\, \tilde{\mathrm{t}}_2 \ldots \tilde{\mathrm{t}}_J = \mathbf{t}$. The probability of a path is

the product of its transition probabilities, times the final-state probability of the last state in the path:

$$P_T(\phi) = \prod_{j=1}^{J} p(q_{j-1}, \mathrm{s}_j, \tilde{\mathrm{t}}_j, q_j) \cdot f(q_J) \tag{16}$$

The *probability of a translation pair* $(\mathbf{s}, \mathbf{t})$ according to T is then defined as the sum of the probabilities of all the paths associated with $(\mathbf{s}, \mathbf{t})$:

$$P_T(\mathbf{s}, \mathbf{t}) = \sum_{\phi} P_T(\phi) \tag{17}$$

Learning finite-state transducers. There are different families of techniques to train an SFST from a parallel corpus of source–target sentences (Casacuberta and Vidal 2007). One of the techniques that has been adopted in this work is the **grammatical inference and alignments for transducer inference** (GIATI) technique. This technique is in the category of *hybrid methods* which use statistical techniques to guide the SFST structure learning and simultaneously train the associated probabilities.

Given a finite sample of string pairs, the inference of SFSTs using the GIATI technique is performed as follows (Casacuberta and Vidal 2004; Casacuberta, Vidal, and Picó 2005): i) *Building training strings*: Each training pair is transformed into a single string from an extended alphabet to obtain a new sample of strings. ii) *Inferring a (stochastic) regular grammar*. Typically, a smoothed n-gram is inferred from the sample of strings obtained in the previous step. iii) *Transforming the inferred regular grammar into a transducer*: The symbols associated with the grammar rules are converted back into input/output symbols, thereby transforming the grammar inferred in the previous step into a transducer. The transformation of a parallel corpus into a string corpus is performed using statistical alignments. These alignments are obtained using the GIZA++ software (Och and Ney 2003).

4. Searching

Searching is an important computational problem in SMT. Algorithmic solutions developed for SMT can be adapted to the IPMT framework. The main general search procedures for each model in Section 3 are presented in the following subsections, each followed by a detailed description of the necessary adaptations to the interactive framework.

4.1 Searching with Alignment Templates

In offline MT, the generation of the best translation for a given source sentence $\mathbf{s}$ is carried out by producing the target sentence in left-to-right order using the model of Equation (10). At each step of the generation algorithm we maintain a set of active hypotheses and choose one of them for extension. A word of the target language is then added to the chosen hypothesis and its costs get updated. This kind of generation fits nicely into a **dynamic programming** (DP) framework, as hypotheses which are indistinguishable by both language and translation models (and that have covered the same source positions) can be recombined. Because the DP search space grows

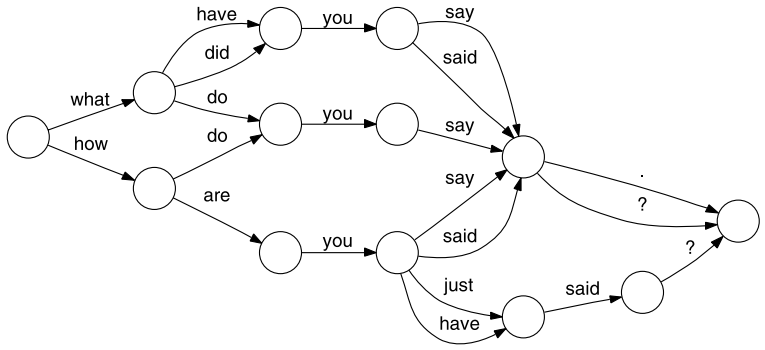

Figure 2
Example of a word graph for the source German sentence *was hast du gesagt?* (English reference translation: "what did you say?").

exponentially with the size of the input, standard DP search is prohibitive, and we resort to a beam-search heuristic.

4.1.1 Adaptation to the Interactive-Predictive Scenario. The most important modification is to rely on a word graph that represents possible translations of the given source sentence. This word graph is generated once for each source sentence. During the process of human–machine interaction the system makes use of this word graph in order to complete the prefixes accepted by the human translator. In other words, after the human translator has accepted a prefix string, the system finds the best path in the word graph associated with this prefix string so that it is able to complete the target sentence. Using the word graph in such a way, the system is able to interact with the human translator in a time efficient way. In Och, Zens, and Ney (2003), an efficient algorithm for interactive generation using word graphs was presented. A word graph is a weighted directed acyclic graph, in which each node represents a partial translation hypothesis and each edge is labeled with a word of the target sentence and is weighted according to the language and translation model scores. In Ueffing, Och, and Ney (2002), the authors give a more detailed description of word graphs and show how they can be easily produced as a by-product of the search process. An example of a word graph is shown in Figure 2.

The computational cost of this approach is much lower, as the whole search for the translation must be carried out only once, and the generated word graph can be reused for further completion requests.

For a fixed source sentence, if no pruning is applied in the production of the word graph, it represents all possible sequences of target words for which the posterior probability is greater than zero, according to the models used. However, because of the pruning generally needed to render the problem computationally feasible, the resulting word graph only represents a subset of the possible translations. Therefore, it may happen that the user sets prefixes which cannot be found in the word graph. To circumvent this problem some heuristics need to be implemented.

First, we look for the node with minimum edit distance to the prefix except for its last (partial) word.[5] Then we select the completion path which starts with the last

5 The edit distance concept for finding the prefix string in a word graph could be refined by casting the edit distance operations into a suitable probabilistic model.

(partial) word of the prefix and has the best backward score—this is the score associated with a path going from the node to the final node. Now, because the original word graph may not be compatible with the new information provided by the prefix, it might be impossible to find a completion in this word graph due to incompatibility with the last (partial) word in the prefix. This problem can be solved to a certain degree by searching for a completion of the last word with the highest probability using only the language model. This supplementary heuristic to the usual search increases the performance of the system, because some of the rejected words in the pruning process can be recovered.

A desirable feature of an IPMT system is the possibility of producing a list of alternative target suffixes, instead of only one. This feature can be easily added by computing the n-best hypotheses. Of course, these n-best hypotheses do not refer to the whole target sentence, but only to the suffixes. However, the problem is that in many cases the sentence hypotheses in the n-best list differ in only one or two words. Therefore, we introduce the additional requirement that the first four words of the n-best hypotheses must be different.

4.2 Searching with Phrase-Based Models

The generation of the best translation with PBMs is similar to the one described in the previous section. Each hypothesis is composed of a prefix of the target sentence, a subset of source positions that are aligned with the positions of the prefix of the target sentence, and a score. In this case, we adopted an extension of the *best-first strategy* where the hypotheses are stored in several sorted lists, depending on which words in the source sentence have been translated. This strategy is related to the well-known *multi-stack-decoding* algorithm (Berger et al. 1996; Tomás and Casacuberta 2004). In each iteration, the algorithm extends the best hypothesis from each available list.

While target words are always generated from left to right, there are two alternatives in the source word extraction: Monotone search, which takes the source words from left to right, and non-monotone search, which can take source words in any order.

4.2.1 Adaptation to the Interactive-Predictive Scenario. Only a simple modification of this search algorithm is necessary: If the new extended hypothesis is not compatible with the fixed target prefix, $\mathbf{t}_p$, then this hypothesis is not considered. This compatibility is verified at the character level; therefore the user does not need to type the whole target word at the end of the target prefix.

In the interactive scenario, speed is a critical aspect. In the PBM approach, monotone search is much faster than non-monotone search in the tasks which are considered in this work (Tomás and Casacuberta 2006). However, monotone search presents a problem for interactive operation: If a user introduces a prefix that cannot be obtained in a monotone way from the source, the search algorithm is not able to complete this prefix. In order to solve this problem without losing computational efficiency, we use the following approach: Non-monotone search is used for target prefixes, whereas completions (suffixes) are generated using monotone search.

As for AT models, a list of target suffixes can also be produced. This list can be obtained easily by keeping the n-best hypotheses in each sorted list. To avoid generating very similar hypotheses in the n-best list, we apply the following procedure: Starting from the n-best list resulting from the normal search, we first add hypotheses obtained

by translating a single untranslated word from the source, along with hypotheses consisting of a single high-probability word according to the target language model; we then re-order the hypotheses, maximizing the diversity at the beginning of the suffixes, and keep only the n first hypotheses in the re-ordered list.

4.3 Searching with Stochastic Finite-State Transducers

As discussed by Picó and Casacuberta (2001), the computation of Equation (4) for SFSTs under a maximum approximation (i.e., using maximization in Equation (17) instead of the sum) amounts to a conventional Viterbi search. The algorithm finds the most probable path among those paths in the SFST which are compatible with the source sentence $\mathbf{s}$. The corresponding translation, $\tilde{\mathbf{t}}$, is simply obtained by concatenating the target strings of the edges of this path.

4.3.1 Adaptation to the Interactive-Predictive Scenario. Here, Equation (7) is used wherein the optimization is performed over the set of target suffixes (completions) rather than the set of complete target sentences. To solve this maximization problem, an approach similar to that proposed for AT in Section 4.1 has been adopted.

First, given the source sentence, a word graph is extracted from the SFST. In this case, the word graph is just (a pruned version of) the Viterbi search trellis obtained when translating the whole source sentence. The main difference between the word graphs generated with ATs and SFSTs is how the nodes and edges are defined in each case. On the one hand, the nodes are defined as partial hypotheses of the search procedure in the AT approach, whereas the nodes in the case of SFSTs can be directly mapped into states in the SFST representing a joint (source word/target string) language model. On the other hand, the scores associated with the edges in the AT approach are computed from a combination of the language and translation models, whereas in the case of SFSTs these scores simply come from the joint language model estimated by the GIATI technique.

Once the word graph has been generated, the search for the most probable completion as stated in Equation (6) is carried out in two steps, in a similar way to that explained for the AT approach. In this case, the computation entailed by both the edit-distance (prefix error-correcting) and the remaining search is significantly accelerated by visiting the nodes in topological order and by the incorporation of the beam-search technique (Amengual and Vidal 1998). Moreover, the error-correcting algorithm takes advantage of the incremental way in which the user prefix is generated, parsing only the new suffix appended by the user in the last interaction.

It may be the case that a user prefix ends in an incomplete word during the interactive translation process. Therefore, it is necessary to start the translation completion with a word whose prefix matches this unfinished word. The proposed algorithm thus searches for such a word. First, it considers the target words of the edges leaving the nodes returned by the error-correcting algorithm. If this initial search fails, then a matching word is looked up in the word-graph vocabulary. Finally, as a last resort, the whole transducer vocabulary is taken into consideration to find a matching word; otherwise this incomplete word is treated as an entire word.

This error-correcting algorithm returns a set of nodes from which the best completion would be selected according to the best backward score. Moreover, n-best completions can also be produced. Among many weighted-graph n-best path algorithms which are available, the **recursive enumeration algorithm** presented in Jiménez and

Marzal (1999) was adopted for its simplicity in calculating best paths *on demand* and its smooth integration with the error-correcting algorithm.

5. Experimental Framework

The models and search procedures introduced in the previous sections were assessed through a series of IPMT experiments with different corpora. These corpora, along with the corresponding pre- and post-processing and assessment procedures, are presented in this section.

5.1 Pre- and Post-Processing

Usually, MT models are trained on a pre-processed version of an original corpus. Pre-processing provides a simpler representation of the training corpus which makes token or word forms more homogeneous. In this way automatic training of the MT models is boosted, and the amount of computation decreases.

The pre-processing steps are: tokenization, removing unnecessary case information, and tagging some special tokens like numerical sequences, e-mail addresses, and URLs ("categorization"). In translation from a source language to a target language, there are some words which are translated identically (because they have the same spelling in both languages). Therefore, we identify them in the corpus and replace them with some generic tags to help the translation system.

Post-processing takes place after the translation in order to hide the internal representation of the text from the user. Thus, the user will only work with an output which is very similar to human-generated texts. In detail, the post-processing steps are: de-tokenization, true-casing, and replacing the tags with their corresponding words.

In an IPMT scenario, the pre-/post-processing must run in real-time and should be reversible as much as possible. In each human–machine interaction, the current prefix has to be pre-processed for the interactive-predictive engine and then the generated completion has to be post-processed for the user. It is crucial that the pre-processing of prefixes is fully compatible with the training corpus.

5.2 Xerox and EU Corpora

Six bilingual corpora were used for two different tasks and three different language pairs in the framework of the TT2 project (SchlumbergerSema S.A. et al. 2001). The language pairs involved were English–Spanish, English–French, and English–German (Khadivi and Goutte 2003), and the tasks were *Xerox* (Xerox printer manuals) and *EU* (Bulletin of the European Union).

The three *Xerox* corpora were obtained from different user manuals for Xerox printers (SchlumbergerSema S.A. et al. 2001). The main features of these corpora are shown in Table 1. Dividing the corpora into training and test sets was performed by randomly selecting (without replacement) a specified amount of *test* sentences and leaving the remaining ones for *training*. It is worth noting that the manuals were not the same in each pair of languages. Even though all training and test sets have similar size, this probably explains why the perplexity varies considerably over the different language pairs. The vocabulary size was computed using the tokenized and true-case corpus.

The three bilingual *EU* corpora were extracted from the *Bulletin of the European Union*, which exists in all official languages of the European Union (Khadivi and Goutte

Table 1
The Xerox corpora. For all the languages, the training/test full-sentence overlap and the rate of out-of-vocabulary test-set words were less than 10% and 1%, respectively. Trigram models were used to compute the test word perplexity. (K and M denote thousands and millions, respectively.)

		English/Spanish	English/German	English/French
Train	Sent. pairs (K)	56	49	53
	Running words (M)	0.7/0.7	0.6/0.5	0.6/0.7
	Vocabulary (K)	15/17	14/25	14/16
Test	Sentences (K)	1.1	1.0	1.0
	Running words (K)	8/10	12/12	11/12
	Running chars. (K)	46/59	63/73	56/65
	Perplexity	99/58	57/93	109/70

2003) and is publicly available on the Internet. The corpora used in the experiments which are described subsequently were again acquired and processed in the framework of the TT2 project. The main features of these corpora are shown in Table 2. The vocabulary size and the training and test set partitions were obtained in a similar way as with the *Xerox* corpora.

5.3 Assessment

In all the experiments reported in this article, system performance is assessed by comparing test sentence translations produced by the translation systems with the corresponding target language references of the test set. Some of the computed assessment figures measure the quality of the translation engines without any system–user interactivity:

- **Word error rate** (WER): The minimum number of substitution, insertion, and deletion operations needed to convert the word strings produced by the translation system into the corresponding single-reference word strings. WER is normalized by the overall number of words in the reference sentences (Och and Ney 2003).

Table 2
The EU corpora. For all the languages, the training/test full-sentence overlap and the rate of out-of-vocabulary test-set words were less than 3% and 0.2%, respectively. Trigram models were used to compute the test word perplexity. (K and M denote thousands and millions, respectively.)

		English/Spanish	English/German	English/French
Train	Sent. pairs (K)	214	223	215
	Running words (M)	5.2/5.9	5.7/5.4	5.3/6.0
	Vocabulary (K)	84/97	86/153	84/91
Test	Sentences (K)	0.8	0.8	0.8
	Running words (K)	20/23	20/19	20/23
	Running chars. (K)	119/135	120/134	119/134
	Perplexity	58/46	57/87	58/45

- **Bilingual evaluation understudy** (BLEU): This is based on the coverage of *n*-grams of the hypothesized translation which occur in the reference translations (Papineni et al. 2001).

Other assessment figures are aimed at estimating the effort needed by a human translator to produce correct translations using the interactive system. To this end, the target translations which a real user would have in mind are simulated by the given references. The first translation hypothesis for each given source sentence is compared with a single reference translation and the longest common character prefix (LCP) is obtained. The first non-matching character is replaced by the corresponding reference character and then a new system hypothesis is produced. This process is iterated until a full match with the reference is obtained.

Each computation of the LCP would correspond to the user looking for the next error and *moving the pointer* to the corresponding position of the translation hypothesis. Each character replacement, on the other hand, would correspond to a *keystroke* of the user. If the first non-matching character is the first character of the new system hypothesis in a given iteration, no LCP computation is needed; that is, no pointer movement would be made by the user. Bearing this in mind, we define the following interactive-predictive performance measures:

- **Keystroke ratio** (KSR): Number of *keystrokes* divided by the total number of reference characters.
- **Mouse-action ratio** (MAR): Number of *pointer movements* plus one more count per sentence (aimed at simulating the user action needed to accept the final translation), divided by the total number of reference characters.
- **Keystroke and mouse-action ratio** (KSMR): KSR plus MAR.

Note that KSR estimates only the user's actions on the keyboard whereas MAR estimates actions for which the user would typically use the mouse. From a user point of view the two types of actions are different and require different types of effort (Macklovitch, Nguyen, and Silva 2005; Macklovitch 2006). In any case, as an approximation, KSMR accounts for both KSR and MAR, assuming that both actions require a similar effort.

In the case of SMT systems, it is well known that an automatically computed quality measure like BLEU correlates quite well with human judgment (Callison-Burch, Osborne, and Koehn 2006). In the case of IPMT, we should keep in mind that the main goal of (automatic) assessment is to estimate the effort of the human translator. Moreover, translation quality is not an issue here, because the (simulated) human intervention ensures "perfect" translation results. The important question is whether the (estimated) productivity of the human translator can really be increased or not by the IPMT approach. In order to answer this question, the KSR and KSMR measures will be used in the IPMT experiments to be reported in the next section.

In order to show the statistical significance of the results, all the assessment figures reported in the next section are accompanied by the corresponding *95% confidence intervals*. These intervals have been computed using bootstrap sampling techniques, as proposed by Bisani and Ney (2004), Koehn (2004), and Zhang and Vogel (2004).

6. Results

Two types of results are reported for each corpus and for each translation approach. The first are conventional MT results, obtained as a reference to give an idea of the "classical" MT difficulty of the selected tasks. The second aim is to assess the interactive MT (IPMT) approach proposed in this article.

The results are presented in different subsections. The first two subsections present the MT and IPMT results for the 1-best translation obtained by the different techniques in the *Xerox* and *EU* tasks, respectively. The third subsection presents further IPMT results for the 5-best translations on a single pair of languages.

Some of these results may differ from results presented in previous works (Cubel et al. 2003; Och, Zens, and Ney 2003; Civera et al. 2004a; Cubel et al. 2004; Bender et al. 2005). The differences are due to variations in the pre-/post-processing procedures and/or recent improvements of the search techniques used by the different systems.

6.1 Experiments with the Xerox Corpora

In this section, the translation results obtained using ATs, PBMs, and SFSTs for all six language pairs of the Xerox corpus are reported. Word-based trigram and class-based five-gram target-language models were used for the AT models (the parameters of the log-linear model are tuned so as to minimize WER on a development corpus); word-based trigram target-language models were used for PBMs and trigrams were used to infer GIATI SFSTs.

Off-line MT Results. MT results with ATs, PBMs, and SFSTs are presented in Figure 3. Results obtained using the PBMs are slightly but consistently better that those achieved using the other models. In general, the different techniques perform similarly for the various translation directions. However, the English–Spanish language pair is the one for which the best translations can be produced.

IPMT Results. Performance has been measured in terms of KSRs and MARs (KSR and MAR are represented as the lower and upper portions of each bar, respectively, and KSMR is the whole bar length). The results are shown in Figure 4.

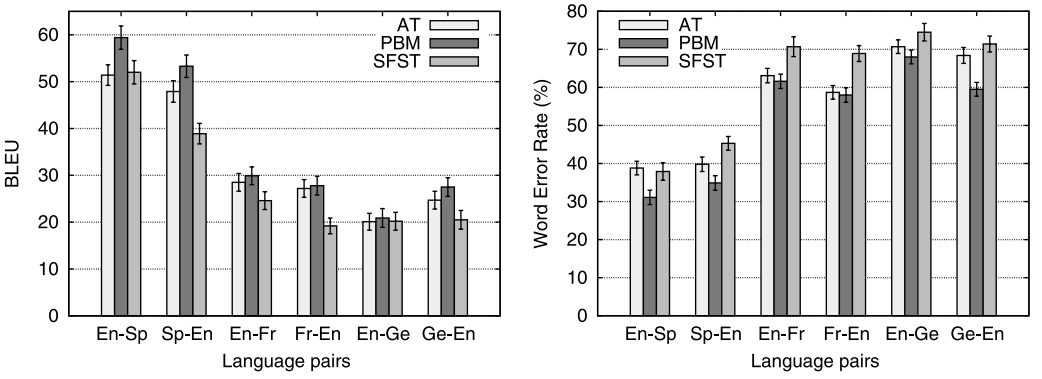

Figure 3
Off-line MT results (BLEU and WER) for the Xerox corpus. Segments above the bars show the 95% confidence intervals. En = English; Sp = Spanish; Fr = French; Ge = German.

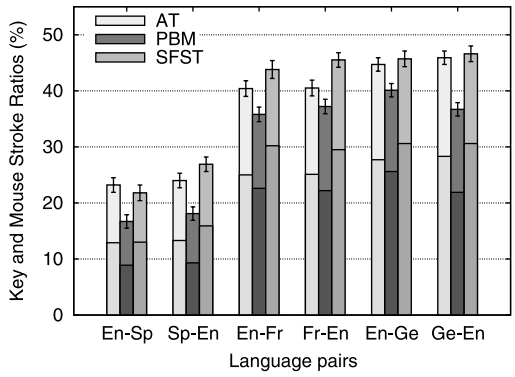

Figure 4
IPMT results for the Xerox corpus. In each bar, KSR is represented by the lower portion, MAR by the upper portion, and KSMR is the whole bar. Segments above the bars show the 95% confidence intervals. En = English; Sp = Spanish; Fr = French; Ge = German.

According to these results, a human translator assisted by an AT-based or a SFST-based interactive system would only need an effort equivalent to typing about 20% of the characters in order to produce the correct translations for the Spanish to English task; or even less than 20% if a PBM-based system is used.

For the Xerox task, off-line MT performance and IPMT results show similar tendencies. The PBMs show better performance for both the off-line MT and for the IPMT assessment figures. The AT and SFST models perform more or less equivalently. In both scenarios, the best results were achieved for the Spanish–English language pair followed by French–English and German–English.

The computing times needed by all the systems involved in these experiments were well within the range of the on-line operational requirements. The average initial time for each source test sentence was very low (less than 50 msec) for PBMs and SFSTs and adequate for ATs (772 msec). In the case of ATs and SFSTs, this included the time required for the generation of the initial word-graph of each sentence. Moreover, the most critical times incurred in the successive IPMT iterations were very low in all the cases: 18 msec for ATs, 99 msec for PBMs, and 9 msec for SFSTs. Note, however, that these average times are not exactly comparable because of the differences in the computer hardware used by each system (2 Ghz AMD, 1.5 Ghz Pentium, and 2.4 Ghz Pentium for ATs, PBMs, and SFSTs, respectively).

6.2 Experiments with the EU Corpora

The translation results using the AT, PBM, and SFST approaches for all six language pairs of the EU corpus are reported in this section. As for the Xerox corpora, in the AT experiments, word-based trigram and class-based five-gram target-language models were used; in the PBM experiments, word-based trigram and class-based five-gram target-language models were also used and five-grams were used to infer GIATI SFSTs.

Off-line MT Results. Figure 5 presents the results obtained using ATs, PBMs, and SFSTs. Generally speaking, the results are comparable to those obtained on the Xerox corpus with the exception of the English–Spanish language pair, which were better. With these corpora, the best results were obtained with the ATs and PBMs for all the pairs and the best translation direction was French-to-English with all the models used.

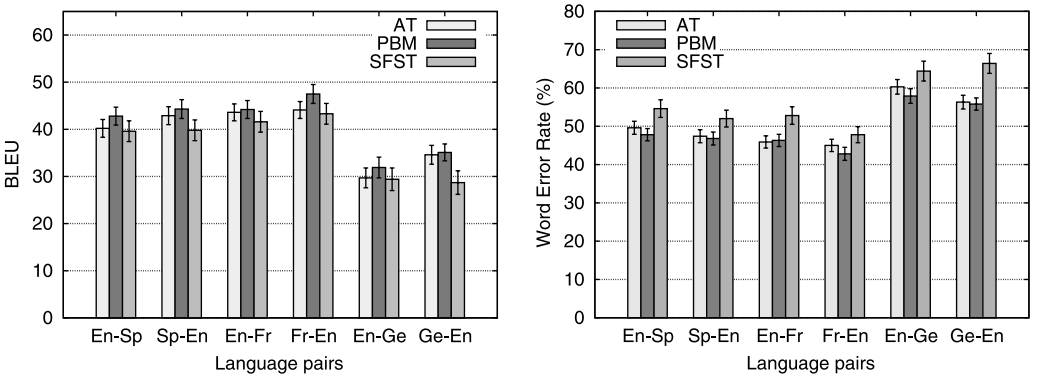

Figure 5
Off-line MT results (BLEU and WER) for the EU corpus. Segments above the bars show the 95% confidence intervals. En = English; Sp = Spanish; Fr = French; Ge = German.

IPMT Results. Figure 6 shows the performance of the AT, PBM, and SFST systems in terms of KSRs and MARs in a similar way as for the Xerox corpora.

As in the MT experiments, the results are comparable to those obtained on the Xerox corpus, with the exception of the English–Spanish pair. Similarly, as in MT, the best results were obtained for the French-to-English translation direction.

Although EU is a more open-domain task, the results demonstrate again the potential benefit of computer-assisted translation systems. Using PBMs, a human translator would only need an effort equivalent to typing about 20% of the characters in order to produce the correct translations for French-to-English translation direction, whereas for ATs and SFSTs the effort would be about 30%. For the other language pairs, the efforts would be about 20–30% and 35% of the characters for PBMs and ATs/SFSTs, respectively.

The systemwise correlation between MT and IPMT results on this corpus is not as clear as in the Xerox case. One possible cause is the much larger size of the EU corpus compared to the Xerox corpus. In order to run the EU experiments within reasonable time limits, all the systems have required the use of beam search and/or other

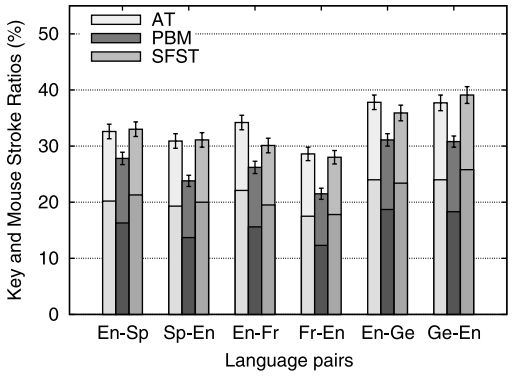

Figure 6
IPMT results for the EU corpus. In each bar, KSR is represented by the lower portion, MAR by the upper portion and KSMR is the whole bar. Segments above the bars show the 95% confidence intervals. En = English; Sp = Spanish; Fr = French; Ge = German.

Table 3
IPMT results (%) for the Xerox corpus (English–Spanish) using ATs, PBMs, and SFSTs for the 1-best hypothesis and 5-best hypotheses. 95% confidence intervals are shown.

	1-best		5-best	
Technique	KSR	KSMR	KSR	KSMR
AT	12.9±0.9	23.2±1.3	11.1±0.8	20.3±1.2
PBM	8.9±0.8	16.7±1.2	7.3±0.6	15.4±1.1
SFST	13.0±1.0	21.8±1.4	11.2±1.0	19.2±1.3

suboptimal pruning techniques, although this was largely unnecessary for the Xerox corpus. Clearly, the pruning effects are different in the off-line (MT) and the on-line (IPMT) search processes and the differences may lead to wide performance variations for the AT, PBM, and SFST approaches.

Nevertheless, as can be seen in Bender et al. (2005), the degradation in system performance due to pruning is generally not too substantial and sufficiently accurate real-time interactive operation could also be achieved in the EU task with the three systems tested.

6.3 Results with *n*-Best Hypotheses

Further experiments were carried out to study the usefulness of *n*-best hypotheses in the interactive framework. In this scenario, the user can choose one out of *n* proposed translation suffixes and then proceed as in the usual IPMT paradigm. As with the previous experiments, the automated evaluation is based on a selected target sentence that best matches a prefix of the reference translation in each IPMT iteration (therefore KSR is minimized).

Here, only IPMT results for the English-to-Spanish translation direction are reported for both Xerox and EU tasks, using a list of the five best translations. These results are shown in Tables 3 and 4.

In all the cases there is a clear and significant accuracy improvement when moving from single-best to 5-best translations. This gain in translation quality diminishes in a log-wise fashion as we increase the number of best translations. From a practical point of view, the improvements provided by using *n*-best completions would come at the cost of the user having to ponder which of these completions is more suitable. In a real operational environment, this additional user effort may or may not outweigh the

Table 4
IPMT results (%) for the EU corpus (English–Spanish) using ATs, PBMs, and SFSTs for the 1-best hypothesis and 5-best hypotheses. 95% confidence intervals are shown.

	1-best		5-best	
Technique	KSR	KSMR	KSR	KSMR
AT	20.2±0.9	32.6±1.3	18.5±0.8	29.9±1.2
PBM	16.3±0.7	27.8±1.1	13.2±0.6	25.0±1.1
SFST	21.3±0.9	33.0±1.3	19.3±0.9	29.9±1.3

benefits of the *n*-best increased accuracy. Consequently, this feature should be offered to the users as an option.

7. Practical Issues

IPMT results reported in the previous section provide reasonable *estimations* of potential savings of human translator effort, assuming that the goal is to obtain high quality translations. In real work, however, several practical issues not discussed in this article may significantly affect the actual system usability and overall user productivity.

One of the most obvious issues is that a carefully designed graphical user interface (GUI) is needed to let the users actually be in command of the translation process, so that they really feel the system is assisting them rather than the other way around. In addition, an adequate GUI has to provide adequate means for the users to easily and intuitively change at will IPMT engine parameters that may have an impact on their way of working with the system. To name just a few: The maximum length of system hypotheses, the value of n for n-best suggestions, or the "interaction step granularity"; that is, whether the system should react at each user keystroke, or at the end of each complete typed word, or after a sufficiently long typing pause, and so on.

Clearly, all these important issues are beyond the scope of the present article. But we can comment that, in the TT2 project, complete prototypes of some of the systems presented in this article, including the necessary GUI, were actually implemented and thoroughly evaluated by professional human translators in their working environment (Macklovitch, Nguyen, and Silva 2005; Macklovitch 2006).

The results of these field tests showed that the actual productivity depended not only on the individual translators, but also on the given test texts. In cases where these texts were quite unrelated to the training data, the system did not significantly help the human translators to increase their productivity. However, when the test texts were reasonably well related to the training data, high productivity gains were registered—close to what could be expected according to the KSR/MAR empirical results.

8. Concluding Remarks

The IPMT paradigm proposed in this article allows for a close collaboration between a human translator and a machine translation system. This paradigm entails an iterative process where, in each iteration, a data-driven machine translation engine suggests a completion for the current prefix of a target sentence which a human translator can accept, modify, or ignore.

This idea was originally proposed in the TransType project (Langlais, Foster, and Lapalme 2000), where a simple engine was used which only supported single-token suggestions. Furthering these ideas, in the TransType2 project (SchlumbergerSema S.A. et al. 2001), state-of-the-art statistical machine translation systems have been developed and integrated in the IPMT framework.

In a laboratory environment, results on two different tasks suggest that the proposed techniques can reduce the typing effort needed to produce a high-quality translation of a given source text by as much as 80% with respect to the effort needed to simply type the whole translation. In real conditions, a high productivity gain was achieved in many cases.

We have studied here IPMT from the point of view of a standalone CAT tool. Nevertheless, IPMT can of course be easily and conveniently combined with other popular translator workbench tools. More specifically, IPMT lends itself particularly

well to addressing the typical lack of generalization capabilities of translation memories. When used as a CAT tool, translation memories allow the human translator to keep producing increasingly long segments of correct target text. Clearly, these segments can be used by an IPMT engine to suggest to the translator possible translations for source text segments that are not found in the translation memories as exact matches.

Acknowledgments

This work has been partially supported by the ST Programme of European Union under grant IST-2001-32091, by the Spanish project TIC–2003-08681-C02-02, and the Spanish research programme Consolider Ingenio-2010 CSD2007-00018. The authors wish to thank the anonymous reviewers for their criticisms and suggestions.

References

Amengual, J. C., J. M. Benedí, A. Castaño, A. Castellanos, V. M. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J. M. Vilar. 2000. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103.

Amengual, J. C. and E. Vidal. 1998. Efficient error-correcting Viterbi parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1109–1116.

Bender, O., S. Hasan, D. Vilar, R. Zens, and H. Ney. 2005. Comparison of generation strategies for interactive machine translation. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation (EAMT 05)*, pages 33–40, Budapest.

Berger, A. L., P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer. 1996. Language translation apparatus and method of using context-based translation models. United States Patent No. 5510981, April.

Berstel, J. 1979. *Transductions and Context-Free Languages*. B. G. Teubner, Stuttgart.

Bisani, M. and H. Ney. 2004. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP 04)*, volume 1, pages 409–412, Montreal.

Bowker, L. 2002. *Computer-Aided Translation Technology: A Practical Introduction*, chapter 5: Translation-memory systems. Didactics of Translation. University of Ottawa Press, pages 92–127.

Brown, P. F., J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roosin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–310.

Callison-Burch, C., M. Osborne, and P. Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 06)*, pages 249–256, Trento.

Casacuberta, F., H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. García-Varea, D. Llorens, C. Martínez, S. Molau, F. Nevado, M. Pastor, D. Picó, A. Sanchis, and C. Tillmann. 2004a. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47.

Casacuberta, F. and E. Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225.

Casacuberta, F. and E. Vidal. 2007. Learning finite-state models for machine translation. *Machine Learning*, 66(1):69–91.

Casacuberta, F., E. Vidal, and D. Picó. 2005. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38:1431–1443.

Casacuberta, F., E. Vidal, A. Sanchis, and J. M. Vilar. 2004b. Pattern recognition approaches for speech-to-speech translation. *Cybernetic and Systems: an International Journal*, 35(1):3–17.

Civera, J., J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, and J. González. 2004a. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 04)*, pages 349–356, Barcelona.

Civera, J., J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, F. Casacuberta, E. Vidal, D. Picó, and J. González. 2004b. A syntactic pattern recognition approach to computer assisted translation. In *Advances in*

Statistical, Structural and Syntactical Pattern Recognition, Proceedings of the Joint IAPR International Workshops on Syntactical and Structural Pattern Recognition (SSPR 04) and Statistical Pattern Recognition (SPR 04)), Lisbon, Portugal, August 18–20, volume 3138 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, pages 207–215.

Cubel, E., J. Civera, J. M. Vilar, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, J. González, and L. Rodríguez. 2004. Finite-state models for computer assisted translation. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 04)*, pages 586–590, Valencia.

Cubel, E., J. González, A. Lagarda, F. Casacuberta, A. Juan, and E. Vidal. 2003. Adapting finite-state translation to the TransType2 project. In *Proceedings of the Joint Conference Combining the 8th International Workshop of the European Association for Machine Translation and the 4th Controlled Language Applications Workshop (EAMT-CLAW 03)*, pages 54–60, Dublin.

Foster, G. 2002. *Text Prediction for Translators*. Ph.D. thesis, Université de Montréal, Canada.

Foster, G., P. Isabelle, and P. Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12(1–2):175–194.

Isabelle, P. and K. Church. 1997. Special issue on new tools for human translators. *Machine Translation*, 12(1–2).

Jelinek, F. 1998. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, MA.

Jiménez, V. M. and A. Marzal. 1999. Computing the k shortest paths: a new algorithm and an experimental comparison. In *Algorithm Engineering: Proceedings of the 3rd International Workshop (WAE 99), London, UK, July 19–21*, volume 1668 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, pages 15–29.

Kay, M. 1997. The proper place of men and machines in language translation. *Machine Translation*, 12:3–23. [This article first appeared as a Xerox PARC Working Paper in 1980].

Khadivi, S. and C. Goutte. 2003. Tools for corpus alignment and evaluation of the alignments (deliverable d4.9). Technical report, TransType2 (IST-2001-32091).

Khadivi, S., R. Zens, and H. Ney. 2006. Integration of speech to computer-assisted translation using finite-state automata. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21th International Conference on Computational Linguistics (COLING/ACL 06)*, pages 467–474, Sydney.

Khadivi, S., A. Zolnay, and H. Ney. 2005. Automatic text dictation in computer-assisted translation. In *Proceedings of the European Conference on Speech Communication and Technology, (INTERSPEECH 05-EUROSPEECH)*, pages 2265–2268, Lisbon.

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 04)*, pages 388–395, Barcelona.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 03)*, pages 127–133, Edmonton.

Langlais, P., G. Foster, and G. Lapalme. 2000. TransType: a computer-aided translation typing system. In *Proceedings of the NAACL/ANLP Workshop on Embedded Machine Translation Systems*, pages 46–52, Seattle, WA.

Langlais, P., G. Lapalme, and M. Loranger. 2002. Transtype: Development-evaluation cycles to boost translator's productivity. *Machine Translation*, 15(4):77–98.

Macklovitch, E. 2006. TransType2: The last word. In *Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC 06)*, pages 167–172, Genoa.

Macklovitch, E., N. T. Nguyen, and R. Silva. 2005. User evaluation report. Technical report, TransType2 (IST-2001-32091).

Marcu, D. and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 02)*, pages 133–139, Philadelphia, PA.

Ney, H., S. Nießen, F. Och, H. Sawaf, C. Tillmann, and S. Vogel. 2000. Algorithms for statistical translation of spoken language. *IEEE Transactions on Speech and Audio Processing*, 8(1):24–36.

Och, F. J. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the 9th Conference of the European Chapter of the Association for*

Computational Linguistics (EACL 99), pages 71–76, Bergen.

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

Och, F. J. and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–450.

Och, F. J., R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 03)*, pages 387–393, Budapest.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176, Thomas J. Watson Research Center.

Picó, D. and F. Casacuberta. 2001. Some statistical-estimation methods for stochastic finite-state transducers. *Machine Learning*, 44:121–142.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 2002. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK.

SchlumbergerSema S.A., Intituto Tecnológico de Informática, Rheinisch Westfälische Technische Hochschule Aachen Lehrstul für Informatik VI, Recherche Appliquée en Linguistique Informatique Laboratory University of Montreal, Celer Soluciones, Société Gamma, and Xerox Research Centre Europe. 2001. TT2. TransType2—computer-assisted translation. Project technical annex. Information Society Technologies (IST) Programme, IST-2001-32091.

Sen, Z., Ch. Zhaoxiong, and H. Heyan. 1997. Interactive approach in machine translation systems. In *Proceedings of IEEE International Conference on Intelligent Processing Systems (ICIPS 97)*, pages 1814–1819, Beijing.

Slocum, J. 1985. A survey of machine translation: Its history, current status and future prospects. *Computational Linguistics*, 11(1):1–17.

Somers, H., 2003. *Computers and Translation: a Translator's Guide*, chapter 3: Translation memory systems. John Benjamins, Amsterdam, pages 31–48.

Tomás, J. and F. Casacuberta. 2001. Monotone statistical translation using word groups. In *Proceedings of the Machine Translation Summit VIII (MT SUMMIT VIII)*, pages 357–361, Santiago de Compostela.

Tomás, J. and F. Casacuberta. 2003. Combining phrase-based and template-based alignment models in statistical translation. In *Pattern Recognition and Image Analysis, Proceedings of the First Iberian Conference (IbPRIA 03), Puerto de Andratx, Mallorca, Spain, June 4-6*, volume 2652 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, pages 1020–1031.

Tomás, J. and F. Casacuberta. 2004. Statistical machine translation decoding using target word reordering. In *Advances in Statistical, Structural and Syntactical Pattern Recognition, Proceedings of the Joint IAPR International Workshops on Syntactical and Structural Pattern Recognition (SSPR 04) and Statistical Pattern Recognition (SPR 04), Lisbon, Portugal, August 18–20*, volume 3138 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, pages 734–743.

Tomás, J. and F. Casacuberta. 2006. Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21th International Conference on Computational Linguistics (COLING/ACL 06)*, pages 835–841, Sydney.

Tomás, J. and F. Casacuberta. 2007. A pattern recognition approach to machine translation: Monotone and non-monotone phrase-based statistical models. Technical Report DSIC-II/18/07, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Tomita, M. 1985. Feasibility study of personal/interactive machine translation systems. In *Proceedings of the First International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 85)*, pages 289–297, New York, NY.

Ueffing, N., F. J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 02)*, pages 156–163, Philadelphia, PA.

Vidal, E. 1997. Finite-state speech-to-speech translation. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP 97)*, volume 1, pages 111–114, Munich.

Vidal, E. and F. Casacuberta. 2004. Learning finite-state models for machine translation. In *Grammatical Inference: Algorithms and Applications, Proceedings of the 7th International Coloquium on Grammatical Inference (ICGI 04), Athens, Greece, October 11–13*, volume 3264 of *Lecture Notes in Artificial Intelligence*. Springer, Heidelberg, pages 16–27.

Vidal, E., F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. 2006. Computer-assisted translation using speech recognition. *IEEE Transactions on Speech and Audio Processing*, 14(3):941–951.

Vidal, E., F. Thollard, F. Casacuberta C. de la Higuera, and R. Carrasco. 2005. Probabilistic finite-state machines—part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1025–1039.

Whitelock, P. J., M. McGee Wood, B. J. Chandler, N. Holden, and H. J. Horsfall. 1986. Strategies for interactive machine translation: The experience and implications of the UMIST Japanese project. In *Proceedings of the 11th International Conference on Computational Linguistics (COLING 86)*, pages 329–334, Bonn.

Yamron, J., J. Baker, P. Bamberg, H. Chevalier, T. Dietzel, J. Elder, F. Kampmann, M. Mandel, L. Manganaro, T. Margolis, and E. Steele. 1993. LINGSTAT: an interactive, machine-aided translation system. In *Proceedings of the Workshop on Human Language Technology*, pages 191–195, Princeton, NJ.

Zajac, R. 1988. Interactive translation: A new approach. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING 88)*, pages 785–790, Budapest.

Zens, R. and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL 04)*, pages 257–264, Boston, MA.

Zens, R., F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Advances in Artificial Intelligence. 25th Annual German Conference on Artificial Intelligence (KI 02), Aachen, Germany, September 16–22, Proceedings*, volume 2479 of *Lecture Notes on Artificial Intelligence*. Springer Verlag, Heidelberg, pages 18–32.

Zhang, Y. and S. Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the Tenth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 04)*, pages 294–301, Baltimore, MD.