Interactive Machine Translation using Hierarchical Translation Models

Jesús González-Rubio, Daniel Ortiz-Martínez, José-Miguel Benedí, Francisco Casacuberta
D. de Sistemas Informáticos y Computación
Universitat Politècnica de València
Camino de Vera s/n, 46021 Valencia (Spain)
{jegonzalez,dortiz,jbenedi,fcn}@dsic.upv.es

Abstract

Current automatic machine translation systems are not able to generate error-free translations and human intervention is often required to correct their output. Alternatively, an interactive framework that integrates the human knowledge into the translation process has been presented in previous works. Here, we describe a new interactive machine translation approach that is able to work with phrase-based and hierarchical translation models, and integrates error-correction all in a unified statistical framework. In our experiments, our approach outperforms previous interactive translation systems, and achieves estimated effort reductions of as much as 48% relative over a traditional post-edition system.

1 Introduction

Research in the field of *machine translation* (MT) aims to develop computer systems which are able to translate between languages automatically, without human intervention. However, the quality of the translations produced by any automatic MT system still remain below than that of human translation. Typical solutions to reach human-level quality require a subsequent manual *post-editing* process. Such decoupled post-edition solution is rather inefficient and tedious for the human translator. Moreover, it prevents the MT system from taking advantage of the knowledge of the human translator and, reciprocal, the human translator cannot take advantage of the adapting ability of MT technology.

An alternative way to take advantage of the existing MT technology is to use them in *collaboration* with human translators within a *computer-assisted translation* (CAT) or *interactive* framework (Isabelle and Church, 1998). The *TransType* and *TransType2* projects (Foster et al., 1998; Langlais and Lapalme, 2002; Barrachina et al., 2009) entailed an interesting focus shift in CAT technology by aiming interaction directly at the production of the target text. These research projects proposed to embed an MT system within an interactive translation environment. This way, the human translator can ensure a high-quality output while the MT system ensures a significant gain of productivity. Particularly interesting is the *interactive machine translation* (IMT) approach proposed in (Barrachina et al., 2009). In this scenario, a statistical MT (SMT) system uses the source sentence and a previously validated part (prefix[1]) of its translation to propose a suitable continuation. Then the user finds and corrects the next system error, thereby providing a longer prefix which the system uses to suggests a new, hopefully better continuation. The reported results showed that IMT can save a significant amount of human effort.

Barrachina et al,. (2009) provide a thorough description of the IMT approach and describe algorithms for its practical implementation. Nevertheless, we identify two basic problems for which we think there is room for improvement. The first problem arises when the system cannot generate the prefix validated by the user. To solve this problem, the authors simply provide an ad-hoc heuristic error-correction technique. The second problem is how the system deals with word reordering. Particularly, the models used by the system were either mono-

[1] We use the terms prefix and suffix to denote any sub-string at the beginning and end respectively of a string of characters (including spaces and punctuation). These terms do not imply any morphological significance as they usually do in linguistics.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 244–254,
Seattle, Washington, USA, 18-21 October 2013.

tonic by nature or non-monotonic but heuristically defined (not estimated from training data).

We work on the foundations of Barrachina et al., (2009) and provide formal solutions to these two challenges. On the one hand, we adopt the statistical formalization of the IMT framework described in (Ortiz-Martínez, 2011), which includes a stochastic error-correction model in its formalization to address prefix coverage problems. Moreover, we refine this formalization proposing an alternative error-correction formalization for the IMT framework (Section 2). Additionally, we also propose a specific error-correction model based on a statistical interpretation of the Levenshtein distance (Levenshtein, 1966). These formalizations provide a unified statistical framework for the IMT model in comparison to the ad-hoc heuristic error-correction methods previously used.

In order to address the problem of properly deal with reordering in IMT, we introduce the use of hierarchical MT models (Chiang, 2005; Zollmann and Venugopal, 2006). These methods provide a natural approach to handle long range dependencies and allow the incorporation of reordering information into a consistent statistical framework. Here, we also describe how state-of-the-art hierarchical MT models can be extended to handle IMT (Sections 3 and 4).

We evaluate the proposed IMT approach on two different translation task. The comparative results against the IMT approach described by Barrachina et al., (2009) and a conventional post-edition approach show that our IMT formalization for hierarchical SMT models indeed outperform other approaches (Sections 5 and 6). Moreover, it leads to large reductions in the human effort required to generate error-free translations.

2 Statistical Framework

2.1 Statistical Machine Translation

Assuming that we are given a sentence $\mathbf{s}$ in a source language, the *translation* problem can be stated as finding its translation $\mathbf{t}$ in a target language of maximum probability (Brown et al., 1993):

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t} \mid \mathbf{s}) \tag{1}$$

$$= \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \tag{2}$$

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

IT-0	$\mathbf{p}$						
	$\mathbf{t}_s$	*To*	*view*	*the*	*resources*	*list*	
IT-1	$\mathbf{p}$	To	view				
	k			$\boxed{\mathbf{a}}$			
	$\mathbf{t}_s$				*list*	*of*	*resources*
IT-2	$\mathbf{p}$	To	view	a	list		
	k				$\boxed{\mathbf{i}}$		
	$\mathbf{t}_s$				*ng*	*resources*	
IT-3	$\mathbf{p}$	To	view	a	listing		
	k					$\boxed{\mathbf{o}}$	
	$\mathbf{t}_s$					*f*	*resources*
END	$\mathbf{p}$	To	view	a	listing	of	resources

Figure 1: IMT session to translate a Spanish sentence into English. The desired translation is the translation the human user wants to obtain. At IT-0, the system suggests a translation ($\mathbf{t}_s$). At IT-1, the user moves the mouse to accept the first eight characters "To view " and presses the $\boxed{\text{a}}$ key (k), then the system suggests completing the sentence with "*list of resources*" (a new $\mathbf{t}_s$). Iterations 2 and 3 are similar. In the final iteration, the user accepts the current translation.

The terms in the latter equation are the *language model* probability $\Pr(\mathbf{t})$ that represents the well-formedness of $\mathbf{t}$ (*n-gram* models are usually adopted), and the *(inverted) translation model* $\Pr(\mathbf{s} \mid \mathbf{t})$ that represents the relationship between the source sentence and its translation.

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, 2002):

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \left\{ \sum_{n=1}^{N} \lambda_n \cdot \log(f_n(\mathbf{t}, \mathbf{s})) \right\} \tag{3}$$

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

2.2 Statistical Interactive Machine Translation

Unfortunately, current MT technology is still far from perfect. This implies that, in order to achieve good translations, manual post-editing is needed. An alternative to this decoupled approach (first MT, then manual correction) is given by the IMT paradigm (Barrachina et al., 2009). Under this paradigm, translation is considered as an iterative left-to-right process where the human and the computer collaborate to generate the final translation.

Figure 1 shows an example of the IMT approach. There, a source Spanish sentence $\mathbf{s}$ =”Para ver la lista de recursos” is to be translated into a target English sentence $\hat{\mathbf{t}}$. Initially, with no user feedback, the system suggests a complete translation $\mathbf{t}_s$ =”To view the resources list”. From this translation, the user marks a prefix $\mathbf{p}$ =”To view” as correct and begins to type the rest of the target sentence. Depending on the system or the user’s preferences, the user might type the full next word, or only some letters of it (in our example, the user types the single next character “a”). Then, the MT system suggests a new suffix $\mathbf{t}_s$ =“list of resources” that completes the validated prefix and the input the user has just typed ($\mathbf{p}$ =”To view a”). The interaction continues with a new prefix validation followed, if necessary, by new input from the user, and so on, until the user considers the translation to be complete and satisfactory.

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. Formally, the best suffix of a given length will be:

$$\hat{\mathbf{t}}_s = \arg\max_{\mathbf{t}_s} \Pr(\mathbf{t}_s \mid \mathbf{s}, \mathbf{p}) \tag{4}$$

which can be straightforwardly rewritten as:

$$\hat{\mathbf{t}}_s = \arg\max_{\mathbf{t}_s} \Pr(\mathbf{p}, \mathbf{t}_s \mid \mathbf{s}) \tag{5}$$

$$= \arg\max_{\mathbf{t}_s} \Pr(\mathbf{p}, \mathbf{t}_s) \cdot \Pr(\mathbf{s} \mid \mathbf{p}, \mathbf{t}_s) \tag{6}$$

Note that, since $\mathbf{p}\,\mathbf{t}_s = \mathbf{t}$, this equation is very similar to Equation (2). The main difference is that now the search process is restricted to those target sentences $\mathbf{t}$ that contains $\mathbf{p}$ as prefix. This implies that we can use the same MT models (including the log-linear approach) if the search procedures are adequately modified (Och et al., 2003). Finally, it should be noted that the statistical models are usually defined at word level, while the IMT process described in this section works at character level. To deal with this problem, during the search process it is necessary to verify the compatibility between $\mathbf{t}$ and $\mathbf{p}$ at character level.

2.3 IMT with Stochastic Error-Correction

A common problem in IMT arises when the user sets a prefix which cannot be explained by the statistical models. To solve this problem, IMT systems typically include ad-hoc error-correction techniques to guarantee that the suffixes can be generated (Barrachina et al., 2009). As an alternative to this heuristic approach, Ortiz-Martínez (2011) proposed a formalization of the IMT framework that does include stochastic error-correction models in its statistical formalization. The starting point of this alternative IMT formalization accounts for the problem of finding the translation $\mathbf{t}$ that, at the same time, better explains the source sentence $\mathbf{s}$ and the prefix given by the user $\mathbf{p}$:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t} \mid \mathbf{s}, \mathbf{p}) \tag{7}$$

$$= \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s}, \mathbf{p} \mid \mathbf{t}) \tag{8}$$

The following naïve Bayes’ assumption is now made: the source sentence $\mathbf{s}$ and the user prefix $\mathbf{p}$ are statistically independent variables given the translation $\mathbf{t}$, obtaining:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{p} \mid \mathbf{t}) \tag{9}$$

where $\Pr(\mathbf{t})$ can be approximated with a language model, $\Pr(\mathbf{s} \mid \mathbf{t})$ can be approximated with a translation model, and $\Pr(\mathbf{p} \mid \mathbf{t})$ can be approximated by an error correction model that measures the compatibility between the user-defined prefix $\mathbf{p}$ and the hypothesized translation $\mathbf{t}$.

Note that the translation result, $\hat{\mathbf{t}}$, given by Equation (9) may not contain $\mathbf{p}$ as prefix because every translation is compatible with $\mathbf{p}$ with a certain probability. Thus, despite being close, Equation (9) is not equivalent to the IMT formalization in Equation (6).

To solve this problem, we define an alignment, $\mathbf{a}$, between the user-defined prefix $\mathbf{p}$ and the hypothesized translation $\mathbf{t}$, so that the unaligned words of $\mathbf{t}$, in an appropriate order, constitute the suffix searched in IMT. This allows us to rewrite the error correction probability as follows:

$$\Pr(\mathbf{p} \mid \mathbf{t}) = \sum_{\mathbf{a}} \Pr(\mathbf{p}, \mathbf{a} \mid \mathbf{t}) \tag{10}$$

To simplify things, we assume that $\mathbf{p}$ is monotonically aligned to $\mathbf{t}$, leaving the potential word-reordering to the language and translation models.

Under this assumption, $\mathbf{a}$ determines an alignment for $\mathbf{t}$, such that $\mathbf{t} = \mathbf{t}_p\mathbf{t}_s$, where $\mathbf{t}_p$ is fully-aligned to $\mathbf{p}$ and $\mathbf{t}_s$ remains unaligned. Taking all these things into consideration, and following a maximum approximation, we finally arrive to the expression:

$$(\hat{\mathbf{t}}, \hat{\mathbf{a}}) = \arg\max_{\mathbf{t},\mathbf{a}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{p}, \mathbf{a} \mid \mathbf{t}) \quad (11)$$

where the suffix required in IMT is obtained as the portion of $\hat{\mathbf{t}}$ that is not aligned with the user prefix.

In practice, we combine the models in Equation (11) in a log-linear fashion as it is typically done in SMT (see Equation (3)).

2.4 Alternative Formalization for IMT with Stochastic Error-Correction

Alternatively to Equation (11), we can operate from Equation (9) and reach a different formalization for IMT with error-correction. We can re-write the first and last terms of Equation (9) as:

$$\Pr(\mathbf{t}) \cdot \Pr(\mathbf{p} \mid \mathbf{t}) = \Pr(\mathbf{p}) \cdot \Pr(\mathbf{t} \mid \mathbf{p}) \quad (12)$$

As in the previous section, we introduce an alignment variable, $\mathbf{a}$, between $\mathbf{t}$ and $\mathbf{p}$, giving:

$$\begin{aligned} \Pr(\mathbf{t} \mid \mathbf{p}) &= \sum_{\mathbf{a}} \Pr(\mathbf{t}, \mathbf{a} \mid \mathbf{p}) && (13) \\ &= \sum_{\mathbf{a}} \Pr(\mathbf{a} \mid \mathbf{p}) \cdot \Pr(\mathbf{t} \mid \mathbf{p}, \mathbf{a}) && (14) \end{aligned}$$

If we consider monotonic alignments, $\mathbf{a}$ defines again an alignment between a prefix of the system translation ($\mathbf{t}_p$) and the user prefix, producing the suffix required in IMT ($\mathbf{t}_s$) as the unaligned part. Thus, we can re-write $\Pr(\mathbf{t} \mid \mathbf{p}, \mathbf{a})$ as:

$$\begin{aligned} \Pr(\mathbf{t} \mid \mathbf{p}, \mathbf{a}) &= \Pr(\mathbf{t}_p, \mathbf{t}_s \mid \mathbf{p}, \mathbf{a}) && (15) \\ &\approx \Pr(\mathbf{t}_p \mid \mathbf{p}, \mathbf{a}) \cdot \Pr(\mathbf{t}_s \mid \mathbf{p}, \mathbf{a}) && (16) \end{aligned}$$

where Equation (16) has been obtained following a naïve Bayes' decomposition.

Combining equations (12), (14), and (16) into Equation (9), and following a maximum approximation for the summation of the alignment variable $\mathbf{a}$, we arrive to the following expression:

$$(\hat{\mathbf{t}}, \hat{\mathbf{a}}) = \arg\max_{\mathbf{t},\mathbf{a}} \Pr(\mathbf{s} | \mathbf{t}) \cdot \Pr(\mathbf{t}_p | \mathbf{p}, \mathbf{a}) \cdot \Pr(\mathbf{t}_s | \mathbf{p}, \mathbf{a}) \quad (17)$$

where $\Pr(\mathbf{p})$ and $\Pr(\mathbf{a}|\mathbf{p})$ have been dropped down because the former does not participate in the maximization and the latter is assumed uniform.

The terms in this equation can be interpreted similarly as those in Equation (9): $\Pr(\mathbf{s} \mid \mathbf{t})$ is the translation model, $\Pr(\mathbf{t}_p \mid \mathbf{p}, \mathbf{a})$ is the error-correction probability that measures the compatibility between the prefix $\mathbf{t}_p$ of the hypothesized translation and the user-defined prefix $\mathbf{p}$, and $\Pr(\mathbf{t}_s \mid \mathbf{p}, \mathbf{a})$ is the language model for the corresponding suffix $\mathbf{t}_s$ conditioned by the user-defined prefix. Again, in the experiments we combine the different models in a log-linear fashion.

The main difference between the two alternative IMT formalization (Equations (11) and (17)) is that in the latter the suffix to be returned is conditioned by the user-validated prefix $\mathbf{p}$. Thus, in the following we will refer to Equation (11) as *independent suffix formalization* while we will denote Equation (17) by *conditioned suffix formalization*.

3 Statistical Models

We now present the statistical models used to estimate the probability distributions described in the previous section. Section 3.1 describes the error-correction model, while Section 3.2 describes the models for the conditional translation probability.

3.1 Statistical Error-Correction Model

Following the vast majority of IMT systems described in the literature, we implement an error-correction model based on the concept of edit distance (Levenshtein, 1966). Typically, IMT systems use non-probabilistic error correction models. The first stochastic error correction model for IMT was proposed in (Ortiz-Martínez, 2011) and it is based on probabilistic finite state machines. Here, we propose a simpler approach which can be seen as a particular case of the previous one. Specifically, the proposed approach models the edit distance as a Bernoulli process where each character of the candidate string has a probability p_e of being erroneous. Under this interpretation, the number of characters that need to be edited E in a sentence of length n is a random variable that follows a binomial distribution, $E \sim \mathrm{B}(n, p_e)$, with parameters n and p_e. The probability of performing exactly k edits in a sentence of n characters is given by the following probability mass function:

$$f(k; n, p_e) = \frac{n!}{k!(n-k)!}\, p_e^k (1-p_e)^{n-k} \qquad (18)$$

Note that this error-correction model penalizes equally all edit operations. Alternatively, we can model the distance with a multinomial distribution and assign different probabilities to different types of edit operations. Nevertheless, we adhere to the binomial approximation due to its simplicity.

Finally, we compute the error-correction probability between two strings from the total number of edits required to transform the candidate translation into the reference translation. Specifically, we define the error-correction distribution in Equation (11) as:

$$\Pr(\mathbf{p}, \mathbf{a} \mid \mathbf{t}) \approx \frac{|\mathbf{p}|!}{k!(|\mathbf{p}|-k)!}\, p_e^k (1-p_e)^{|\mathbf{p}|-k} \qquad (19)$$

where $k = \mathrm{Lev}(\mathbf{p}, \mathbf{t_a})$ is the character-level Levenshtein distance between the user defined prefix $\mathbf{p}$ and the prefix $\mathbf{t_a}$ of the hypothesized translation $\mathbf{t}$ defined by alignment $\mathbf{a}$. The error-correction probability $\Pr(\mathbf{t}_p \mid \mathbf{p}, \mathbf{a})$ in Equation (17) is computed analogously.

The probability of edition p_e is the single free parameter of this formulation. We will use a separate development corpus to find an adequate value for it.

3.2 Statistical Machine Translation Models

Next sections briefly describe the statistical translation models used to estimate the conditional probability distribution $\Pr(\mathbf{s} \mid \mathbf{t})$. A detailed description of each model can be found in the provided citations.

3.2.1 Phrase-Based Translation Models

Phrase-based translation models (Koehn et al., 2003) are an instance of the noisy-channel approach in Equation (2). The translation of a source sentence $\mathbf{s}$ is obtained through a generative process composed of three steps: first, the $\mathbf{s}$ is divided into K segments (phrases), next, each source phrase, $\tilde{\mathbf{s}}$, is translated into a target phrase $\tilde{\mathbf{t}}$, and finally the target phrases are reordered to compose the final translation.

The usual phrase-based implementation of the translation probability takes a log-linear form:

$$\Pr(\mathbf{s} \mid \mathbf{t}) \approx \lambda_1 \cdot |\mathbf{t}| + \lambda_2 \cdot K + \sum_{k=1}^{K} \big[\lambda_3 \cdot \log(P(\tilde{\mathbf{s}}_k \mid \tilde{\mathbf{t}}_k)) + \lambda_4 \cdot \mathrm{d}(j)\big] \qquad (20)$$

where $P(\tilde{\mathbf{s}} \mid \tilde{\mathbf{t}})$ is the translation probability between source phrase $\tilde{\mathbf{s}}$ and target phrase $\tilde{\mathbf{t}}$, and $\mathrm{d}(j)$ is a function (distortion model) that returns the score of translating the k-th source phrase given that it is separated j words from the (k–1)-th phrase. Weights λ_1 and λ_2 play a special role since they are used to control the number of words and the number of phrases of the target sentence to be generated, respectively.

3.2.2 Hierarchical Translation Models

Phrase-based models have shown a very strong performance when translating between languages that have similar word orders. However, they are not able to adequately capture the complex relationships that exist between the word orders of languages of different families such as English and Chinese. Hierarchical translation models provide a solution to this challenge by allowing gaps in the phrases (Chiang, 2005):

$$\text{yu } X_1 \text{ you } X_2 \rightarrow \text{have } X_2 \text{ with } X_1$$

where subscripts denote placeholders for subphrases. Since these rules generalize over possible phrases, they act as discontinuous phrase pairs and may also act as phrase-reordering rules. Hence, they are not only considerably more powerful than conventional phrase pairs, but they also integrate reordering information into a consistent framework.

These hierarchical phrase pairs are formalized as rewrite rules of a synchronous context-free grammar (CFG) (Aho and Ullman, 1969):

$$X \rightarrow <\boldsymbol{\gamma}, \boldsymbol{\alpha}, \sim> \qquad (21)$$

where X is a non-terminal, $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ are both strings of terminals (words) and non-terminals , and $\sim$ is a one-to-one correspondence between non-terminal occurrences in $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$. Given the example above, $\boldsymbol{\gamma} \equiv$“yu X_1 you X_2”, $\boldsymbol{\alpha} \equiv$“have X_2 with X_1”, and $\sim$ is indicated by the subscript numbers.

Additionally, two *glue rules* are also defined:

$$S \rightarrow < S_1 X_2\, ,\ S_1 X_2 > \qquad S \rightarrow < X_1\, ,\ X_1 >$$

These give the model the option to build only partial translations using hierarchical phrases, and then combine them serially as in a phrase-based model.

The typical implementation of the hierarchical translation model also takes the form of a log-linear model. Let $\mathbf{s}_{\boldsymbol{\delta}}$ and $\mathbf{t}_{\boldsymbol{\delta}}$ be the source and target strings generated by a derivation $\boldsymbol{\delta}$ of the grammar. Then, the conditional translation probability is given by:

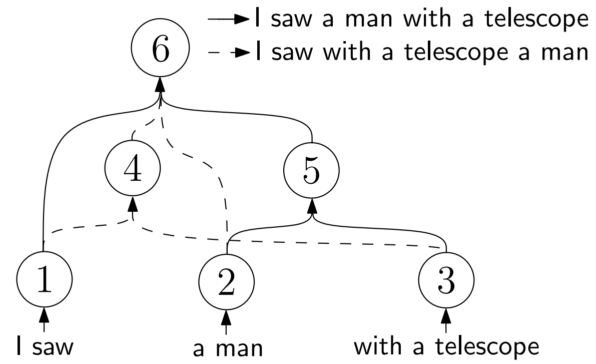
$$\Pr(\mathbf{s}_{\boldsymbol{\delta}} \mid \mathbf{t}_{\boldsymbol{\delta}}) \approx \lambda_1 \cdot |\mathbf{t}_{\boldsymbol{\delta}}| + \lambda_2 \cdot |\boldsymbol{\delta}| + \lambda_3 \cdot \#_{\mathrm{g}}(\boldsymbol{\delta}) + \sum_{r \in \boldsymbol{\delta}} [\lambda_4 \cdot w(r)] \quad (22)$$

where $|\boldsymbol{\delta}|$ denotes the total number of rules used in $\boldsymbol{\delta}$, $\#_{\mathrm{g}}(\boldsymbol{\delta})$ returns the number of applications of the glue rules, $r \in \boldsymbol{\delta}$ are the rules in $\boldsymbol{\delta}$, and $w(r)$ is the weight of rule r. Weights λ_1 and λ_2 have a similar interpretation as for phrase-based models, they respectively give some control over the total number of words and rules that conform the translation. Additionally, λ_3 controls the model's preference for hierarchical phrases over serial combination of phrases. Note that no distortion model is included in the previous equation. Here, reordering is defined at rule level by the one-to-one non-terminal correspondence. In other words, reordering is a property inherent to each rule and it is the individual score of each rule what defines, at each step of the derivation, the importance of reordering.

It should be noted that the IMT formalizations presented in Section 2 can be applied to other hierarchical or syntax-based SMT models such as those described in (Zollmann and Venugopal, 2006; Shen et al., 2010).

4 Search

In offline MT, the generation of the best translation for a given source sentence is carried out by incrementally generating the target sentence[2]. This process fits nicely into a *dynamic programming* (DP) (Bellman, 1957) framework, as hypotheses which are indistinguishable by the models can be recombined. Since the DP search space grows exponentially with the size of the input, standard DP search is prohibitive, and search algorithms usually resort to a beam-search heuristic (Jelinek, 1997).



6
→ I saw a man with a telescope
- → I saw with a telescope a man
4
5
1
2
3
I saw
a man
with a telescope

Figure 2: Example of a hypergraph encoding two different translations (one solid and one dotted) for the Spanish sentence "Vi a un hombre con un telescopio".

Due to the demanding temporal constraints inherent to any interactive environment, performing a full search each time the user validates a new prefix is unfeasible. The usual approach is to rely on a certain representation of the search space that includes the most probable translations of the source sentence. 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 representation can be reused for further completion requests.

Next, we introduce *hypergraphs*, the formalism chosen to represent the search space of both phrase-based and hierarchical systems (Section 4.1). Then, we describe the algorithms implemented to search for suffix completions in them (Section 4.2).

4.1 Hypergraphs

A hypergraph is a generalization of the concept of graph where the edges (now called hyperedges) may connect several nodes (hypernodes) at the same time. Formally, a hypergraph is a weighted acyclic graph represented by a pair $< \mathcal{V}, \mathcal{E} >$, where $\mathcal{V}$ is a set of hypernodes and $\mathcal{E}$ is a set of hyperedges. Each hyperedge $e \in \mathcal{E}$ connects a head hypernode and a set of tail hypernodes. The number of tail nodes is called the *arity* of the hyperedge and the arity of a hypergraph is the maximum arity of its hyperedges.

We can use hypergraphs to represent the derivations for a given CFG. Each hypernode represents a partial translation generated during the decoding process. Each ingoing hyperedge represents the rule with which the corresponding non-terminal was substituted. Moreover, hypergraphs can represent a whole set of possible translations. An example is

[2] Phrase-based systems follow a left-to-right generation order while hierarchical systems rely on a CYK-like order.

shown in Figure 2. Two alternative translations are constructed from the leave nodes (1, 2 and 3) up to the root node (6) of the hypergraph. Additionally, hypernodes and hyperedges may be shared among different derivations if they represent the same information. Thus, we can achieve a compact representation of the translation space that allows us to derive efficient search algorithms.

Note that *word-graphs* (Ueffing et al., 2002), which are used to represent the search space for phrase-based models, are a special case of hypergraphs in which the maximum arity is one. Thus, hypergraphs allow us to represent both phrase-based and hierarchical systems in a unified framework.

4.2 Suffix Search on Hypergraphs

Now, we describe a unified search process to obtain the suffix $\mathbf{t}_s$ that completes a prefix $\mathbf{p}$ given by the user according to the two IMT formulations (Equation (11) and Equation (17)) described in Section 2.

Given an hypergraph, certain hypernodes define a possible solution to the maximization defined in the two IMT formulations. Specifically, only those hypernodes that generate a prefix of a potential translation are to be taken into account[3]. The probability of the solution defined by each hypernode has two components, namely the probability of the SMT model (given by the language and translation models) and the probability of the error-correction model. On the one hand, the SMT model probability is given by the translation of maximum probability through the hypernode. On the other hand, the error-correction probability is computed between $\mathbf{p}$ and the partial translation of maximum probability actually covered by the hypernode. Among all the solutions defined by the hypernodes, we finally select that of maximum probability.

Once the best-scoring hypernode is identified, the rest of the translation not covered by it is returned as the suffix completion required in IMT.

5 Experimental Framework

The models and search procedure introduced in the previous sections were assessed through a series of IMT experiments with different corpora. These corpora, the experimental methodology, and the evaluation measures are presented in this section.

	EU (Es/En)		
	Train	Development	Test
Sentences	214K	400	800
Token	5.9M / 5.2M	12K / 10K	23K / 20K
Vocabulary	97K / 84K	3K / 3K	5K / 4K
	TED (Zh/En)		
	Train	Development	Test
Sentences	107K	934	1664
Token	2M / 2M	22K / 20K	33K / 32K
Vocabulary	42K / 52K	4K / 3K	4K / 4K

Table 1: Main figures of the processed EU and TED corpora. K and M stand for thousands and millions of elements respectively.

5.1 EU and TED corpora

We tested the proposed methods in two different translation tasks each one involving a different language pair: Spanish-to-English (Es–En) for the *EU* (Bulletin of the European Union) task, and Chinese-to-English (Zh–En) for the *TED* (TED[4] talks) task.

The EU corpora were extracted from the Bulletin of the European Union, which exists in all official languages of the European Union and is publicly available on the Internet. Particularly, the chosen Es–En corpus was part of the evaluation of the TransType2 project (Barrachina et al., 2009). The TED talks is a collection of recordings of public speeches covering a variety of topics, and for which high quality transcriptions and translations into several languages are available. The Zh–En corpus used in the experiments was part of the MT track in the 2011 evaluation campaign of the workshop on spoken language translation (Federico et al., 2011). Specifically, we used the `dev2010` partition for development and the `test2010` partition for test.

We process the Spanish and English parts of the EU corpus to separate words and punctuation marks keeping sentences truecase. Regarding the TED corpus, we tokenized and lowercased the English part (Chinese has no case information), and split Chinese sentences into words with the Stanford word

[3] For example, in Figure 2 the hypernodes that generate prefixes are those labeled with numbers 1 (“I saw”), 4 (“I saw with a telescope) and 6 (“I saw a man with a telescope” and “I saw with a telescope a man”).

[4] `www.ted.com`

segmenter (Tseng et al., 2005). Table 1 shows the main figures of the processed EU and TED corpora.

5.2 Model Estimation and User Simulation

We used the standard configuration of the `Moses` toolkit (Koehn et al., 2007) to estimate one phrase-based and one hierarchical model for each corpus; log-linear weights were optimized by minimum error-rate training (Och, 2003) with the development partitions. Then, the optimized models were used to generate the word-graphs and hypergraphs with the translations of the development and test partitions.

A direct evaluation of the proposed IMT procedures involving human users would have been slow and expensive. Thus, following previous works in the literature (Barrachina et al., 2009; González-Rubio et al., 2010), we used the references in the corpora to simulate the translations that a human user would want to obtain. Each time the system suggested a new translation, it was compared to the reference and the *longest common prefix* (LCP) was obtained. Then, the first non-matching character was replaced by the corresponding character in the reference and a new system suggestion was produced. This process is iterated until a full match with the reference was obtained.

Finally, we used this user simulation to optimize the value for the probability of edition p_e in the error-correction model (Section 3.1), and for the log-linear weights in the proposed IMT formulations. In this case, these values were chosen so that they minimize the estimated user effort required to interactively translate the development partitions.

5.3 Evaluation Measures

Different measures have been adopted to evaluate the proposed IMT approach. On the one hand, different IMT systems can be compared according to the effort needed by a human user to generate the desired translations. This effort is usually estimated as the number of actions performed by the user while interacting with the system. In the user simulation described above these actions are: looking for the next error and *moving the mouse pointer* to that position (LCP computation), and correcting errors with some *key strokes*. Hence, we implement the following IMT effort measure (Barrachina et al., 2009):

Key-stroke and mouse-action ratio (KSMR): number of key strokes plus mouse movements performed by the user, divided by the total number of characters in the reference.

On the other hand, we also want to compare the proposed IMT approach against a conventional CAT approach without interactivity, such as a decoupled post-edition system. For such systems, character-level user effort is usually measured by the *Character Error Rate* (CER). However, it is clear that CER is at a disadvantage due to the auto-completion approach of IMT. To perform a fairer comparison between post-edition and IMT, we implement a post-editing system with autocompletion. Here, when the user enters a character to correct some incorrect word, the system automatically completes the word with the most probable word in the task vocabulary. To evaluate the effort of a user using such a system, we implement the following measure proposed in (Romero et al., 2010):

Post-editing key stroke ratio (PKSR): using a post-edition system with word-autocompleting, number of user key strokes divided by the total number of reference characters.

The counterpart of PKSR in an IMT scenario is (Barrachina et al., 2009):

Key-stroke ratio (KSR): number of key strokes, divided by the number of reference characters.

PKSR and KSR are fairly comparable and the relative difference between them gives us a good estimate of the reduction in human effort that can be achieved by using IMT instead of a conventional post-edition system.

We also evaluate the quality of the automatic translations generated by the MT models with the widespread BLEU score (Papineni et al., 2002).

Finally, we provide both confidence intervals for the results and statistical significance of the observed differences in performance. Confidence intervals were computed by pair-wise re-sampling as in (Zhang and Vogel, 2004) while statistical significance was computed using the Tukey's HSD (honest significance difference) test (Hsu, 1996).

	EU		TED	
	WG	HG	WG	HG
1-best BLEU [%]	45.0	45.1	11.0	11.2
1000-best Avg. BLEU [%]	43.6	44.2	10.2	11.0

Table 2: BLEU score of the word-graphs (WG) and hypergraphs (HG) used to implement the IMT procedures.

IMT	**EU**		**TED**	
Setup	PB	HT	PB	HT
ISF	27.4±.5	26.5±.5*	53.0±.4	52.3±.4*
CSF	26.6±.5*	**25.1±.5**★	52.2±.4*	**50.8±.4**★

Table 3: IMT results (KSMR [%]) for the EU and TED tasks using the independent suffix formalization (ISF) and the conditioned suffix formalization (CSF). PB stands for phrase-based model and HT stands for hierarchical translation model. For each task, the best result is displayed boldface, an asterisk * denotes a statistically significant better result (99% confidence) with respect to ISF with PB, and a star ★ denotes a statistically significant difference with respect to all the other systems.

EU		**TED**	
PE	**IMT**	**PE**	**IMT**
PKSR [%]	KSR [%]	PKSR [%]	KSR [%]
27.1	**14.1 (48%)**	40.8	**29.7 (27.2%)**

Table 4: Estimation of the effort required to translate the test partition of the EU and TED tasks using post-editing with word-completion (PE) and IMT under the independent suffix formalization (IMT). We used hierarchical MT in both approaches. In parenthesis we display the estimated effort reduction of IMT with respect to PE.

6 Results

We start by reporting conventional MT quality results to test if the generated word-graphs and hypergraphs encode translations of similar quality. Table 2 displays the quality (BLEU (Papineni et al., 2002)) of the automatic translations generated for the test partitions. The lower 1-best BLEU results obtained for TED show that this is a much more difficult task than EU. Additionally, the similar average BLEU results obtained for the 1000-best translations indicate that word-graphs and hypergraphs encode translations of similar quality. Thus, the IMT systems that use these word-graphs and hypergraphs can be compared in a fair way.

Then, we evaluated different setups of the proposed IMT approach. Table 3 displays the IMT results obtained for the EU and TED tasks. We report KSMR (as a percentage) for the independent suffix formalization (ISF) and the conditioned suffix formalization (CSF) using both phase-based (PB) and hierarchical (HT) translation models. The KSMR result of ISF using a phrase-based model can be considered our baseline since this setup is comparable to that used in (Barrachina et al., 2009). Results for HT consistently outperformed the corresponding results for PB. Similarly, results for CSF were consistently better than those for ISF. More specifically, no statistically significant difference were found between ISF with HT and CSF with PB but both statistically outperformed the baseline (ISF with PB). Finally, CSF with HT statistically outperformed the other three configurations reducing KSMR by ~ 2.2 points with respect to the baseline. We hypothesize that the better results of HT can be explained by its more efficient representation of word reordering. Regarding the CSF, its better results are due to the better suffixes that can be obtained by taking into account the actual prefix validated by the user.

Finally, we compared the estimated human effort required to translate the test partitions of the EU and TED corpora with the best IMT configuration (independent suffix formalization with hierarchical translation model) and a conventional post-editing (PE) CAT system with word-completion. That is, when the user corrects a character, the PE system automatically proposes a different word that begins with the given word prefix but, obviously, the rest of the sentence is not changed. According to the results, the estimated human effort to generate the error-free translations was significantly reduced with respect to using the conventional PE approach. IMT can save about 48% of the overall eastimated effort for the EU task and about 27% for the TED task.

7 Summary and Future Work

We have proposed a new IMT approach that uses hierarchical SMT models as its underlying translation technology. This approach is based on a statistical formalization previously described in the literature that includes stochastic error correction. Additionally, we have proposed a refined formalization that improves the quality of the IMT suffixes by taking into account the prefix validated by the user. Moreover, since word-graphs constitute a particular case of hypergraphs, we are able to manage both phrase-based and hierarchical translation models in a unified IMT framework.

Simulated results on two different translation tasks showed that hierarchical translation models outperform phrase-based models in our IMT framework. Additionally, the proposed alternative IMT formalization also allows to improve the results of the IMT formalization previously described in the literature. Finally, the proposed IMT system with hierarchical SMT models largely reduces the estimated user effort required to generate correct translations in comparison to that of a conventional post-edition system. We look forward to corroborating these result in test with human translators.

There are many ways to build on the work described here. In the near future, we plan to explore the following research directions:

- Alternative IMT scenarios where the user is not bounded to correct translation errors in a left-to-right fashion. In such scenarios, the user will be allowed to correct errors at any position in the translation while the IMT system will be required to derive translations compatible with these isolated corrections.

- Adaptive translation engines that take advantage of the user's corrections to improve its statistical models. As the translator works and corrects the proposed translations, the translation engine will be able to make better predictions. One of the first works on this topic was proposed in (Nepveu et al., 2004). More recently, Ortiz-Martínez et al. (2010) described a set of techniques to obtain an incrementally updateable IMT system, solving technical problems encountered in previous works.

- More sophisticated measures to estimate the human effort. Specifically, measures that estimate the cognitive load involve in reading, understanding and detecting an error in a translation (Foster et al., 2002), in contrast KSMR simply considers a constant cost. This will lead to a more accurate estimation of the improvements that may be expected by a human user.

Acknowledgments

Work supported by the European Union 7$^{\text{th}}$ Framework Program (FP7/2007-2013) under the CasMaCat project (grans agreement n$^{\text{o}}$ 287576), by Spanish MICINN under grant TIN2012-31723, and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014).

References

Alfred V. Aho and Jeffrey D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and Systems Science*, 3(1):37–56, February.

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

Richard Bellman. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition.

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

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270.

M. Federico, L. Bentivogli, M. Paul, and S. Stüker. 2011. Overview of the iwslt 2011 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 11–20.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1998. Target-text mediated interactive machine translation. *Machine Translation*, 12(1/2):175–194, January.

George Foster, Philippe Langlais, and Guy Lapalme. 2002. User-friendly text prediction for translators. In *Proceedings of the 2002 conference on Empirical methods in natural language processing - Volume 10*, pages 148–155.

Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2010. Balancing user effort and translation error in interactive machine translation via confidence measures. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 173–177.

Jason Hsu. 1996. *Multiple Comparisons: Theory and Methods*. Chapman and Hall/CRC.

Pierre Isabelle and Ken Church. 1998. *Special issue on: New tools for human translators*, volume 12. Kluwer Academic Publishers, January.

Frederick Jelinek. 1997. *Statistical methods for speech recognition*. MIT Press.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics, demonstration session*, June.

Philippe Langlais and Guy Lapalme. 2002. TransType: development-evaluation cycles to boost translator's productivity. *Machine Translation*, 17(2):77–98, September.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February.

Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George Foster. 2004. Adaptive language and translation models for interactive machine translation. In *Proceedings of the conference on Empirical Methods on Natural Language Processing*, pages 190–197.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302.

Franz Josef Och, Richard Zens, and Hermann Ney. 2003. Efficient search for interactive statistical machine translation. In *Proceedings of the European chapter of the Association for Computational Linguistics*, pages 387–393.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167. Association for Computational Linguistics.

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

Daniel Ortiz-Martínez. 2011. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation*. Ph.D. thesis, Universitat Politècnica de València. Advisors: Ismael García Varea and Francisco Casacuberta.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318. Association for Computational Linguistics.

Veronica Romero, Alejandro H. Toselli, and Enrique Vidal. 2010. Character-level interaction in computer-assisted transcription of text images. In *Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition*, pages 539–544.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-dependency statistical machine translation. *Computational Linguistics*, 36(4):649–671, December.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Nicola Ueffing, Franz J. Och, and Hermann Ney. 2002. Generation of word graphs in statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 156–163.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of The international Conference on Theoretical and Methodological Issues in Machine Translation*.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141.