CHAPTER 6

Interactive Machine Translation

WITH CONTRIBUTION OF:
Jorge Civera, Jesús González-Rubio and Daniel Ortiz-Martínez

Achieving high-quality translation between any pair of languages is not possible with the current *machine-translation* (MT) technology being necessary a human post-editing of the outputs of the MT system. Therefore, MT is a suitable area to apply the *interactive pattern recognition* (IPR) framework and this application has led to the nowadays known as *interactive machine translation* (IMT). IMT can predict the translation of a given source sentence, and the human translator can accept or correct some of the errors. The text amended by the human translator can be used by the system to suggest new improved translations with the same translation models in an iterative process until the whole output is accepted by the human.

As in other areas where IPR is being applied, IMT offers a nice framework for adaptive learning. The consolidated translations obtained through the successive steps of the interaction process can easily be converted into new, fresh, training data, useful for dynamically adapting the system to the changing environment. On the other hand, IMT also allows to take advantage of some available multi-modal interfaces to increase of the productivity of high-quality translations. Multimodal interfaces and adaptive learning in IMT will be covered in Ch. 7 and 8, respectively

Chapter Outline

6.1 Introduction

The application of statistical pattern recognition techniques to the field of *machine translation* (MT) has allowed the development of new MT systems with less effort than was previously required under the formerly dominant rule-based paradigm [22]. These systems (that are known as *statistical MT* -SMT- systems) together with the *memory-based* ones constitute the *data-driven* approach to MT. However, the quality of the translations produced by any (statistical, memory-based or rule-based) MT system remains below that of human translation. This quality could be enough for many applications, but for other, the output of the MT systems has to be revised in a *post-editing* phase. An alternative to the use of pure post-editing is the approach proposed in the TransType project [16, 25, 26] and its successor TransType2 (TT2) [9, 2]. In this approach, a full-fledged MT engine is embedded in an interactive editing environment and used to generate suggested completions of each target sentence being translated. These completions may be accepted or amended by the translator; but once validated, they are exploited by the MT engine to produce further, hopefully improved suggestions. This new approach is known as *interactive machine translation* (IMT). 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 idea was extended to complete full target sentences in the TT2 project. This interactive approach offers a significant advantage over traditional post-editing where there is no way for the system to benefit from the corrections of the user.

Interactivity in translation (more precisely, in *computer-assisted translation* -CAT-) has been explored for a long time to solve different types of ambiguities [2]. However, there are only few research groups that have published, to our knowledge, contributions in this IMT topic. As we have mentioned, the first publications are related with the TransType project [16, 25, 17, 26, 29, 34, 37]. The second group of publications are around the TransType2 project [15, 32, 12–14, 3, 38, 2, 9]. More recently other research groups have started to work on this topic [23].

In this section, we present a summary of the state-of-the-art in SMT. Sec. 6.2 is devoted to the applications of IPR in MT. The specific search problem in IMT is presented in Sec. 6.3. In Sec. 6.4.3 we discuss the evaluation measures that will be used in the experimental framework. The adopted tasks and experimental settings, and the results obtained are presented in Sec. 6.4. All the aspects related with adaptability and multi-modality in IMT are introduced in the next chapters.

6.1.1 Statistical Machine Translation

SMT is based on the application of the Bayes' decision rule to the problem of conversion of a *source* sentence x from a source language $\mathcal{X}$ to a *target* sentence h from a target language $\mathcal{H}$. This decision rule can be stated as the search for a target sentence $\hat{h}$ that maximize the posterior probability that a sentence h is a translation of a given x [5, 6]:

$$\hat{h} = \arg\max_{h} \Pr(h|x) \tag{6.1}$$

The state-of the art in SMT is based on *bilingual segments* or *bilingual phrases* as translation units and *log-linear* models to approach $\Pr(h|x)$ [22]. Bilingual phrases are pairs of word sequences $(\tilde{x}, \tilde{h})$ in which all words within the source-language phrase $\tilde{x}$ are aligned only to words of the target-language phrase $\tilde{h}$ and vice versa [22]. On the other hand, log-linear models [31] are combinations of N different feature functions $f_i(x, h)$ for $1 \leq i \leq N$:

$$\Pr(h|x) \approx P(h|x;\lambda) = \frac{\exp \sum_{i=1}^{N} \lambda_i f_i(x,h)}{\sum_{h'} \exp \sum_{i=1}^{N} \lambda_i f_i(x,h')} \tag{6.2}$$

A feature function $f_i(x, h)$ [22] can be 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.

Some of these feature functions are based on the segmentation of the pair (x, h) in terms of a sequence of phrases $\tilde{x}_1, \ldots, \tilde{x}_K$ ($x = \tilde{x}_1 \ldots \tilde{x}_K$) and $\tilde{h}_1, \ldots, \tilde{h}_K$ ($h = \tilde{h}_1 \ldots \tilde{h}_K$) for a given K. If the correspondence (*alignment*) between source and target phrases is represented as a function $a : \{1, \ldots, K\} \rightarrow \{1, \ldots, K\}$, the Eq. (6.1) can be rewritten using a modified version of Eq. (6.2) as:

$$\hat{h} = \arg\max_{h} \max_{a} \sum_{i} \lambda_i f_i(x, h, a) \tag{6.3}$$

One of these feature functions can represent the direct translation:

$$f_i(x, h, a) = \sum_{k=1}^{K} \Big(\log p(a_k|a_{k-1}) + \log p(\tilde{h}_k|\tilde{x}_{a_k}) \Big) \tag{6.4}$$

where $p(a_k|a_{k-1})$ is the probability that a source phrase in position k is aligned with a target phrase in position a_k, given that a previous source phrase in position $k-1$ was aligned with a target phrase in position a_{k-1} and $p(\tilde{h}_k|\tilde{x}_{k'})$ is the probability that a target phrase $\tilde{h}_k$ is the translation of a given source phrase $\tilde{x}_{k'}$. Another feature function is based on a target language model, typically a n-gram model (trigrams for example for a target sentence of length J):

$$f_i(x, h, a) = \sum_{j=1}^{J} \log p(h_j|h_{j-2}, h_{j-1}) \tag{6.5}$$

Other feature functions are based on the inverse version of Eq. (6.4) and on other target language models and diverse (target and/or source) length models. There is an interesting feature function that is based on a language model (trigrams for example and with $a(j) = j$) of bilingual phrases:

$$f_i(x, h, a) = \sum_{k=1}^{K} \log p(\tilde{x}_k, \tilde{h}_k|\tilde{x}_{k-2}, \tilde{h}_{k-2}, \tilde{x}_{k-1}, \tilde{h}_{k-1}) \tag{6.6}$$

This model can also be efficiently implemented with *stochastic finite-state transducers* (SFSTs) [11, 10] or as another feature in the log-linear modeling [28].

In the learning phase, all bilingual phrases are extracted from a bilingual training corpus and the normalized counts of how often a bilingual phrase occurred in the aligned training corpus are computed [33, 24, 22]. The parameters of the n-grams are estimated by a counting process on a target training set. On the other hand, the weights of the log-linear combination in Eq. (6.2) are computed by means of *minimum error rate training* (MERT) [30]. In the case that SFSTs are adopted, the Grammatical Inference Algorithms for Transducer Inference (GIATI) algorithm can be used [11].

The search for the best translation of a given source sentence x in is carried out by producing the target sentence in left-to-right order using the log-linear model in Eq. (6.2). At each step of the generation algorithm, a set of active hypotheses are maintained and one of them is chosen for extension. A segment of the target language is then added to the chosen hypothesis and its costs get updated [31, 22]. If SFSTs are adopted, the Viterbi algorithm can used for the generation of the target sentence [10]. In both cases, the search space can be huge and prunning techniques have to be used.

6.2 Interactive Machine Translation

The systems described in the Sec. 6.1.1 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 IMT paradigm. This approach is exemplified in Fig. 6.1. Let us suppose that a source English sentence x ="Click OK to close the print dialog" is to be translated into a target Spanish sentence h. Initially, with no user information, the system provides a complete translation suggestion (s ="Haga clic para cerrar el diálogo de impresión"). From this translation, the user marks a prefix as correct ("Haga clic") and begins to type the rest of the target sentence. Depending on the system or the user's preferences, the new input can be the next word or some letters from it (in our example, the input is the next correct word "en"). A new target prefix p is then defined by the previously validated prefix together with the new input the user has just typed (p ="Haga clic en"). The system then generates a new suffix s to complete the translation: "ACEPTAR para cerrar el diálogo de impresión". The interaction continues with a new validation followed, if necessary, by new input from the user, and so on, until such time as a complete and satisfactory translation is obtained.

In this problem, we can apply the concepts and ideas that have been developed in Sec. 1.4.2 in the algorithm IPR-History of Sec. 1.3.2. More specifically, we can use the concepts of prefix and suffix introduced in the left-to-right interactive-predictive processing: Given a source sentence x and a target prefix p validated by the human, the optimization problem can be stated as the search for a target suffix s that completes p as a translation of the source sentence x:

$$\hat{s} = \arg\max_{s} \Pr(s|x, p) \tag{6.7}$$

This equation can be rewritten as

$$\hat{s} = \arg\max_{s} \Pr(p, s|x) \tag{6.8}$$

	Input	(x)	Click OK to close the print dialog
0	System	$(\hat{s})$	*Haga clic para cerrar el diálogo de impresión*
1	User	(p)	Haga clic **en**
	System	$(\hat{s})$	*ACEPTAR para cerrar el diálogo de impresión*
2	User	(p)	Haga clic en ACEPTAR para cerrar el **cuadro**
	System	$(\hat{s})$	*de diálogo de impresión*
3	User	(p)	Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión **#**
	Output	(h)	Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión

Figure 6.1: An example of IMT 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 p, from which the system suggests a suffix $\hat{s}$. Then the user accepts a part of this suffix (in black colour) and types some key-strokes (in red colour), possibly in order to amend the remaining part of s. This produces a new prefix, composed by the prefix from the previous iteration and the accepted and typed text, to be used as 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 h, text that has been typed by the user is underlined.

Since $p\ s = h$, this equation is very similar to Eq. (6.1). The main difference is that the maximization search now is performed over the set of suffixes s that complete p instead of complete sentences (h in Eq. (6.1)). This implies that we can use the same models if the search procedures are adequately modified [2].

The optimization problem in IMT have been reduced as a search problem constrained by the prefix, obviously, there can be another alternatives, but this one has the advantage that in this approach we used the same models as for SMT and therefore we use the same training algorithm as for SMT [2]. On the other hand, the search for IMT is similar as for SMT but constrained by a fixed prefix in each iteration. This search can be carried out by a modification of the available search algorithms [2]. However, high speed is needed because typically a new system hypothesis must be produced in real time after each user keystroke [32, 2], therefore, we use word graph that represents all (or a selected) possible translations of the given source sentence.

6.2.1 Interactive Machine Translation with Confidence Estimation

Under the IMT paradigm, the user is asked to mark a correct prefix and, possibly, type some corrections for each suffix provided by the system. To interact with the system, the user makes use of his knowledge about the languages being translated, but, potentially, user effort reductions could be achieved if information about the correctness of the suffixes provided by the system is made available to the user. This information can be derived by estimating the confidence the system has on their predicted suffixes, as introduced in Sec. 1.5.2. For a given source sentence x and a validated target prefix p we compute the confidence measure $CM(\hat{s}, x, p)$ for the target suffix $\hat{s}$ generated.

Confidence estimation have been extensively studied for other *natural language processing* (NLP) applications and more recently have been applied to SMT [18, 4, 39, 35, 40]. Confidence information have been previously used in IMT to improve translation prediction accuracy [17, 18, 39]. Alternatively, confidence information can be used not only to improve the translations provided by the system, but also to reduce the user effort.

The use of confidence information within the IMT scenario results in a modification of the interaction protocol. In the IMT scenarios discussed so far, the operator was assumed to systematically supervise each system suffix and find the point where the next translation error appears. As discussed in Sec. 1.4.1, within this "passive" protocols the system just waits for the human feedback, without taking into account how the supervision is performed by the user. In contrast, in an "active" protocol, the system is in charge of taking decisions about what needs user supervision (See Sec. 1.4.3). Suffix confidence measures can be used to estimate which hypothesis may be worth asking for user supervision in order to optimise the overall human-computer performance.

According to this "active" protocol, we define an alternative IMT scenario where not all the sentences are interactively translated by the user. Specifically, only those suffixes classified as incorrect, according to a confidence measure, are interactively amended by the user [20, 19]. Therefore, the quality of the final translations may depend on the system ability to select appropriate suffixes for supervision. However, this "active" interaction may provide a better trade-off between overall human interaction effort and translation accuracy.

This "active" protocol can be seen as a generalisation of the IMT scenario in which confidence estimation acts as a regulator of the effort required for the user. Depending on the confidence threshold defined, the behaviour of the system can range from a fully automatic SMT system where all suffixes are considered to be correct, to a conventional IMT system where all suffixes are considered to be incorrect.

Confidence Measure for IMT

We estimate the reliability of the suffixes generated by the system by combining the confidence scores of their individual words. We choose a word confidence measure based on the IBM model 1 [6]. The confidence score of a word $\hat{s}_i$ of the suffix $\hat{s}$ generated from the source sentence x given the prefix p from Eq. (6.7) is computed as:

$$CM(\hat{s}_i, x, p) \approx CM(\hat{s}_i, x) = \max_{0 \leq j \leq J} P(\hat{s}_i | x_j) \tag{6.9}$$

where $P(\hat{s}_i|x_j)$ is a bilingual lexicon probability [6] and x_0 is the empty source word.

We choose this confidence measure instead of using the posterior probability due to response-time constrains. The confidence score based on the simplest model proposed in [6] (Model 1) is much faster to compute than applying the forward-backward algorithm as described in Sec. 1.5.2. Moreover, it performs similarly to the word confidence measures based on word-graphs as shown in [4, 40, 35].

The confidence measure for the full suffix $\hat{s}$ is computed as the ratio of its words classified as correct by the word confidence measure. A word $\hat{s}_i$ is classified as correct if its confidence score

$CM(\hat{s}_i, x)$ exceeds a word classification threshold τ_w.

$$CM(\hat{s}, x, p) \approx CM(\hat{s}, x) = \frac{|\{\hat{s}_i \mid CM(\hat{s}_i, x) > \tau_w\}|}{|\hat{s}|} \tag{6.10}$$

Each suffix is classified as either correct or incorrect depending on whether its confidence score exceeds or not a suffix classification threshold τ_s. It is worth of notice that with a threshold value $\tau_s = 0.0$ all the suffixes will be classified as correct whereas with a threshold value $\tau_s = 1.0$ all the suffixes will be classified as incorrect.

6.3 Search in Interactive Machine Translation

As mentioned above, the search problem in IMT can be seen as a search constrained by the prefix p validated by the user. Real-time user interaction dictates the need of efficient search techniques, such as the word-graph representation and Viterbi algorithm presented in Sec. 1.5.1.

Analogously to the search procedure in CATTI and CAST (See Sec. 2.5), the first step is to generate a word graph as a pruned version of the search space for the translation of the source sentence x [2]. Once the word graph is constructed, error-correcting parsing is used to accommodate the user-validated prefix to those prefixes available in the word graph. This step is followed by a Viterbi suffix search to provide the most probable completion.

6.3.1 Word-graph Generation

For each source sentence, a word graph representing possible translations is generated. This word graph is generated once for each source sentence, so it is repeatedly used to find the completions of all the different prefixes provided by the user. Using the word graph in such a way makes the system be able to interact with the user under tight real-time constraints [2].

A word graph may also be understood as a weighted directed acyclic graph, in which each node represents a partial translation hypothesis, and each edge is labeled with the word or the segment of the target sentence being expanded and is weighted according to the underlying models. Indeed, 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. An example of a word graph for the source Spanish sentence “seleccionar el siguiente” is shown in Fig. 6.2.

However, due to the pruning performed in the word-graph generation for efficiency and response-time constraints, the word graph only contains a subset of the possible translations. Moreover, it is also possible that the user incorporates words unknown by the system. For these reasons, it may occur that the prefix validated by the user is not present in the word graph. This problem requires the application of error-correcting parsing to allow for user prefixes that may not exist in the word-graph.