UNIVERSIDAD
POLITECNICA
DE VALENCIA

Departamento de Sistemes Informáticos y Computación

IARFID Master Thesis:

ITGs for Phrase Extraction and Mouse Actions in IMT

Author:
Germán Sanchis-Trilles

Advisor:
Francisco Casacuberta
Joan Andreu Sánchez

November 17, 2008

Contents

Abstract

This thesis presents two main contributions in the fields of Statistical Machine Translation and Interactive Machine Translation.

In the field of Statistical Machine Translation, the efforts have been focused on obtaining high quality, linguistically motivated phrase pairs by means of Statistical Inversion Transduction Grammars. By using a SITG for parsing a bilingual corpus, spans are defined over both input and output strings, yielding the possibility of considering these spans as translations of each other. By doing so, phrase tables can be built from the bilingual corpus and fed to an off-the-shelf Statistical Machine Translation decoder. Moreover, novel syntax-based models are introduced in this thesis, and experimental results are shown which back up the inclusion of such models into the standard phrase translation table. Since these models are inherent to SITGs, they cannot be included into other standard phrase-based models.

In the field of Interactive Machine Translation, a new interface between the user and the machine is proposed. By considering the Mouse Actions the user performs as an important input source for the system, it is shown that important and consistent performance gains may be achieved. These gains come in some cases at the cost of having the user ask for new suffix hypotheses, but in other cases these gains come at no cost, hence yielding true improvements to the state of the art.

Overview

This Masters Thesis is structured into five chapters. The first two intend to be an introduction to the main aspects of Statistical Machine Translation and Interactive Machine Translation. Then, in Chapter 4, an improvement of the current machine–human interaction is proposed, leading to a significant improvement over the state of the art. This chapter finishes the first part of this thesis, devoted to Interactive Machine Translation. Then, in Chapter 2, the Stochastic Inversion Transduction Grammars are defined, along with the way in which they can be used for phrase extraction and the experiments carried out in this framework. In the last chapter, other work done during the period of elaboration of this thesis is presented.

Chapter 1

Statistical Machine Translation

1.1 Word-Based Statistical Machine Translation

Machine Translation (MT) is a research field of great importance in the European Community, where language plurality implies both a very important cultural richness and not negligible obstacle towards building a unified Europe. Because of this, a growing interrest on MT has been shown both by politicians and research groups, which become more and more specialised in this field.

On the other hand, Statistical Machine Translation (SMT), systems have proved in the last years to be an important alternative to rule-based MT systems, being even able of outperforming commercial machine translation systems in the tasks they have been trained on. Moreover, the development effort behind a rule-based machine translation system and an SMT system is dramatically different, the latter being able to adapt to new language pairs with little or no human effort, whenever suitable corpora are available.

The grounds of modern SMT, the pattern recognition approach to Machine Translation, were established in [Brown et al., 1993], where the problem of machine translation was defined as following: given a sentence $\mathbf{x} = x_1...x_j...x_{|x|}$ from a certain source language, an adequate sentence $\hat{\mathbf{y}} = y_1...y_i...y_{|y|}$ that maximises the posterior probability is to be found. Such a statement can be specified with the following formula:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, Pr(\mathbf{y}|\mathbf{x}) \tag{1.1}$$

Applying the Bayes theorem on this definition, one can easily reach the next formula

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, \frac{Pr(\mathbf{y}) \cdot Pr(\mathbf{x}|\mathbf{y})}{Pr(\mathbf{x})} \tag{1.2}$$

and, since we are maximising over t, the denominator can be neglected, arriving to

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, Pr(\mathbf{y}) \cdot Pr(\mathbf{x}|\mathbf{y}) \tag{1.3}$$

where $Pr(\mathbf{y}|\mathbf{x})$ has been decomposed into two different probabilities: the *statistical language model* of the target language $Pr(\mathbf{y})$ and *the (inverse) translation model* $Pr(\mathbf{x}|\mathbf{y})$.

Although it might seem odd to model the probability of the source sentence given the target sentence, this decomposition has a very intuitive interpretation: the translation model $Pr(\mathbf{x}|\mathbf{y})$ will capture the word or phrase relations between both input and output language, whereas the language model $Pr(\mathbf{y})$ will ensure that the output sentence is a well-formed sentence belonging to the target language.

A great variety of models have been proposed in order to model the probability $Pr(\mathbf{y}|\mathbf{x})$ adequately. In [Brown et al., 1993], five *alignment* models (known as IBM models) were already described, in which the correspondance between source and target sentences was estabilised by means of a hidden alignment variable $\mathbf{a} = a_1...a_i...a_{|y|}$, which was defined as a function over the target words. Being a function, each target word is assigned a source word which is evidenced as being a good translation. However, and in order to account for possible target words with no mapping in the source sentence, the artificial zero (or NULL) position was introduced, yielding

$$Pr(\mathbf{y}|\mathbf{x}) = \sum_{a \in \mathcal{A}(\mathbf{x},\mathbf{y}))} p(\mathbf{y}, \mathbf{a}|\mathbf{x}), \tag{1.4}$$

where $\mathcal{A}(\mathbf{x}, \mathbf{y}))$ denotes the set of all possible alignments between $\mathbf{x}$ and $\mathbf{y}$.

In practise, the direct modelling of the posterior probability $Pr(\mathbf{y}|\mathbf{x})$ has been widely adopted. To this purpose, different authors [Papineni et al., 1998, Och and Ney, 2002] propose the use of the so-called log-linear models, where the decision rule is given by the expression

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x}, \mathbf{y}) \tag{1.5}$$

where $h_m(\mathbf{x}, \mathbf{y})$ is a score function representing an important feature for the translation of $\mathbf{x}$ into $\mathbf{y}$, M is the number of models (or features) and λ_m are the weights of the log-linear combination. Under this perspective, Equation 1.1 can be seen as a special case of Equation 1.5, where $Pr(\mathbf{y}|\mathbf{x})$ and $Pr(\mathbf{y})$ are the important features, and there are two λ_m, both set to 1.

1.2 Phrase-Based Statistical Machine Translation

One of the most popular instantiations of log-linear models is that including Phrase-Based (PB) models [Tomas and Casacuberta, 2001, Marcu and Wong, 2002, Zens et al., 2002, Zens and Ney, 2004], which have proved to provide a very efficient framework for MT. Computing the translation probability of a given *phrase*, i.e. a sequence of words, and hence introducing information about context, these SMT systems seem to have mostly outperformed single-word models, quickly evolving into the predominant technology in the state of the art [Koehn and Monz, 2006a, Callison-Burch et al., 2007, Fordyce, 2007].

1.2.1 The model

The derivation of PB models stems from the concept of bilingual segmentation, i.e. sequences of source words and sequences of target words. It is assumed that only segments of contiguous words are considered, the number of source segments being equal to the number of target segments (say K) and each source segment being aligned with only one target segment and vice versa.

Let I and J be the lengths of t and s respectively[1]. Then, the bilingual segmentation is formalised through two segmentation functions: μ for the target segmentation ($\mu_1^K : \mu_k \in \{1, 2, \ldots, I\},\ 0 < \mu_1 \leq \mu_2 \leq \cdots \leq \mu_k = I$) and γ for the source segmentation ($\gamma_1^K : \gamma_k \in \{1, 2, \ldots, J\},\ 0 < \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_k = J$). The alignment between segments is introduced through the alignment function α ($\alpha_1^K : \alpha_k \in \{1, 2, \ldots, K\}, \alpha(k) = \alpha(k')$ iff $k = k'$).

By assuming that all possible segmentations of s in K phrases and all possible segmentations of t in K phrases have the same probability independent of K, then $p(s|t)$ can be written as:

$$p(s|t) \quad \propto \quad \sum_{K} \sum_{\mu_1^K} \sum_{\gamma_1^K} \sum_{\alpha_1^K} \prod_{k=1}^{K} p(\alpha_k|\ \alpha_{k-1}) \cdot p(s_{\gamma_{\alpha_k-1}+1}^{\gamma_{\alpha_k}} | t_{\mu_{k-1}+1}^{\mu_k}) \tag{1.6}$$

where the distortion model $p(\alpha_k|\ \alpha_{k-1})$ (the probability that the target segment k is aligned with the source segment α_k) is usually assumed to depend only on the previous alignment α_{k-1} (first order model).

1.2.2 Learning phrase-based models

Ultimately, when learning a PB model, the purpose is to compute a *phrase translation table*, in the form

$$\{(s_j \ldots s_{j'}), (t_i \ldots t_{i'}), p(s_j \ldots s_{j'} | t_i \ldots t_{i'})\}$$

[1] Following a notation used in [Brown et al., 1993], a sequence of the form $z_i, \ldots, z_j$ is denoted as z_i^j. For some positive integers N and M, the image of a function $f : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, M\}$ for n is denoted as f_n, and all the possible values of the function as f_1^N

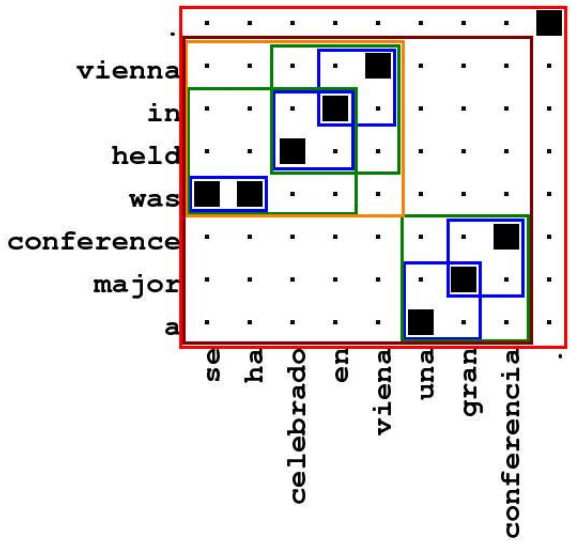

Figure 1.1: Example of how consistent phrases are extracted from a word alignment.

where the first term represents the input (source) phrase, the second term represents the output (target) phrase and the last term is the probability assigned by the model to the given phrase pair.

In the last years, a wide variety of techniques to produce PB models have been researched and implemented [Koehn et al., 2003]. Firstly, a direct learning of the parameters of the equation $p(s_j^{j'}|t_i^{i'})$ was proposed [Tomas and Casacuberta, 2001, Marcu and Wong, 2002]. Other approaches have been suggested, exploring more linguistically motivated techniques [Sánchez and Benedí, 2006b, Watanabe et al., 2003]. However, the one technique which has been more widely adopted is the one developed by [Zens et al., 2002], in which all phrase pairs coherent with a given word alignment are extracted. In most cases, one of the IBM alignments described in Section 1.1 is used for this purpose. Since these word alignments are very restrictive because each target word is assigned only zero or one source words, source-to-target and target-to-source alignments are combined heuristically. This procedure is often called *symmetrization*. Once this is done, the set of phrases consistent with the symmetrized word alignments is extracted from every sentence pair in the training set. An illustration of how this is done can be seen in Figure 1.1

Most typically, the different features that are included into the translation model are:

- Inverse translation probability, given by the formula

$$p(\mathbf{t}|\mathbf{s}) = \frac{C(\mathbf{s},\mathbf{t})}{C(\mathbf{s})} \tag{1.7}$$

  where $C(\mathbf{s},\mathbf{t})$ is the number of times segments $\mathbf{s}$ and $\mathbf{t}$ were extracted throughout the whole corpus, and $C(\mathbf{s})$ is the count for phrase $\mathbf{s}$.

- Direct translation probability, $p(\mathbf{s}|\mathbf{t})$, which is obtained analogously.

- Inverse and direct lexicalized features, which attempt to account for the lexical soundness of each phrase pair, estimating how well each of the words in one language translates to each of the words in the other language. These lexicalized features were defined in [Zens et al., 2002]

- A constant feature, or *phrase penalty*, whose purpose is to avoid the use of many small phrases in decoding time, and favour the use of longer ones.

1.2.3 Decoding in phrase-based models

Once a SMT system has been trained, a decoding algorithm is needed. Different search strategies have been suggested to define the way in which the search space is organised. Some authors [Ortiz et al., 2003, Germann et al., 2001] have proposed the use of an $A^\star$ algorithm, which adopts a *best-first* strategy that uses a stack (priority-queue) in order to organise the search space. On the other hand, a *depth-first* strategy was also suggested in [Berger et al., 1996], using a set of stacks to perform the search.

1.3 Corpora: Europarl

The Europarl corpus [Koehn, 2005], built from the proceedings of the European Parliament, is a reference corpus in SMT, and has been used in several MT campaigns. For this reason, most of the experiments conducted in this thesis were performed on the partition of this corpus established for the Workshop on Statistical Machine Translation of the NAACL 2006 [Koehn and Monz, 2006b], where the language pairs involved were German–English, Spanish–English and French–English. The curpus is divided into four separate sets: one for training, one for development, one for test and another test set which was the one used in the workshop for the final evaluation. This test set will be referred to as "Test", whereas the test set provided for evaluation purposes outside the final evaluation will be referred to as "Devtest". It must be noted that the Test set included a surprise out-of-domain subset, and hence the translation quality on this set will be significantly lower. The characteristics of the corpus can be seen in Table 1.1. It might seem surprising that the average sentence length in the training set is significantly lower than in the rest of the subsets. This is due to the fact that, for the competition, the training corpus pruned to contain only those sentences with a maximum length of 40, whereas this restriction was not imposed on the other subsets.

Table 1.1: Characteristics of Europarl for each of the subcorpora. OoV stands for "Out of Vocabulary" words, Dev. for Development, K for thousands of elements and M for millions of elements.

		German	English	Spanish	English	French	English
Training	Sentences	751M		730M		688M	
	Running words	15.3M	16.1M	15.7M	15.2M	15.6M	13.8M
	Average length	20.3	21.4	21.5	20.8	22.7	20.1
	Vocabulary size	195K	66K	103K	64K	80K	62K
Dev.	Sentences	2000		2000		2000	
	Running words	55K	59K	61K	59K	67K	59K
	Average length	27.6	29.3	30.3	29.3	33.6	29.3
	OoV	432	125	208	127	144	138
Devtest	Sentences	2000	2000	2000	2000	2000	
	Running words	54K	58K	60K	58K	66K	58K
	Average length	27.1	29.0	30.2	29.0	33.1	29.3
	OoV	377	127	207	125	139	133
Test	Sentences	3064		3064		3064	
	Running words	82K	85K	92K	85K	101K	85K
	Average length	26.9	27.8	29.9	27.8	32.9	27.8
	OoV	1020	488	470	502	536	519

Chapter 2

Stochastic Inverse Transduction Grammars

Being closely related to context-free grammars, Stochastic Inverse Transduction Grammars [Wu, 1997] specify a subset of syntax directed stochastic grammars for translation. Analysing two strings simultaneously, SITGs may be used to extract bilingual segments from a parallel corpus while taking into account syntax-motivated restrictions.

2.1 SITG definition

The definition of SITG relies on the concept of *bilingual parsing*, where the input is a bilingual sentence pair, rather than a single monolingual sentence. As such, their aim is not to obtain syntactic derivation trees, but to extract structure from the input data, and see how the output data relates to this structure.

A SITG in Chomsky Normal Form is defined as a tuple (N,S,W_1,W_2,R,p), where N is a finite set of non-terminals, $S \in N$ is the initial symbol or axiom, W_1 is a finite set of terminal symbols pertaining to the first language, W_2 is a set of terminals belonging to the second language, R is a set of rules in the form $A \rightarrow x/\epsilon$, $A \rightarrow \epsilon/y$ or $A \rightarrow x/y$, with $A \in N$, $x \in W_1$ and $y \in W_2$ and p defines the probability of a given rule.

On the other hand, derivation rules can be direct, in which case they are noted as $A \rightarrow [BC]$, or inverse, in which case they are written as $A \rightarrow \langle BC \rangle$, with $B, C \in N$. Whenever a sentence pair is analysed with the direct transduction rule, both strings are analysed with a derivation rule of the type $A \rightarrow BC$. However, when they are analysed with an inverse rule, one of the strings is parsed with the rule $A \rightarrow BC$, but the other one is parsed with the rule $A \rightarrow CB$. Figure 2.1 illustrates an example of these two types of derivation rules.

In this work we used only binary bracketing SITGs, although SITGs may have more than two non-terminals on the right side. The reason for this is that such a SITG admits an efficient bitext parsing algorithm, without adding any language-

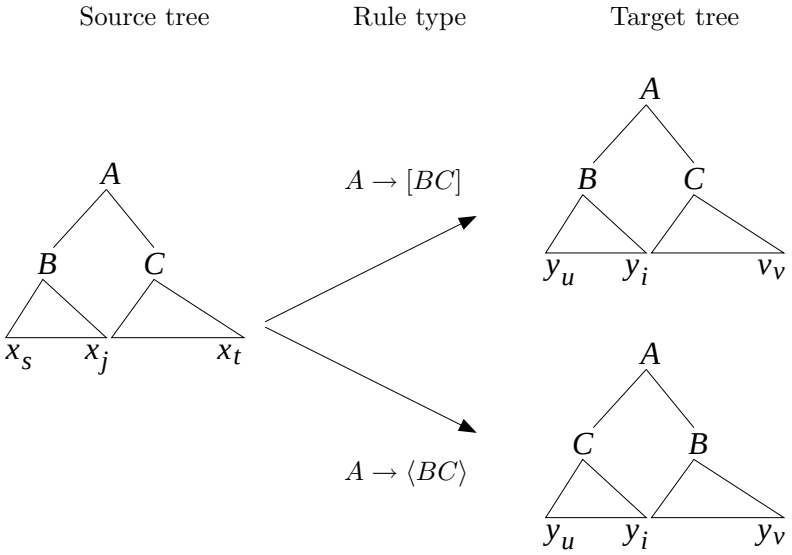

Figure 2.1: Direct and inverse derivation rules in a SITG. In the case of the direct rule ($A \rightarrow [BC]$), string $\{x_s...x_j\}$ is matched with string $\{y_u...y_i\}$ and string $\{x_{j+1}...x_t\}$ is matched with string $\{y_{i+1}...y_v\}$, whereas in the case of the inverse rule, the matching is $\{x_s...x_j\}$ with $\{y_{i+1}...y_v\}$ and $\{x_{j+1}...x_t\}$ with $\{y_u...y_i\}$.

specific bias. Nevertheless, binary bracketing SITGs are not able of representing all possible permutations that may occur during translation. More specifically, reorderings that contain as subsequence (3,1,4,2) or (2,4,1,3) have been shown to be impossible to achieve in a SITG parsing tree [Zens and Ney, 2003]. However, and despite this restriction, SITGs have proved to be useful in SMT tasks [Zens and Ney, 2003].

2.2 Parsing with SITGs

In [Wu, 1997], an algorithm similar to the CYK of context free grammars is proposed in order to parse a sentence pair with a SITG. Let be $\mathbf{x} = x_1...x_s...x_t...x_T$ the input sentence, $\mathbf{y} = y_1...y_u...y_v...y_V$ the output sentence, x_s^t the substring composed by words $x_{s+1}...x_t$ and y_u^v the substring $y_{u+1}...y_v$. Then, each node of the parse tree can be identified by a tuple $q = (s, t, u, v)$, meaning that node q will derive substrings x_s^t and y_u^v. $\delta_q(n)$ is defined as the maximum probability of any derivation from $n \in N$ that successfully parses x_s^t and y_u^v. Hence, $\delta_{0,T,0,V}(S)$, with S the axiom of the SITG, is the probability of the best parse of the given sentence pair.

Then, the following dynamic programming algorithm is established:

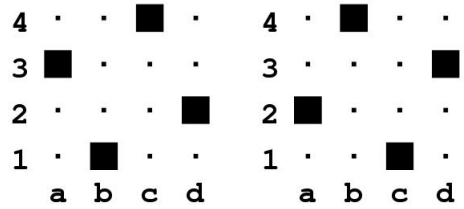

Figure 2.2: Illustration of the two reordering patterns which are not possible under the SITG framework.

1. **Initialization**

$$\delta_{t-1,t,v-1,v}(n) = b_n(x_t/y_v), \qquad 1 \le t \le T,\ 1 \le v \le V \tag{2.1}$$
$$\delta_{t-1,t,v,v}(n) = b_n(x_t/\epsilon), \qquad 1 \le t \le T,\ 0 \le v \le V \tag{2.2}$$
$$\delta_{t,t,v-1,v}(n) = b_n(\epsilon/y_v), \qquad 0 \le t \le T,\ 1 \le v \le V \tag{2.3}$$

2. **Recursion**

For all n, s, t, u, v such that $\begin{cases} n \in N \\ 0 \le s \le t \le T \\ 0 \le u \le v \le V \\ t - s + v - u > 2 \end{cases}$

$$\delta_{s,t,u,v}(n) = \max\left[\delta^{[\,]}_{s,t,u,v}(i), \delta^{\langle\,\rangle}_{s,t,u,v}(n)\right] \tag{2.4}$$

Where $b_n(x_t/y_v)$ is the probability of the lexical rule $n \to (x_t/y_v)$ and $\delta^{[]}_{s,t,u,v}(n)$ indicates that the direct rule was used in the first derivation of n, and $\delta^{\langle\rangle}_{s,t,u,v}$ indicates that the first rule in the derivation was an inverse rule. The condition $t-s+v-u > 2$ establishes that the substring in one, but not both, languages may be split into the empty string. This ensures that the recursion will terminate, but allows words of one language to have no match in the other language.

This algorithm has a time complexity of $O(T^3V^3|R|)$, being and $|R|$ the number of rules in the SITG. However, if the corpus has been previously parsed with a syntactical parser and is given in a bracketed form, [Sánchez and Benedí, 2006a] suggest the use of a version of the algorithm by [Wu, 1997] which is more efficient while performing the analysis, achieving a time complexity of $O(TV|R|)$ when x and y are fully bracketed. Then, the parsing algorithm by [Wu, 1997] is adequately modified, following a similar approach than [F. Pereira, 1992] did for CYK. Let be $B_{\mathbf{x}}$ the bracketing of $\mathbf{x}$ and $B_{\mathbf{y}}$ the bracketing of $\mathbf{y}$. Then, a derivation of $(\mathbf{x}, \mathbf{y})$ is compatible with $B_{\mathbf{x}}$ and $B_{\mathbf{y}}$ if, and only if, all the spans defined by such parsing are compatible with $B_{\mathbf{x}}$ and $B_{\mathbf{y}}$. Expressing the compatibility as a function, we have

$$c(s,t,u,v) = \begin{cases} 1 \text{ if} \begin{cases} (s,t) \text{ does not overlap any } b \in B_{\mathbf{x}} \\ (u,v) \text{ does not overlap any } b \in B_{\mathbf{y}} \end{cases} \\ 0 \text{ otherwise} \end{cases} \tag{2.5}$$

Once the compatibility function is defined, we can reformulate the recursion step in order to take into account only those parsings that are compatible with the given bracketing:

2 **Recursion**

$$\delta_{s,t,u,v}(n) = c(s,t,u,v) \max \left[\delta^{[\ ]}_{s,t,u,v}(i), \delta^{\langle\ \rangle}_{s,t,u,v}(n) \right] \tag{2.6}$$

At this point, it is important to emphasise that it is the bracketing of the bilingual corpus which enables training of SITGs with real-sized corpora. Only one iteration of the estimation algorithm by [Sánchez and Benedí, 2006a] for a SITG of only one non-terminal symbol takes about a day of computing time. Because of this, applying the algorithm by [Wu, 1997] with more than one non-terminal symbol becomes almost impracticable.

Moreover, the algorithm by [Sánchez and Benedí, 2006a] takes into account bracketing information contained in parsed corpora. This does not only imply a significant speedup, but also that the bilingual segments obtained will obey the constraints determined by the linguistic parsing. Hence, the segments obtained in this manner are bound to be linguistically motivated and are expected to be of better quality than those obtained with a purely heuristic algorithm.

2.3 SITGs for phrase extraction

Analysing two strings simultaneously and defining spans over each of the strings, SITGs consitute a natural fit for phrase probability estimation, while taking into account syntax motivated restrictions. The way this is done is easily explained with an example (Figure 2.3), in which it can be seen how strings in the input and output sentences relate either in a direct fashion (upper part of the figure) or in an inverse fashion (lower part of the figure). Since non-terminal symbols define spans over both input and output sentences, each non-terminal symbol will generate a new count in the resulting phrase table.

In our case, we will be first parsing the input or output string (or both) with a linguistic parser, in order to benefit of the algorithm by [Sánchez and Benedí, 2006a]. Then, we will reestimate the probabilities in a heuristically obtained SITG with such algorithm, and finally we will parse the bilingual corpus in order to generate the final phrase translation table. For Spanish and English, we used FreeLing [Asterias et al., 2006], which is an open-source suite of language analysers. For German, we used the Stanford Parser [Klein and Manning, 2003].

As in the case of traditional PB models, we used for our experimentation the direct and inverse translation probabilities (see Section 1.2.2). We also investigated the effect of adding the lexicalised weights and syntactic translation probabilities. These probabilities can be obtained by considering the probability with which each SITG derives a given string.

2.4 Syntactic probabilities with SITGs

In order to introduce a score that determines how probable is a given phrase according to the SITG trained, we introduced the following *syntax-based models*.

Let be $\mathbf{f} = x_s^t$ and $\mathbf{e} = y_u^v$. In the process of obtaining the best parse tree $\widehat{t}_{\mathbf{f},\mathbf{e}}$ a given pair of strings $(\mathbf{f}, \mathbf{e})$, a joint probability $\widehat{p}(\mathbf{f}, \mathbf{e})$ for several overlapping spans is obtained, which matches with the function $\delta_{s,t,u,v}(n)$ described in Section 2.2. However, this probability may be different depending on the non-terminal symbol the strings derive from, and, furthermore, depending on the bracketing of the particular sentence being parsed. Based on this information, it is possible to define a new translation model. Let Ω the multiset of spans (word segments) obtained from the training sample, and $\Omega_{\mathbf{f},\mathbf{e}} \subseteq \Omega$ the multiset of $(\mathbf{f}, \mathbf{e})$ spans. We define the expected value of $\widehat{p}(\mathbf{f}, \mathbf{e})$ according to the empirical distribution as:

$$E_\Omega(\widehat{p}(\mathbf{f}, \mathbf{e})) = \frac{\sum_{(\mathbf{a},\mathbf{b})\in\Omega_{\mathbf{f},\mathbf{e}}} \widehat{p}(\mathbf{a}, \mathbf{b})}{|\Omega|}. \tag{2.7}$$

If we marginalise for the input side of the word segments and for the output side of the segments, then we get:

$$E_\Omega(\widehat{p}(\mathbf{f})) = \sum_{\mathbf{e}} E_\Omega(\widehat{p}(\mathbf{f}, \mathbf{e}))$$

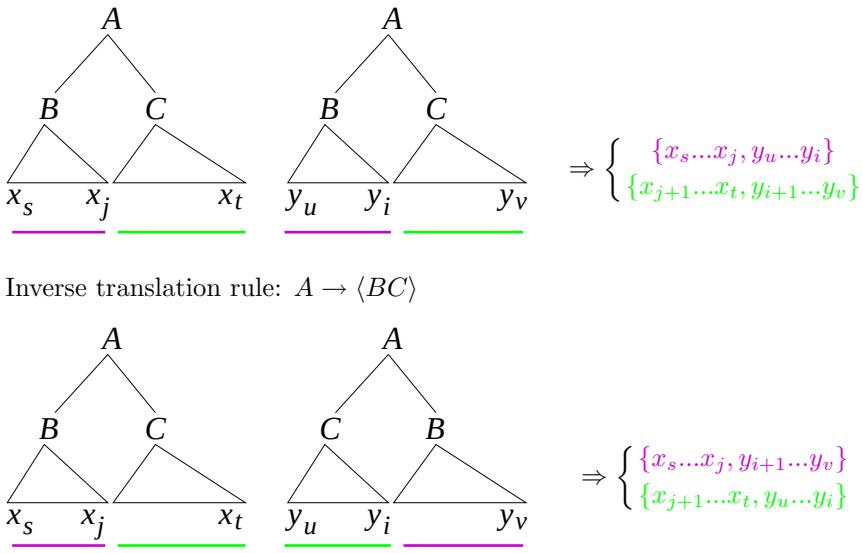

Figure 2.3: Example of phrase pairs that would be extracted.

and

$$E_\Omega(\widehat{p}(\mathbf{e})) = \sum_s E_\Omega(\widehat{p}(\mathbf{f}, \mathbf{e})).$$

In this way we obtain these two new *syntax-based* models:

$$p(\mathbf{f}|\mathbf{e}) = \frac{E_\Omega(\widehat{p}(\mathbf{f}, \mathbf{e}))}{E_\Omega(\widehat{p}(\mathbf{e}))}, \quad p(\mathbf{e}|\mathbf{f}) = \frac{E_\Omega(\widehat{p}(\mathbf{f}, \mathbf{e}))}{E_\Omega(\widehat{p}(\mathbf{f}))}. \tag{2.8}$$

2.5 Experimentation

2.5.1 System evaluation

The SMT system developed has been automatically evaluated by measuring the following rates:

WER *(Word Error Rate)*: The WER criterion computes the minimum number of editions (substitutions, insertions and deletions) needed to convert the translated sentence into the sentence considered ground truth. This measure is because of its nature a pessimistic one, when applied to Machine Translation.

BLEU *(Bilingual Evaluation Understudy) score*: This score measures the precision of unigrams, bigrams, trigrams, and 4-grams with respect to a set of reference

translations, with a penalty for too short sentences [Papineni et al., 2002]. BLEU is not an error rate, i.e. the higher the BLEU score, the better. BLEU can be single- or multi-reference. In this case we will be using single-reference BLEU because of corpus restrictions.

TER *(Translation Edit Rate)*: Translation Error Rate [Snover et al., 2006] is an error metric for machine translation that messures the number of edits required to change a system output into one of the references. TER is computed as the minumum number of edits required to modify the system hypothesis so that it matches the reference translation, normalized by the average number of reference words. In this case, possible edits include insertion, deletion, substitution of single words and shifts of word sequences. In the original paper, the authors claimed that single-reference TER correlates as well with human judgments of MT quality as the for-reference variant of BLEU. As in BLEU, TER can also be multi-reference, but we will be using single-reference TER.

2.5.2 Corpora

Europarl

We conducted experiments on the Europarl [Koehn, 2005] corpus, described in Section 1.3. In this case, we will focus on the German–English task, and evaluation was performed on the *Devtest* subset.

Parsing the Europarl corpus with a SITG can take very long even with 4 non-terminal symbols and having the corpus fully bracketed. For this reason, only pairs with input and output sentence length less than 25 were considered when reestimating the probabilities of the SITG. After the reestimation procedure, the resulting SITG was smoothed by adding all those rules in the heuristically obtained SITG, but with a probability 1000 times less than the rule with the least probability in the reestimated SITG. This was done so that extracting phrases from the complete corpus is still possible, since otherwise the amount of sentence pairs that the SITG would not be able to parse would grow considerably. Hence, the whole corpus was used for phrase extraction, since this procedure may be run in a parallelised fashion easily.

Albayzin 2008

Recently, we have taken part in the Albayzin Machine Translation competition, which has been organized in conjunction with the 2008 Jornadas en Tecnología del Habla. For this competition, the corpus chosen was the Albayzin corpus, a Spanish–Basque translation task. The statistics of this corpus can be seen in Table 2.1. As it can be seen on the Table, translating both from or into Basque is a difficult task, since the amount of Out of Vocabulary words quickly becomes very high.

Table 2.1: Characteristics of Albayzin corpus. K stands for thousands of elements, and M for millions.

		Spanish	Basque
Training	Sentences	58K	
	Running words	1151K	885M
	Vocabulary size	49.4K	87.8K
	Average length	19.8	15.2
Development	Sentences	1456	
	Running words	29K	23K
	Average length	20.1	15.5
	Out of Vocabulary	489	8376
Test	Sentences	1446	
	Running words	28K	22K
	Average length	19.3	14.9
	Out of Vocabulary	483	8096

2.5.3 Experimental results

First, we built an initial SITG by following the method described in [Sánchez and Benedí, 2006b]. Then, both source and target languages in the training corpus were bracketed by using FreeLing [Asterias et al., 2006], which is an open-source suite of language analysers. This being done, we then used the bracketed corpus to perform one estimation iteration on the initial SITG and obtain improved SITGs. Finally, the SITG obtained after the estimation iteration was used to parse the bracketed training corpus and extract segment pairs to setup a phrase-based translation model.

Initial SITGs with increasing number of non-terminal symbols were built and then estimated. The purpose of building SITGs with several non-terminal symbols was to analyse whether augmenting the number of non-terminals would improve word reorderings between both input and output languages. Adding non-terminal symbols may provide more complexity to the grammar built, and hence increases its expressive power. [Sánchez and Benedí, 2006b]

Results with Europarl

The results of this setup can be seen in Table 2.2. It can be seen that performing one reestimation iteration on the heuristical SITG proves to be beneficial, as shown by all three translation measures used, and throughout all the experiments carried out. Moreover, adding the syntax-based models or the lexicalised weights also proves to be beneficial, although the syntax-based models do not seem to improve the results obtained once the lexicalised weights have been added.

Comparatively, the best score obtained by the Moses toolkit [Koehn et al., 2007], which is a state of the art SMT system, in its default monotonic setup is 18.5/71.6/67.4,

Table 2.2: Translation results for a SITG with only one, two and four non-terminal symbols. Results are shown in BLEU/WER/TER. 0 iterations means the SITG was obtained by the heuristic technique, and *+syntactic* means that the syntax-based models were added to the phrase-table obtained with one reestimation iteration, *+lexical* that the lexical weights were added, and *+both* that both lexical weights and syntax-based models were added. The NT column lists the number of non-terminal symbols.

German–English

NT	It. 0	It. 1	+syntactic	+lexical	+both
1	19.9/73.1/66.2	20.2/71.1/66.2	21.5/69.7/64.4	22.5/68.6/62.9	22.8/68.6/62.8
2	20.1/72.3/66.4	21.2/70.1/65.1	22.3/69.4/64.0	23.3/67.9/62.2	**23.4/67.7/62.0**
4	20.5/71.3/66.5	21.2/69.9/64.9	22.2/69.0/63.5	23.4/67.8/62.0	23.1/68.0/62.3

English–German

NT	It. 0	It. 1	+syntactic	+lexical	+both
1	15.1/75.9/72.4	15.4/75.8/72.2	16.2/74.3/70.5	17.5/72.5/68.7	17.3/72.8/69.0
2	15.3/75.9/72.4	15.7/75.2/71.7	16.4/74.2/70.4	17.3/72.9/69.1	17.4/73.1/69.2
4	15.4/75.8/72.1	16.0/74.9/71.2	16.5/74.2/70.4	17.5/72.8/69.0	**17.6/72.8/68.9**

for English→German, and 25.0/66.7/60.6 for German→English. Although Moses obtains a slightly better score, it must be taken into consideration that this toolkit achieves this by using about 1.5 times more phrases than the system built in this Section. This fact has important implications: being the model smaller, less computational resources are used in decoding time, but also the final translation is produced faster.

Results with Albayzin

Being corpus is smaller than the Europarl corpus allowed for a second reestimation iteration of the SITG. Hence, the results presented here involve two reestimation iterations. Since it had already been shown in the case of the Europarl corpus that adding syntax-based probabilities or lexical weights proved to be beneficial, in this case no experiments involving only direct and inverse translation models were performed.

As Table 2.3 shows, the translation quality tends to get better when increasing number of non-terminal symbols are used, as measured by BLEU. Moreover, the combination in which all translation models are used seems to yield improvements over the other alternatives, as measured by BLEU, WER and TER. However, it must be noted that these differences are not statistically significant. The results shown in this table were obtained restricting the decoder to perform a monotonic translation procedure, since at this stage we have not yet implemented a SITG-based reordering model. In this case, the language model used was a 5-gram, applying interpolation with Knesser-Ney discount.

Table 2.3: Translation results for Spanish-Basque translation when using a SITG with only one, three and five non-terminal symbols

non terms	combination	BLEU	WER	TER
1	+syntactic	8.8	82.0	78.5
	+lexical	8.8	81.8	78.2
	+both	9.0	81.7	78.1
3	+syntactic	8.9	81.9	78.6
	+lexical	8.9	81.8	78.3
	+both	9.1	**81.4**	**77.9**
5	+syntactic	9.1	82.2	78.7
	+lexical	9.2	81.5	78.9
	+both	**9.3**	81.6	78.1

For comparison purposes, the best scores obtained by the Moses toolkit in its default monotonic setup are 9.4 BLEU, 81.7 WER and 78.3 TER, which are not significantly better than the scores obtained by our system trained with 5 non-terminal symbols with all translation models.

2.6 Conclusions

In this work, an alternative method for phrase extraction is presentend, which is competitive in terms of quality and produces smaller phrase-based models when compared to the traditional phrase-based extraction algorithms used. This method obtains phrase segments from paired sentences by parsing both of them in a completely unlexicalixed manner.

In the future, we plan to compute more complex SITGs and introduce further models to improve our translation table, such as other models obtained by combining the various probabilities that SITG estimation entails. In this line, we also plan to investigate which effect has the combination of our phrase table with the phrase table produced by Moses.

Lastly, we also plan on investigating how to make use of the direct/inverse translation rule probabilities in order to obtain an adequate reordering model.

Chapter 3

Interactive Machine Translation

Information technology advances in modern society have led to the need of more efficient methods of translation. It is important to remark that current MT systems are not able to produce ready-to-use texts [Kay, 1997, Hutchins, 1999, Arnold, 2003]. Indeed, MT systems are usually limited to specific semantic domains and the translations provided require human post-editing in order to achieve a correct high-quality translation.

A way of taking advantage of MT systems is to combine them with the knowledge of a human translator, constituting the so-called Computer-Assisted Translation (CAT) paradigm. CAT offers different approaches in order to benefit from the synergy between humans and MT systems.

An important contribution to interactive CAT technology was carried out around the TransType (TT) project [Langlais et al., 2002, Foster et al., 2002, Foster, 2002, Och et al., 2003]. This project entailed an interesting focus shift in which interaction directly aimed at the production of the target text, rather than at the disambiguation of the source text, as in former interactive systems. The idea proposed was to embed data driven MT techniques within the interactive translation environment.

Following these TT ideas, [Barrachina et al., 2008] propose the usage of fully-fledged statistical MT (SMT) systems to produce full target sentence hypotheses, or portions thereof, which can be partially or completely accepted and amended by a human translator. Each partial correct text segment is then used by the SMT system as additional information to achieve further, hopefully improved suggestions. In this paper, we also focus on the interactive and predictive, statistical MT (IMT) approach to CAT. The IMT paradigm fits well within the *Interactive Pattern Recognition* framework introduced in [Vidal et al., 2007].

Figure 3.1 illustrates a typical IMT session. Initially, the user is given an input sentence $\mathbf{x}$ to be translated. The reference $\mathbf{y}$ provided is the translation that the user would like to achieve at the end of the IMT session. At iteration 0, the user does

SOURCE (x):		Para encender la impresora:
REFERENCE (y):		To power on the printer:
ITER-0	($\mathbf{p}$)	( )
	($\hat{\mathbf{s}}_h$)	*To switch on:*
ITER-1	($\mathbf{p}$)	To
	($\mathbf{s}_l$)	*switch on:*
	(k)	power
	($\hat{\mathbf{s}}_h$)	*on the printer:*
ITER-2	($\mathbf{p}$)	To power on the printer:
	($\mathbf{s}_l$)	( )
	(k)	(#)
	($\hat{\mathbf{s}}_h$)	( )
FINAL	($\mathbf{p} \equiv \mathbf{y}$)	To power on the printer:

Figure 3.1: IMT session to translate a Spanish sentence into English. Non-validated hypotheses are displayed in italics, whereas accepted prefixes are printed in normal font.

not supply any correct text prefix to the system, for this reason $\mathbf{p}$ is shown as empty. Therefore, the IMT system has to provide an initial complete translation $\mathbf{s}_h$, as it were a conventional SMT system. At the next iteration, the user validates a prefix $\mathbf{p}$ as correct by positioning the cursor in a certain position of $\mathbf{s}_h$. In this case, after the words "*To print a*". Implicitly, he is also marking the rest of the sentence, the suffix $\mathbf{s}_l$, as potentially incorrect. Next, he introduces a new word k, which is assumed to be different from the first word s_{l_1} in the suffix $\mathbf{s}_l$ which was not validated, $k \neq s_{l_1}$. This being done, the system suggests a new suffix hypothesis $\hat{\mathbf{s}}_h$, subject to $\hat{s}_{h_1} = k$. Again, the user validates a new prefix, introduces a new word and so forth. The process continues until the whole sentence is correct that is validated introducing the special word "#".

As the reader could devise from the IMT session described above, IMT aims at reducing the effort and increasing the productivity of translators, while preserving high-quality translation. For instance, in Figure 3.1, only three interactions were necessary in order to achieve the reference translation.

3.1 Statistical Interactive-Predictive Machine Translation

Relying on the basic formulation of SMT, to establish the fundamental equation for IMT we need to modify Equation 1.1 according to the IMT scenario in order to take into account part of the target sentence that is already translated, that is $\mathbf{p}$ and k

$$\hat{\mathbf{s}}_h = \underset{\mathbf{s}_h}{\operatorname{argmax}}\, Pr(\mathbf{s}_h|\mathbf{x}, \mathbf{p}, k) \tag{3.1}$$

where the maximisation problem is defined over the suffix $\mathbf{s}_h$. This allows us to rewrite Eq. 3.1, by decomposing the right side appropriately and eliminating constant terms, achieving the equivalent criterion

$$\hat{\mathbf{s}}_h = \underset{\mathbf{s}_h}{\operatorname{argmax}} \, Pr(\mathbf{p}, k, \mathbf{s}_h | \mathbf{x}). \tag{3.2}$$

An example of the intuition behind these variables can be seen in Figure 3.1.

Note that, since $(\mathbf{p}\, k\, \mathbf{s}_h) = \mathbf{y}$, Eq. 3.2 is very similar to Eq. 1.1. The main difference is that the argmax search is now performed over the set of suffixes $\mathbf{s}_h$ that complete $(\mathbf{p}\, k)$ instead of complete sentences ($\mathbf{y}$ in Eq. 1.1). This implies that we can use the same models if the search procedures are adequately modified [Barrachina et al., 2008].

3.2 Phrase-based IMT

The phrase-based approach presented above can be easily adapted for its use in an IMT scenario. The most important modification is to rely on a word graph that represents possible translations of the given source sentence. The use of word graphs in IMT has been studied in [Barrachina et al., 2008] in combination with two different translation techniques, namely, the Alignment Templates technique [Och et al., 1999, Och and Ney, 2004], and the Stochastic Finite State Transducers technique [Casacuberta and Vidal, 2007].

3.3 IMT using word graphs

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 scores given by an SMT model (see [Ueffing et al., 2002] for more details).

In [Och et al., 2003], the use of a wordgraph is proposed as interface between an alignment-template SMT model and the IMT engine. Analogously, in this work we will be using a wordgraph built during the search procedure performed on a PB SMT model. Since a such a model would generate a *phrase-graph*, instead of a word-graph, it is necessary to convert the former into the latter. However, such procedure is quite simple, and is achieved by adding artificial nodes and edges between each one of the words that constitute the phrases and assigning the score of the phrase to the final edge. An example of this procedure can be seen in 3.3. Note that the scores on the edges are not probabilities, since the maximisation in Formula 1.4 is performed without normalisation.

During the process of IMT for a given source sentence, the system makes use of the word graph generated for that sentence in order to complete the prefixes accepted by the human translator. Specifically, the system finds the best path in the word graph associated with a given prefix so that it is able to complete the target sentence, being capable of providing several completion suggestions for each prefix.

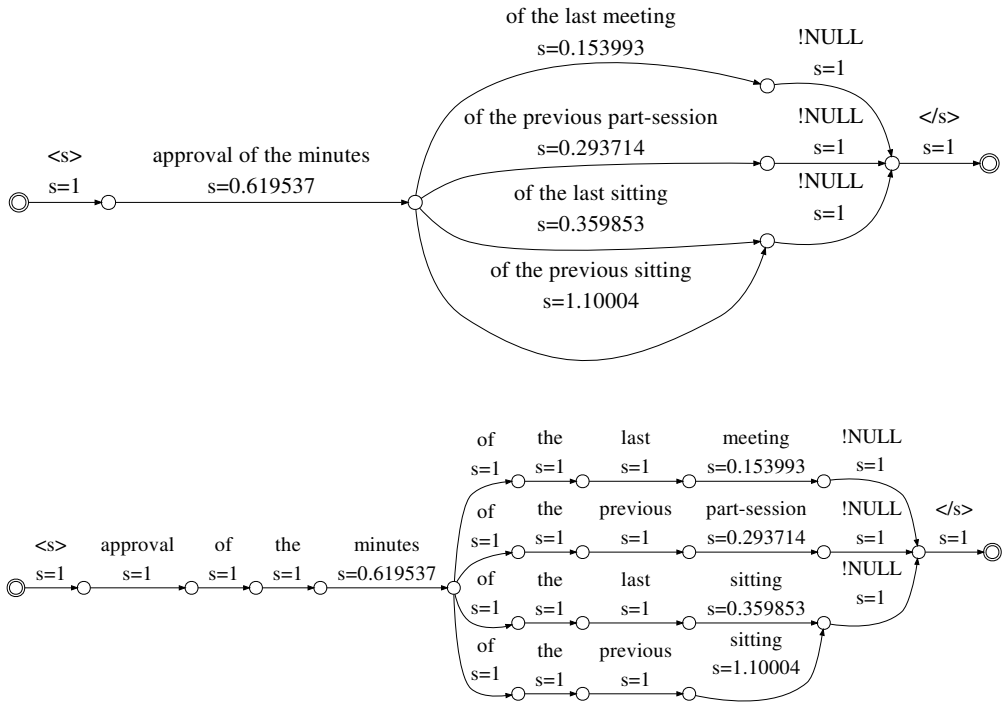

Figure 3.2: Example of word-graph (up) obtained from a phrase-graph (down).

A common problem in IMT arises when the user sets a prefix which cannot be found in the word graph, since in such a situation the system is unable to find a path through the word graph and provide an appropriate suffix. The common procedure to face this problem is to perform a tolerant search in the word graph. This tolerant search uses the well known concept of Levenshtein distance in order to obtain the most similar string for the given prefix (see [Och et al., 2003] for more details).

Chapter 4

Enriching user–machine interaction

Although the IMT paradigm has proved to offer interesting benefits to potential users, one aspect that has not been reconsidered as of yet is the user–machine interface. Hence, in traditional IMT the system only received feedback whenever the user typed in a new word. In this work, we show how to enrich user–machine interaction by introducing Mouse Actions (MA) as an additional information source for the system. By doing so, we will consider two types of MAs, i.e. *non-explicit* (or *positioning*) MAs and *interaction-explicit* MAs.

4.1 Non-explicit positioning MAs

Before typing in a new word in order to correct a hypothesis, the user needs to position the cursor in the place where he wants to type such a word. In this work, we will assume that this is done by performing a MA, although the same idea presented can also be applied when this is done by some other means. It is important to point out that, by doing so, the user is already providing some very useful information to the system: he is validating a prefix up to the position where he positioned the cursor, and, in addition, he is signalling that whatever word is located after the cursor is to be considered incorrect. Hence, the system can already capture this fact and provide a new translation hypothesis, in which the prefix remains unchanged and the suffix is replaced by a new one in which the first word is different to the first word of the previous suffix. We are aware that this does not mean that the new suffix will be correct, but given that we know that the first word in the previous suffix was incorrect, the worst thing which can happen is that the the first word of the new suffix is incorrect as well. However, if the new suffix happens to be correct, the user will happily find that he does not need to correct that word any more.

An example of such behaviour can be seen in Figure 4.1. In this example, the SMT system first provides a translation which the user does not like. Hence, he positions

SOURCE (x):		Para encender la impresora:
REFERENCE (y):		To power on the printer:
ITER-0	(**p**)	( )
	($\hat{\mathbf{s}}_h$)	*To switch on:*
ITER-1	(**p**)	To
	($\mathbf{s}_l$)	|*switch on:*
	($\hat{\mathbf{s}}_h$)	*power on the printer:*
ITER-2	(**p**)	To power on the printer:
	($\mathbf{s}_l$)	( )
	(k)	(#)
	($\hat{\mathbf{s}}_h$)	( )
FINAL	(**p** ≡ **y**)	To power on the printer:

Figure 4.1: Example of non-explicit positioning MA which solves an error of a missing word. In this case, the system produces t he correct suffix $\mathbf{s}_h$ immediately after the user validates a prefix **p**, implicitly indicating that we wants the suffix to be changed, without need of any further action. In **ITER-1**, character | indicates the position where a MA was performed, $\mathbf{s}_l$ is the suffix which was rejected by that MA, and $\hat{\mathbf{s}}_h$ is the new suffix that the system suggests after observing that $\mathbf{s}_l$ is to be considered incorrect. Character # is a special character introduced by the user to indicate that the hypothesis is to be accepted.

the cursor before word "*postscript*", with the purpose of typing in "*lists*". By doing so, he is validating the prefix "*To print a*", and signalling that he wants "*postscript*" to be replaced. Before typing in anything, the system realises that he is going to change the word located after the cursor, and replaces the suffix by another one, which is the one the user had in mind in the first place. Finally, the user only has to accept the final translation.

We are naming this kind of MA *non-explicit* because it does not require any additional action from the user: he has already performed a MA in order to position the cursor at the place he wants, and we are taking advantage of this fact to suggest a new suffix hypothesis.

Since the user needs to position the cursor before typing in a new word, it is important to point out that any improvement achieved by introducing non-explicit MAs does not require any further effort from the user, and hence is considered to have no cost.

Hence, we are now considering two different situations: the first one, the traditional IMT framework, in which the system needs to find a suffix according to Eq. 3.2, and a new one, in which the system needs to find a suffix in which the first word does not need to be a given k, but needs to be *different* to a given s_{l_1}. This constraint can be expressed by the following equation:

$$\hat{\mathbf{s}}_h = \underset{\mathbf{s}_h : s_{h_1} \neq s_{l_1}}{\operatorname{argmax}} \; Pr(\mathbf{p}, \mathbf{s}_h | \mathbf{x}, \mathbf{s}_l) \tag{4.1}$$

SOURCE (x):		Seleccione el tipo de instalación.
REFERENCE (y):		Select the type of installation.
ITER-0	(**p**) ($\hat{\mathbf{s}}_h$)	( ) *Select the installation wizard.*
ITER-1	(**p**) ($\mathbf{s}_l$) ($\hat{\mathbf{s}}_h$)	Select the |*installation wizard.* *install script.*
ITER-2	(**p**) (k) ($\hat{\mathbf{s}}_h$)	Select the type *installation wizard.*
ITER-3	(**p**) ($\mathbf{s}_l$) ($\hat{\mathbf{s}}_h$)	Select the type |*installation wizard.* *of installation.*
ITER-4	(**p**) ($\mathbf{s}_l$) (k) ($\hat{\mathbf{s}}_h$)	Select the type of installation. ( ) (#) ( )
FINAL	(**p** ≡ **y**)	Select the type of installation.

Figure 4.2: Example of explicit interactive MA which corrects an erroneous suffix. In this case, a non-explicit MA is performed in **ITER-1** with no success. Hence, the user introduces word "*type*" in **ITER-2**, which leaves the cursor pos ition located immediately after word "*type*". In this situation the user would not need to perform a MA to re-position the cursor and continue typing in order to further correct the remaining errors. However, since he has learnt the potential benefit of MAs, he performs an interaction-explicit MA in order to ask for a new suffix hypothesis, which happens to correct the error.

where $\mathbf{s}_l$ is the suffix generated in the previous iteration, already discarded by the user, and s_{l_1} is the first word in $\mathbf{s}_l$. k is omitted in this formula because the user did not type any word at all.

4.2 Interaction-explicit MAs

If the system is efficient and provides suggestions which are good enough, one could easily picture a situation in which the expert would ask the system to replace a given suffix, without typing in any word. We will be modelling this as another kind of MA, *interaction-explicit* MA, since the user needs to indicate *explicitly* that he wants a given suffix to be replaced, in contrast to the non-explicit positioning MA. However, if the underlying MT engine providing the suffixes is powerful enough, the user would quickly realise that performing a MA is less costly that introducing a whole new word, and would take advantage of this fact by systematically clicking before introducing any new word. In this case, as well, we assume that the user clicks before an incorrect

word, hence demanding a new suffix whose first word is different, but by doing so he is adopting a more participative and interactive attitude, which was not demanded in the case of non-explicit positioning MAs. An example of such an explicit MA correcting an error can be seen in Figure 4.2

In this case, however, there is a cost associated to this kind of MAs, since the user does need to perform additional actions, which may or may not be beneficial. It is very possible that, even after asking for several new hypothesis, the user will even though need to introduce the word he had in mind, hence wasting the additional MAs he had performed.

If we allow the user to perform n MAs before introducing a word, this problem can be formalised in an analogous way as in the case of non-explicit MAs as follows:

$$\hat{\mathbf{s}}_h = \underset{\mathbf{s}_h : s_{h_1} \neq s^i_{l\,1} \forall i \in \{1..n\}}{\operatorname{argmax}} Pr(\mathbf{p}, \mathbf{s}_h | \mathbf{x}, \mathbf{s}^1_l, \mathbf{s}^2_l, \ldots, \mathbf{s}^n_l) \tag{4.2}$$

where $s^i_{l\,1}$ is the first word of the i-th suffix discarded and $\mathbf{s}^1_l, \mathbf{s}^2_l, \ldots, \mathbf{s}^n_l$ is the set of all n suffixes discarded.

Note that this kind of MA could also be implemented with some other kind of interface, e.g. by typing some special key such as `F1` or `Tab`. However, the experimental results would not differ, and in our user interface we found it more intuitive to implement it as a MA.

4.3 Experimental results in IMT

4.3.1 System evaluation

Automatic evaluation of results is a difficult problem in MT. In fact, it has evolved to a research field with own identity. This is due to the fact that, given an input sentence, a large amount of correct *and* different output sentences may exist. Hence, there is no sentence which can be considered ground truth, as is the case in speech or text recognition. By extension, this problem is also applicable to IMT.

In this paper, we will be reporting our results as measured by *Word Stroke Ratio* (WSR) [Barrachina et al., 2008], which is computed as the quotient between the number of word-strokes a user would need to perform in order to achieve the translation he has in mind and the total number of words in the sentence. In this context, a word-stroke is interpreted as a single action, in which the user types a complete word, and is assumed to have constant cost. Moreover, each word-stroke also takes into account the cost incurred by the user when reading the new suffix provided by the system.

In the present work, we decided to use WSR instead of *Key Stroke Ratio* (KSR), which is used in other works on IMT such as [Och et al., 2003]. The reason for this is that KSR is clearly an optimistic measure, since in such a scenario the user is often overwhelmed by receiving a great amount of translation options, as much as one per key stroke, and it is not taken into account the time the user would need to read all those hypotheses.

In addition, and because we are also introducing MAs as a new action, we will also present results in terms of *Mouse Action Ratio* (MAR), which is the quotient between the amount of explicit MAs performed and the number of words of the final translation. Hence, the purpose is to elicit the number of times the user needed to request a new translation (i.e. performed a MA), on a per word basis.

Lastly, we will also present results in terms of uMAR (useful MAR), which indicates the amount of MAs which were *useful*, i.e. the MAs that actually produced a change in the first word of the suffix and such word was accepted. Formally, uMAR is defined as follows:

$$uMAR = \frac{MAC - n \cdot WSC}{MAC} \tag{4.3}$$

where MAC stands for "Mouse Action Count", WSC for "Word Stroke Count" and n is the maximum amount of MAs allowed before the user types in a word. Note that $MAC - n \cdot WSC$ is the amount of MAs that were useful since WSC is the amount of word-strokes the user performed even though he had already performed n MAs.

Since we will only use single-reference WSR and MAR, the results presented here are clearly pessimistic. In fact, it is relatively common to have the underlying SMT system provide a perfectly correct translation, which is "corrected" by the IMT procedure into another equivalent translation, increasing WSR and MAR significantly by doing so.

Table 4.1: WSR improvement when considering non-explicit MAs. "rel." indicates the relative improvement. All results are given in %.

pair	baseline	non-explicit	rel.
Es–En	63.0±0.9	59.2±0.9	6.0±1.4
En–Es	63.8±0.9	60.5±1.0	5.2±1.6
De–En	71.6±0.8	69.0±0.9	3.6±1.3
En–De	75.9±0.8	73.5±0.9	3.2±1.2
Fr–En	62.9±0.9	59.2±1.0	5.9±1.6
En–Fr	63.4±0.9	60.0±0.9	5.4±1.4

4.3.2 Corpora

Our experiments were carried out on the Europarl [Koehn, 2005] corpus, described in Section 1.3. The results shown here are over the *Devtest* subcorpus.

4.3.3 Experimental results

As a first step, we built a SMT system for each of the language pairs cited in the previous subsection. This was done by means of the Moses toolkit [Koehn et al., 2007], which is a complete system for building Phrase-Based SMT models. This toolkit involves the estimation from the training set of four different translation models, which are in turn combined in a log-linear fashion by adjusting a weight for each of them by means of the MERT [Och, 2003] procedure, optimising the BLEU [Papineni et al., 2002] score obtained on the development partition.

This being done, word graphs were generated for the IMT system. For this purpose, we used a multi-stack phrase-based decoder which will be distributed in the near future together with the Thot toolkit [Ortiz-Martínez et al., 2005]. We discarded the use of the Moses decoder because preliminary experiments performed with it revealed that the decoder by [Ortiz-Martínez et al., 2005] performs clearly better when used to generate word graphs for use in IMT. In addition, we performed an experimental comparison in regular SMT with the Europarl corpus, and found that the performance difference was negligible. The decoder was set to only consider monotonic translation, since in real IMT scenarios considering non-monotonic translation leads to excessive waiting time for the user.

Finally, the word graphs obtained were used within the IMT procedure to produce the reference translation contained in the test set, measuring WSR and MAR. The results of such a setup can be seen in Table 4.1. As a baseline system, we report the traditional IMT framework, in which no MA is taken into account. Then, we introduced non-explicit MAs, obtaining an average improvement in WSR of about 3.2% (4.9% relative). The table also shows the confidence intervals at a confidence level of 95%. These intervals were computed following the bootstrap technique described in [Koehn, 2004]. Since the confidence intervals do not overlap, it can be stated that

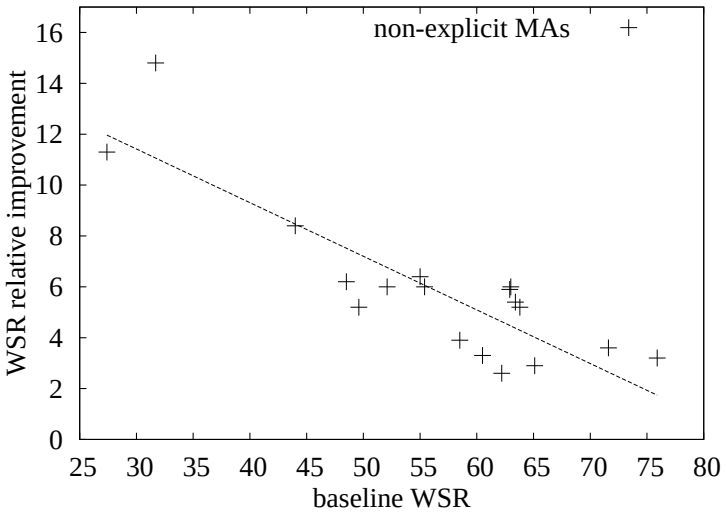

Figure 4.3: Plot evidencing the inverse relation between baseline WSR and WSR relative improvement when considering non-explicit and explicit MAs. This graph has been obtained by including preliminary results with other corporae.

the improvements obtained are statistically significant.

Once the non-explicit MAs were considered and introduced into the system, we analysed the effect of performing up to a maximum of 5 explicit MAs. Here, we modelled the user in such a way that, in case a given word is considered incorrect, he will always ask for another translation hypothesis until he has asked for as many different suffixes as MAs considered. The results of this setup can be seen in Figure 4.4. This yielded a further average improvement in WSR of about 16% (25% relative improvement) when considering a maximum of 5 explicit MAs. However, relative improvement in WSR and uMAR increase drop significantly when increasing the maximum allowed amount of explicit MAs from 1 to 5. For this reason, it is difficult to imagine that a user would perform more than two or three MAs before actually typing in a new word. Nevertheless, just by asking twice for a new suffix before typing in the word he has in mind, the user might be saving about 15% of word-strokes.

Although the results in Figure 4.4 are only for the translation direction "foreign"→English, the experiments in the opposite direction (i.e. English→"foreign") were also performed. However, the results were very similar to the ones displayed here. Because of this, and for clarity purposes, we decided to omit them and only display the direction "foreign"→English.

Moreover, it must be noted that, according to these results, it seems that the lower the baseline WSR, the higher the relative improvement when introducing both non-

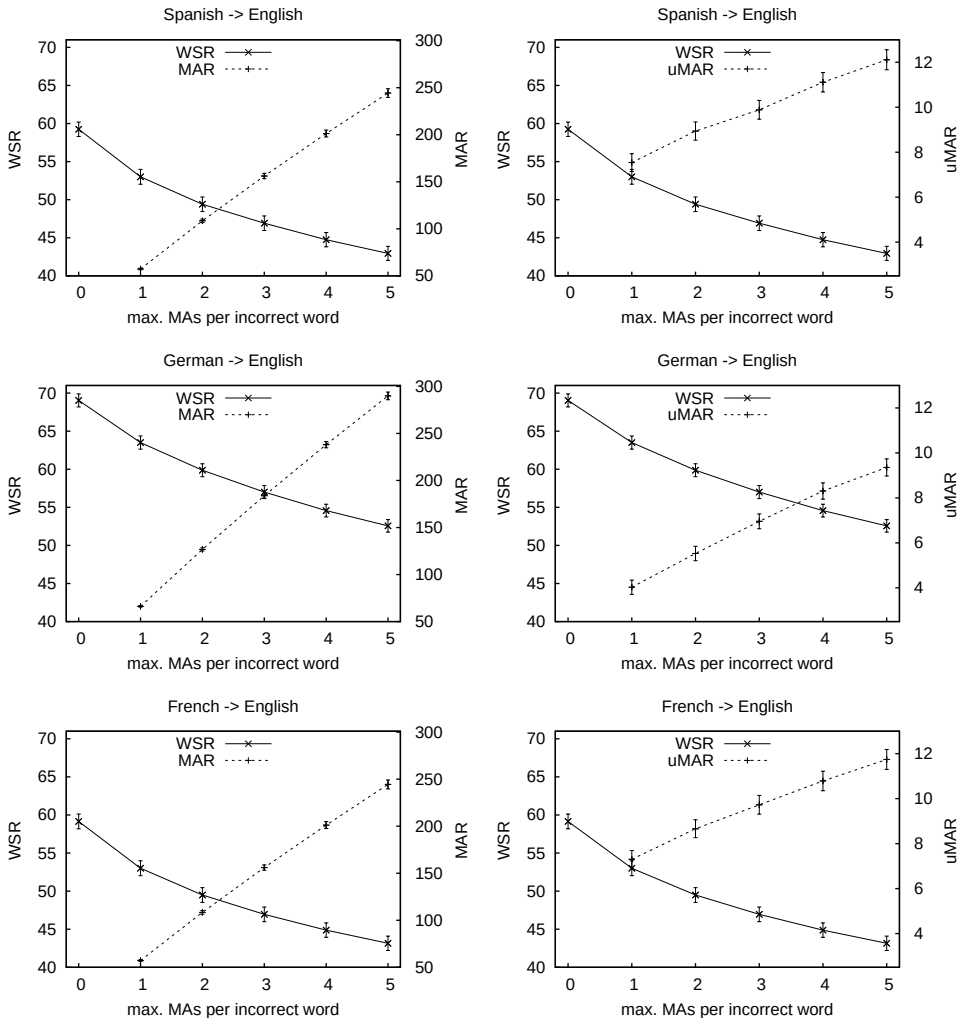

Figure 4.4: WSR improvement when considering one to five maximum MAs. All figures are given in %. The left column lists WSR improvement versus MAR degradation, and the right column lists WSR improvement versus uMAR. Confidence intervals at 95% confidence level following [Koehn, 2004].

explicit and explicit MAs. This is due to the fact that a higher baseline points towards a better translation model, which will, in turn, be able to provide a more useful suffix hypotheses when asking it to return a new $\hat{\mathbf{s}}_h$ such that $s_{h_1} \neq s_{l_1}$. If the translation model is not complex enough, it will most probably return an empty suffix, since the only suffix hypothesis which it is able provide is the one the user already discarded.

This (inverse) relation is illustrated in Figure 4.3.

4.4 Conclusions

New input sources for IMT have been considered. More specifically, it has been shown that considering Mouse Actions can lead to significant and consistent improvements in terms of word-stroke reduction, both when considering only non-explicit MAs and when considering MAs as a way of offering the user several suffix hypotheses. In addition, we have applied these ideas on a state-of-the-art SMT baseline, such as phrase-based models. To achieve this, we have first obtained a word graph for each sentence which is to be translated. Experiments were carried out on a reference corpus in SMT.

Note that there are other systems [Esteban et al., 2004] that, for a given prefix, provide n-best lists of suffixes. However, the functionality of our system is slightly (but fundamentally) different, since the suggestions are demanded to be different in their first word, which implies that the n-best list is scanned deeper, going directly to those hypotheses that may be of interest to the user. In addition, this can be done "on demand", which implies that the system's response is faster and that the user is not confronted with a large list of hypotheses, which often results overwhelming.

4.5 Future work

As future work, we are planning on performing a human evaluation that assesses the appropriateness of the improvements described.

Other future work in this field involves the reestimation of the weights in the log-linear model of the underlying SMT system with the purpose of performing an adaptation to the current human translator or topic. This will be done following the ideas of Bayesian Adaptation.

Chapter 5

Other contributions

This section lists other contributions to Statistical Machine Translation which were achieved during the research period that culminated in this Masters Thesis.

5.1 Source sentence reordering

In the last years, SMT systems have evolved to become the present state of the art, two of the most representative techniques being the phrase based models [Koehn et al., 2003, Och and Ney, 2004] and the Weighted Finite State Transducers for Machine Translation [Casacuberta and Vidal, 2004, Kumar and Byrne, 2003]. Both of these frameworks typically rely on word-aligned corpora, which often lead them to incur in word ordering related errors. Although there have been different efforts aiming towards enabling them to deal with non-monotonicity, the algorithms developed often only account for very limited reorderings, being unable to tackle with the more complex reorderings that e.g. some Asian languages introduce with respect to European languages. Because of this, not only will monotone systems present incorrectly ordered translations, but, in addition, the parameters of such models will be incorrectly estimated, whenever a certain input phrase is erroneously assumed to be the translation of a certain output phrase in training time.

Although no efficient solution has still been found, this problem is well known already since the origin of what is known as statistical machine translation: [Berger et al., 1996] already introduced in their alignment models what they called distortion models, in an effort towards including in their SMT system a solution for the reordering problem. However, these distortion models are usually implemented within the decoding algorithms and imply serious computational problems, leading ultimately to restrictions being applied to the set of possible permutations of the output sentence. Hence, the search performed turns sub-optimal, and an important loss in the representational power of the distortion models takes place.

On the other hand, dealing with arbitrary word reordering and choosing the one which best scores given a translation model has been shown not to be a vi-

able solution, since when allowing all possible word permutations the search is NP-hard [Knight, 1999].

The present work is based on the work of Zens, Matusov and Kanthak [Zens et al., 2004, Matusov et al., 2005, Kanthak et al., 2005], who introduced the idea of monotonizing a corpus. The key idea behind this concept is to use the IBM alignment models to efficiently reorder the input sentence s and produce a new bilingual, monotone pair, composed by the reordered input sentence s' and the output sentence t. Hence, once this new bilingual pair has been produced, the translation model to be applied will not have to tackle with the problems derived from different word reorderings, since this problem will not be present any more. Still, there is one more problem to be solved: in search time, only the input sentence is available, and hence the pair cannot be monotonized. To solve this, a very simple reordering model will be introduced, together with a *reordered* language model and n-best hypothesis generation. In this work, a phrase based model is trained using these monotone pairs.

5.1.1 Brief overview of existing approaches

Three main possibilities exist when trying to solve the reordering problem: input sentence reordering, output sentence reordering, or reordering both. The latter is, to the best of our knowledge, as yet unexplored.

[Vilar et al., 1996], tried to partially solve the problem by monotonizing the most probable non-monotone alignment patterns and adding a mark in order to be able to remember the original word order. This being done, a new output language has been defined and a new language and translation model can be trained, making the translation process now monotone.

More recently, [Kumar and Byrne, 2005] learned weighted finite state transducers accounting for local reorderings of two or three positions. These models were applied to phrase reordering, but the training of the models did not yield statistically significant results with respect to the introduction of the models with fixed probabilities.

When dealing with input sentence reordering [Zens et al., 2004, Matusov et al., 2005, Kanthak et al., 2005], the main idea is to reorder the input sentence in such a way that the translation model will not need to account for possible word reorderings. To achieve this, alignment models are used, in order to establish which word order should be the appropriate for the translation to be monotone, and then the input sentence is reordered in such a manner that the alignment is monotone.

However, this approach has an obvious problem, since the output sentence is not available in search time and the sentence pair cannot be made monotone.

The naïve solution, test on all possible permutations of the input sentence, has already been discussed earlier, being NP-hard [Knight, 1999], as $J!$ possible permutations can be obtained from a sentence of length J. Hence, the search space must be restricted, and such restrictions are bound to yield sub-optimal results. In their work, Kanthak et al. present four types of constraints: IBM, inverse IBM, local and ITG constraints.

- Let:
  - s a source sentence, and s_j its j-th word
  - t a target sentence, and t_i its i-th word
- Let C be a cost matrix
  $c_{ij} = \text{cost}(\text{align}(s_j, t_i))$
- Let $\{s^r\} = \{\text{all possible permutations of } s\}$.
  1. compute alignment $A_D(j) = \underset{i}{\text{argmin}}\, c_{ij}$
  2. $s' = \{s^r | \forall j : A_D(j) \leq A_D(j+1)\}$
  3. recompute (reorder) C, obtaining C'.
  4. set $A'_I(i) = \underset{j}{\text{argmin}}\, {c'}_{ij}$.
  5. Optional: Compute minimum-cost *monotonic* path through cost matrix C'.

Figure 5.1: Algorithm for obtaining a monotonic alignment by reordering the source sentence.

Although the restrictions presented in their work (IBM, inverse IBM, local and ITG constraints) did yield interesting results, the search space still remained huge, and the computational price paid for a relatively small benefit was far too high.

5.1.2 The reordering model and N-Best reorderings

An important motivation behind the approach in this work is that the reordering constraints presented by Kanthak et al. [Kanthak et al., 2005] do not take into account extremely significant information that can be extracted from monotonized corpora: while reordering the input sentence in such a fashion that the alignment turns monotone, we are performing the reordering step needed further on when this action is needed to be taken on the input test set. Hence, what we would ideally want to do is learn a model using this information that will be capable of reordering a given, unseen, input sentence in the same way that the monotonization procedure would have done, in the hope that the benefits introduced will be greater than the error that an additional model will add into the translation procedure.

Once the alignments made monotonic according to the algorithm shown in Figure 5.1 [Kanthak et al., 2005], a new source "*language*" has been established, meaning that a reordered language model can be trained with the reordered input sentences s'. Such a language will have the words of the original source language, but the distinctive ordering of the target language. An example of this procedure is shown in Figure 5.2. Hence, a reordering model can be learnt from the monotonized corpus, which will most likely not depend on the output sentence, whenever the word-by-word translation is accurate enough.

Hence, the reordering problem can be defined as:

		Spanish	Basque
Training	Sentences	38940	
	Different pairs	20318	
	Words	368314	290868
	Vocabulary	722	884
	Average length	9.5	7.5
Test	Sentences	1000	
	Test independent	434	
	Words	9507	7453
	Average length	9.5	7.5

Table 5.1: Characteristics of the Tourist corpus.

$$s' = \underset{s^r}{\operatorname{argmax}}\, Pr(s^r) \cdot Pr(s|s^r)$$

where $Pr(s^r)$ is the reordered language model, and $Pr(s|s^r)$ is the reordering model. Being this problem very similar to the translation problem but with a very constrained translation table, it seems only natural to use the same methods developed to solve the translation problem to face the reordering problem. Hence, in this paper we will be using an exponential model as reordering model, defined as:

$$Pr(s|s') \approx exp(-\sum_i d_i)$$

where d_i is the distance between the last reordered word position and the current candidate position.

However, and in order to reduce the error that will introduce a reordering model into the system, we found to be very useful to compute an n-best list of reordering hypothesis and translate them all, selecting then as final output sentence the one which obtains the highest probability according to the models $Pr(t) \cdot Pr(s^r|t)$. Ultimately, what we are actually doing with this procedure is to constrain the search space of permutations of the source sentence as well, but taking into account the information that monotonized alignments entail. In addition, this technique implies a much stronger restriction of the search space than previous approaches, reducing significantly the computational effort needed.

5.1.3 Translation experiments

Corpora used

Basque-Spanish Tourist corpus The *Tourist* corpus [Pérez et al., 2005], is an adaptation of a set of Spanish-German grammars generating bilingual sentence pairs [Vidal, 1997] in such languages. Hence, the corpus is semi-synthetic. In this task, the sentences describe typical human dialogues in the reception desk of a hotel,

		Spanish	Basque
Training	Sentences	14615	
	Different pairs	8462	
	Words	191156	187462
	Vocabulary	722	1149
	Average length	13.1	12.8
Test	Sentences	1500	
	Test independent	702	
	Words	18978	18711
	Average length	12.7	12.5

Table 5.2: Characteristics of MetEus corpus.

being mainly extracted from tourist guides. However, because of its design, there is some asymmetry between both languages, and a concept being expressed in several manners in the source language will always be translated in the same manner in the target language. Because of this, the target language is meant to be simpler than the source language. Since the input language during the design of the corpus was Spanish, the vocabulary size of Basque should be smaller. In spite of this fact, the vocabulary size of Basque is bigger than that of Spanish, and this is due to the agglutinative nature of the Basque language. The corpus has been divided into two separate subsets, a bigger one for training and a smaller one for test. The characteristics of this corpus can be seen in Table 5.1.

Basque-Spanish MetEus corpus MetEus [Pérez et al., 2006] is a weather forecast corpus composed of 28 months of daily weather forecast reports in Spanish and Basque languages. These reports were picked from those published in Internet by the Basque Institute of Meteorology. Again, this corpus consists of two separate subsets: one for training and one for test. The characteristics of this corpus are shown in Table 5.2.

System evaluation

The SMT system developed has been automatically evaluated by measuring the following rates:

WER *(Word Error Rate)*: The WER criterion computes the minimum number of editions (substitutions, insertions and deletions) needed to convert the translated sentence into the sentence considered ground truth. This measure is because of its nature a pessimistic one, when applied to Machine Translation.

PER *(position-independent WER)*: This criterion is similar to WER, but word order is ignored, accounting for the fact that an acceptable (and even grammatically correct) translation may be produced that differs only in word order.

	Baseline	Reordered, $n = 5$
WER	20.7%	16.2%
BLEU	77.9%	79.8%
PER	12.6%	11.0%

	Baseline	Reordered, $n = 5$
WER	19.5%	10.9%
BLEU	81.0%	87.1%
PER	6.2%	4.9%

Table 5.3: Results for Spanish to Basque (left) and Basque to Spanish (right) translation in the Tourist corpus.

BLEU *(Bilingual Evaluation Understudy) score*: This score measures the precision of unigrams, bigrams, trigrams, and 4-grams with respect to a set of reference translations, with a penalty for too short sentences [Papineni et al., 2002]. BLEU is not an error rate, i.e. the higher the BLEU score, the better.

Experimental setup and translation results

We used the reordering technique described above to obtain an n-best reordering hypothesis list and translate them, keeping the best scoring one.

First, the bilingual pairs were aligned using IBM model 4 by means of the GIZA++ toolkit [Och and Ney, 2000]. After this, the alignments were made monotone in the way described in Figure 5.1 and a new alignment was recalculated, determining the new monotone alignment between the reordered source sentence and the target, and a reordered source sentence language model was built. In addition, a phrase based model involving reordered source sentences and target sentences was learnt by using the Thot toolkit [Ortiz et al., 2005].

For the next step, the reordering model, we used the reordering model built in the toolkit Pharaoh. This was done by including in the translation table only the words contained in the vocabulary of the desired source language, and allowing the toolkit to reorder the words by taking into account the language model and the phrase-reordering model it implements, which is an exponential model. Since in this case, the phrases are just words, what results is an effective implementation of an exponential word-reordering model, just as we wanted.

Once the n best reordering hypothesis had been calculated, we translated them all by using Pharaoh once again, and kept the best scoring translation, being the score determined as the product of the (inverse) translation model and the language model.

As a baseline, we took the results of translating the same test set, but without the reordering pipeline, i.e. just using GIZA++ for aligning, Thot for phrase extraction and Pharaoh for translating.

Tourist The results of this setup applied to the Tourist corpus can be seen in Table 5.3, with n-best list size set to 5. At this point, it must be noted that Pharaoh by itself also performs some reordering of the output sentence, but only on a per-phrase basis.

These results show that the reordering pipeline established does have significant benefits on the overall quality of the translation, almost achieving a relative improvement of 50% in WER. Furthermode, it is interesting to point out that even in the case of the PER criterion the results obtained are better. At first sight, this might seem odd, since the PER criterion does not take into account word order errors within a sentence, which is the main problem reordering techniques try to solve. However, this improvement is explained because reordering the source sentence allows for better phrases to be extracted.

It is also interesting to point out that the translation quality when translating from Spanish to Basque is much higher than in the opposite sense. This is due to the corpus characteristics described in the previous section: Spanish being the input language of the corpus, it is only natural that the translation quality will worsen when reversing the meant translation direction. In addition, it can also be observed that the reordering pipeline has less beneficial effects when translating from Basque to Spanish.

Lastly, in Figure 5.3, the result of increasing the size of the n-best reordering hypothesis list can be seen. In the case of Spanish-Basque translation, it can be seen how the translation quality still increases until size 20, where as in the case of Basque-Spanish the translation quality already reaches its maximum with the first 5 best hypothesis. However, it can also be seen that just using the best reordering hypothesis already yields better results than without introducing the reordering pipeline. Hence, these figures also show that the phrase extraction process obtains better quality phrases when the monotonization procedure has been implemented before the extraction takes place.

MetEus corpus Given the extremely encouraging results presented in the previous section, we decided to carry on and test our setup with a more complex corpus. Since our system seemed to perform well with Basque, we decided to continue using Basque and pursued our experiments with the MetEus corpus. The results can be seen in Figure 5.4.

Disappointingly enough, however, the translations of our reordering system did not achieve to perform better than the baseline system, even for n-best lists of size 100.

One possible reason for this is that the MetEus corpus is a much more complex corpus than the Tourist corpus: although the vocabulary sizes are almost identical, the amount of (different) training pairs is almost one third. Moreover, the average sentence length in the MetEus corpus is much higher in relation with the Tourist corpus. This also implies that the exponential reordering model described in 5.1.2 will have a much stronger effect than with short sentences, hindering the appearance in the n-best list of long-term reorderings, which are frequently observed in Basque.

In the figures above only some of the experiments carried out can be seen. Actually, the Thot toolkit can perform phrase extraction according to five different algorithms, named as “and, or, sum, sym1” and “sym2”. See [Ortiz et al., 2005] for a detailed description about how these algorithms are implemented. Experiments were performed with all five intersection operations, and the results can be found in appendix A.

5.1.4 Conclusions

A reordering technique has been implemented, taking profit of the information that monotonized corpora provide. By doing so, better quality phrases can be extracted and the overall performance of the system may improve significantly in the case of a pair of languages with heavy reordering complications.

This technique has been applied to translate a semi-synthetic corpus which deals with the task of Spanish-Basque translation, and the results obtained prove to be statistically significant and show to be very promising, specially taking into account that Basque is an extremely complex language that poses many problems for state of the art systems.

However, the improvements observed in the simple corpus did not carry over to the other more complex corpora on which our system has been tested. A reason for this might be that the sentences in these corpora are way too long for our exponential reordering model to allow enough reordering. Nevertheless, other experiments show that the effectiveness of a reordering step in SMT entails variable results, depending strongly on the language pair. As an example, introducing a reordering model for French to English translation has, to the best of our knowledge, always a negative impact on the translation quality.

The technique we propose in this paper is learnt automatically, without any need of linguistic annotation or manually specified syntactic reordering rules, which means that out technique can be applied to any language pair without need for any additional development effort.

Both reordered corpora and reordering techniques seem to have a very important potential for the case of very different language pairs, which are the most difficult translation tasks. However, a much more thorough insight into this question is needed before extracting definitive conclusions.

5.2 Phrase Table size reduction

Another important drawback of Phrase-Based systems is the enormous size the phase tables need, which has as consequence the high requirements such models need, in terms of space but also time. In this paper, we propose a novel technique for reducing the amount of segment pairs needed for translating a given test set.

Related work was performed by [Johnson et al., 2007], where the authors present a method for reducing the phrase table by performing significance testing. The present work, however, does not perform a statistical analysis of the phrases in the phrase table, but instead uses the concept of optimal segmentation of each sentence pair to reduce significantly the amount of segments to be included in the final phrase table. In addition, a speed analysis of the different systems built, both before and after the reduction, is performed.

5.2.1 Phrase table reduction via suboptimal bilingual segmentation

The problem of segmenting a bilingual sentence pair in such a manner, that the resulting segmentation is the one that contains, without overlap, the best phrases that can be extracted from that pair is a difficult problem. In the first place, because all possible segmentations must be considered, and this number is a combinatorial number. In the second place, because a measure of *"optimality"* must be established. Consider the following example:

Source: *The table is red .*
Target: *La mesa es roja .*

At the sight of this example, one would probably state that {{*The table , La mesa*}, {*is red, es roja*}, {*. , .*}} is a good segmentation for this bilingual pair. However, why is such a segmentation better than {{*The , La*},{*table is , mesa es*},{*red . , roja .*}}? As humans, we could argue with more or less convincing linguistic terms in favour of the first option, but that does not necessarily mean that such a segmentation is the most appropriate one for SMT, and, moreover, one could easily think of several *linguistically appropriate* segmentations of this small example. To overcome this problem, PB SMT systems are forced to extract a large number of possible overlapping segmentations, and hope that one of them will be useful. Obviously, such an aggressive approach is bound to be computationally costly, and decoding time greatly suffers because of this issue.

When considering all possible segmentations of a bilingual sentence pair and assuming a "bag of words" model for the target sentence, the probability $Pr(\mathbf{x}|\mathbf{y})$ in Equation 1.3 can be modelled as:

$$P(\mathbf{x}|\mathbf{y}) = \sum_{K} \sum_{\mu} \sum_{\gamma} \prod_{k=1}^{K} p(x_{\gamma_{k-1}+1}^{\gamma_k} | y_{\mu_{k-1}+1}^{\mu_k}) \tag{5.1}$$

where K is the number of bilingual segments into which each bilingual pair is divided, μ is the set of possible segmentations of the source sentence $\mathbf{x}$ and γ the set of possible

segmentations of the target sentence $\mathbf{y}$. In this formula we have assumed monotonic translation, in which no word (or segment) reordering is performed for the sake of simplicity.

Our approach for solving the problem of the overwhelming amount of possible segmentations, and the consequent increase of the phrase table, is based on the concept of Viterbi re-estimation [Viterbi, 1967]. Following this idea, we can approximate $P(\mathbf{x}|\mathbf{y})$ by changing the summations by maximisations:

$$P(\mathbf{x}|\mathbf{y}) \approx \hat{P}(\mathbf{x}|\mathbf{y}) = \max_{K} \max_{\mu} \max_{\gamma} \prod_{k=1}^{K} p(x_{\gamma_{k-1}+1}^{\gamma_k}|y_{\mu_{k-1}+1}^{\mu_k}) \tag{5.2}$$

Given that the phrase table establishes the probability of an input segment given a certain output segment, we can use the scores within the phrase table to compute $\hat{P}(\mathbf{x}|\mathbf{y})$, and then build a phrase table by only taking into account those segments used to compute the optimal segmentation of each bilingual sentence in the training corpus.

However, computing $\hat{P}(\mathbf{x}|\mathbf{y})$ according to a given phrase table is not an easy task: if we establish a certain maximum length for the segments contained in the phrase table, it is common that, due to non-monotonic alignments, certain words of a sentence will not be contained in the segments extracted. Observing all possible segments without constraining the maximum length is not a solution either, since the number of entries in the phrase table would grow too much. This implies that the phrase table has coverage problems even on the training set.

Nevertheless, our intention is to discard unnecessary segment pairs contained in the phrase table. To this purpose, a *suboptimal* bilingual segmentation, in which we *translate* the source sentence, may be enough. We are aware, nevertheless, that translating the input sentence will not necessarily produce the output sentence in the training pair, but our experiments show that this might be good enough to prune the phrase table without a significant loss in translation quality.

5.2.2 Experiments

We conducted our experiments on the Europarl corpus [Koehn, 2005], described in Section 1.3. The translation systems were tuned using the development set with the MERT [Och et al., 2003] optimisation procedure, where the measure to be optimised was BLEU [Papineni et al., 2002].

We performed experiments on both test sets, yielding similar results for both of them. Because of this, and in order not to provide an overwhelming amount of results, we only report the results obtained with the Test set, being this result more interesting because of the out-of-domain data it contains.

Suboptimal segmentation filtering

As a baseline system, we used the same system as the one used in the workshop. To filter the phrase table as described in the previous section, we translated the whole training subcorpus using the baseline model, and kept only those entries of the phrase

Table 5.4: Performance comparison between the baseline system and our suboptimal-segmentation-reduced approach. Lexicalised reordering is considered. *Speed* is measured in number of translated source words per second, and *fsize* is the size of the phrase table when filtered for the test set.

	baseline					reduced						
pair	WER	BLEU	size	fsize	speed	WER	BLEU	size	fsize	speed	size red.	S_p
Es–En	57.8	30.6	19M	1.6M	5.3	57.5	30.9	1.9M	0.15M	13.1	91%	2.5
En–Es	57.5	30.3	19M	1.8M	5.7	57.4	30.6	1.7M	0.16M	11.3	92%	2.0
De–En	68.1	23.7	12M	1.1M	6.6	68.2	23.9	1.8M	0.18M	11.4	84%	1.7
En–De	72.5	16.4	13M	1.7M	4.3	72.4	16.5	1.9M	0.23M	9.0	86%	2.1
Fr–En	60.2	28.3	15M	1.6M	5.6	60.1	28.3	1.5M	0.12M	17.7	92%	3.2
En–Fr	60.5	30.5	16M	1.7M	4.5	60.1	30.9	1.6M	0.15M	9.5	91%	2.1

table which were used while doing this. Since the baseline system uses lexicalised reordering [Koehn et al., 2005], we also filtered the reordering table according to the segments used. The result of this setup can be seen in Table 5.4.

In this table, the sizes are given in number of entries in the phrase table and the speed is given in words per second. *fsize* is the size of the phrase table after filtering out all segments which will not be needed for translating the current test set, which is usual when dealing with big phrase tables. In this context, it must be noted that the translation speed detailed in Table 5.4 was measured in all cases when translating using the filtered phrase table, since loading the complete phrase table into memory without any filtering is unfeasible with the baseline model. Moreover, the speed does not take into account the time the system needs to load the model files (i.e. phrase table and lexicalised reordering table), which is reduced in a factor of ten due to the difference in model size. S_p is the *speedup*, which is given by the formula $S_p = T_b/T_r$, where T_b is the time taken by the baseline system and T_r is the time taken by the filtered system. The values appearing as "size red." in the table represent the *fsize* reduction in percentage with respect to the original *fsize*. Hence, this column displays the effective reduction of data loaded into the decoder when translating.

Translation quality, as measured with BLEU [Papineni et al., 2002] is not affected by the reduction of the size of the phrase table we proposing. Moreover, we can see that, in the worst case, we get exactly the same score than with the baseline system, and in the best case we are improving BLEU by 0.35 points. As measured with WER, which is an adaptation of the edit distance used in Speech Recognition, the translation quality is slightly worsened in some cases (with a maximum of 0.1 points), and in some cases improved. The behaviour difference between BLEU and WER can be explained because of the measure being optimised in MERT, which was BLEU.

Although the differences named in the previous paragraph are not significant, it is important to stress that we are improving translation speed by a factor of 3.2 in the best case and 1.7 in the worst case, without a significant loss of translation quality even in cases where out-of-domain sentences were translated.

Table 5.5: Performance comparison between the baseline system and our suboptimal-segmentation-reduced approach. Monotonic search is considered. *Speed* is measured in number of translated source words per second, and *fsize* is the size of the phrase table when filtered for the test set.

	baseline				reduced				
pair	WER	BLEU	fsize	speed	WER	BLEU	fsize	speed	S_p
Es–En	58.8	29.6	1.6M	17.6	58.4	29.7	0.13M	91.5	5.2
En–Es	58.5	29.2	1.8M	19.1	58.6	29.2	0.08M	125.0	6.5
De–En	68.9	22.6	1.1M	20.6	69.0	22.5	0.14M	107.0	5.2
En–De	73.1	16.0	1.7M	23.5	72.6	16.2	0.20M	80.0	3.4
Fr–En	60.3	27.6	1.6M	15.8	60.9	27.4	0.11M	147.0	9.3
En–Fr	61.7	29.4	1.7M	19.0	61.5	29.4	0.16M	74.7	3.9

Increasing translation speed further

Although the speeds achieved in the previous subsection are already competitive, they may not be enough for real time applications: translating an average sentence of 25 words may take more than two seconds, and this might not be enough for the user who is waiting for the translation.

A common resource for increasing translation speed is to consider only monotonic translation. Under this decoding strategy, a given bilingual segment must occupy the same position in both input and output sentences. For example, if the source part of a certain bilingual segment is placed at the start of the source sentence, it cannot be placed at the end of the target sentence (or anywhere else but at the start). Although it is true that some translation quality is lost by doing so, the difference is relatively small the language pairs considered in our work. Our phrase table reduction technique can also be applied to monotonic translation. The results for this setup are shown in Table 5.5, yielding, again, no significant worsening (or improvement) of the translation scores, but achieving speedups ranging from 3.2 to 9.5, depending mainly on the language pair chosen and when compared to the non-reduced monotonic search.

In this case, it must be emphasised that the *fsize* of the baseline is the same as in the case of the lexicalised reordering search, since the reordering has no effect on the number of phrases extracted. This is not so, however, with our suboptimal segmentation, since the monotonicity constraint is also imposed when obtaining the segments that will be part of the final phrase table, which implies that fewer (but shorter) segments will be kept.

5.2.3 Analysis and side notes

A question which could be asked at this point is whether we can truly obtain the same translation quality by just taking into account the suboptimal segmentation, or rather what we are doing is simply a filtering, but we actually would need the probabilities contained within the complete phrase table. In order to clarify this, we re-normalised

Table 5.6: Performance as measured by BLEU and WER for the re-normalised system. Both monotonic and non-monotonic search are considered.

	baseline				re-normalised			
	monotonic		reordering		monotonic		reordering	
pair	WER	BLEU	WER	BLEU	WER	BLEU	WER	BLEU
Es–En	58.8	29.6	57.8	30.6	59.0	29.1	57.8	30.5
En–Es	58.5	29.2	57.5	30.3	58.8	29.0	57.6	30.4
De–En	68.9	22.6	68.1	23.7	69.1	22.5	68.3	23.8
En–De	73.1	16.0	72.5	16.4	72.7	16.3	72.7	16.4
Fr–En	60.3	27.6	60.2	28.3	61.0	27.2	60.2	28.1
En–Fr	61.7	29.4	60.5	30.5	61.8	29.3	60.4	30.9

Table 5.7: BLEU and WER scores for the Training set, with both monotonic and non-monotonic search.

pair	monotonic		reordering	
	WER	BLEU	WER	BLEU
Es-En	44.9	48.2	43.2	50.6
En-Es	47.1	46.3	44.8	49.4
De-En	53.9	41.6	51.8	43.6
En-De	55.6	37.9	55.6	37.9
Fr-En	46.7	45.9	46.9	46.0
En-Fr	51.5	44.4	46.4	49.8

the phrase table, assigning to each segment the score obtained by only taking into account those phrase pairs contained within the reduced phrase table. In Table 5.6 we can see the results of performing such a renormalisation.

As can be seen in the table, the performance is not significantly affected by the renormalisation. In our opinion, this clearly reveals that computing the phrase translation probabilities by only taking into account the segments used to translate the training set obtains a similar result than taking into account all possible segmentations that are consistent with the word alignments, as is common in regular SMT systems. A possible interpretation is that those segments which were selected to stay in the final, filtered table are those which account for the biggest part of the probability mass.

Lastly, and since we already had translated the training set, we found interesting to compute the BLEU and WER scores over the training data. These scores, which can be seen in Table 5.7, constitute an upper bound of the score that could be achieved in the test set. However, these results are not as good as could be expected, which hints towards a relatively weak (but even though state-of-the-art) performance of the translation models and (or) decoding algorithm.

5.2.4 Conclusions and future work

In this work we have presented a straight-forward method for reducing the size of the phrase table by a factor of ten, and increasing translation speed up to nine times. By doing so, the translation quality as measured by WER and BLEU remains unaffected, for both in-domain and out-of-domain data. Given that translation speed is a serious issue in systems implementing phrase-based models, the approach presented in this paper provides an efficient solution for the problem.

As future work, we are planning on researching ways to obtain the optimal segmentation of the sentences in the training corpus, without going through the drawback of having to translate the corpus. This includes both segmenting the sentences according to a phrase table, and without having the phrase table as a starting point.

Chapter 6

Scientific publications

The content of this thesis has been published in several national and international workshops and conferences. In this section, we review these publications and their relation with the topics of this thesis.

The content of Chapter 4 was published in an international workshop and an international conference:

- **G. Sanchis-Trilles**, M.T. González, F. Casacuberta, E. Vidal, J. Civera. Introducing Additional Input Information into IMT Systems. In *Proceedings of the 5th Joint Workshop on Multimodal Interaction and Related Machine Learning ALgorithms*, volume 5237 of *Lecture Notes in Computer Science*, pages 284–295. Sringer-Verlag. Utrecht (The Netherlands), Septiembre 2008. (CORE category B)

- **G. Sanchis-Trilles**, D. Ortiz-Martínez, J. Civera, F. Casacuberta, E. Vidal, H. Hoang. Improving Interactive Machine Translation via Mouse Actions. In *Proceedings of the 2008 conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii (USA), October 2008. (CORE category A)

The work presented in Chapter 2 originated two publications, one in an international conference, and one in a national workshop. However, some of the work is still unpublished.

- **G. Sanchis-Trilles**, J.A. Sánchez. Using Parsed Corpora for Estimating Stochastic Inversion Transduction Grammars. In *Proceedings of the 6th edition of the International Conference on Language Resources and Evaluation*. Marrakech, Morocco, May 2008. (CORE category C)

- **G. Sanchis-Trilles**, J.A. Sánchez. Phrase segments obtained with Stochastic Inversion Transduction Grammars for Spanish-Basque Translation. Accepted for publication in the *V Jornadas en Tecnología del Habla*. Bilbao, Spain, November 2008.

The reordering technique described in Chapter 5 was published in a national workshop and an international conference:

- **G. Sanchis-Trilles**, F. Casacuberta. N-Best reordering in Statistical Machine Translation. In *Proceedings of IV Jornadas en Tecnología del Habla.* Zaragoza, Spain, November 2006.
- **G. Sanchis-Trilles**, F. Casacuberta. Reordering via N-best lists for Spanish-Basque translation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation.* Skövde, Sweden, September 2007.

The phrase table size reduction technique originated a publication in an international conference and an international workshop:

- J. González, **G. Sanchis-Trilles**, F. Casacuberta. Learning Finite State Transducers using Bilingual Phrases. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*, volume 4919 of *Lecture Notes in Computer Science*, pages 411–422. Springer-Verlag. Haifa, Israel, February 2008. (CORE category B)
- **G. Sanchis-Trilles**, F. Casacuberta. Increasing translation speed in phrase-based models via suboptimal segmentation. In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems (PRIS 2008)*, Barcelona, Spain, June 2008.

Finally, other publications with significant contributions from the author of this thesis appeared in two international conferences and an international workshop:

- J. González-Rubio, **G. Sanchis-Trilles**, A. Juan, F. Casacuberta. A novel alignment model inspired on IBM model 1. In *Proceedings of the 12th Annual Meeting of the European Association for Machine Translation*, Hamburg, Germany, October 2008.
- **G. Sanchis-Trilles**, J.A. Sánchez. Vocabulary extension via pos information for SMT. In *Proceedings of Mixing Approaches to Machine Translation*, San Sebastián, Spain, February 2008.

Chapter 7

Acknowledgements

The work reflected in this Master's Thesis was partially supported by the Spanish MEC under grants TIC 2003-08681-C02-02, CONSOLIDER Ingenio-2010 CSD2007-00018 and scholarship AP2005-4023, by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01, by the Universidad Politécnica de Valencia with the ILELA project, and by the Generalitat Valenciana under grant GVPRE/2008/331, research project "Traducción Automótica del Corpus UPenn Treebank mediante Tócnicas Interactivas (UPennSpanish)."

Bibliography

[Arnold, 2003] Arnold, D. J. (2003). *Computers and Translation: A translator's guide*, chapter 8, pages 119–142.

[Asterias et al., 2006] Asterias, J., Casas, B., Comelles, E., González, M., Padró, L., and Padró, M. (2006). Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy.

[Barrachina et al., 2008] Barrachina, S., , Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Ney, A. L. H., Tomás, J., and Vidal, E. (2008). Statistical approaches to computer-assisted translation. *Computational Linguistics*, page In press.

[Berger et al., 1996] Berger, A., Brown, P., Pietra, S. D., Pietra, V. D., Gillet, J., Kehler, A., and Mercer, R. (1996). Language translation apparatus and method of using context-based translation models. In *United States Patent 5510981*.

[Brown et al., 1993] Brown, P., Pietra, S. D., Pietra, V. D., and Mercer, R. (1993). The mathematics of machine translation. In *Computational Linguistics*, volume 19, pages 263–311.

[Callison-Burch et al., 2007] Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

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

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

[Esteban et al., 2004] Esteban, J., Lorenzo, J., Valderrábanos, A., and Lapalme, G. (2004). Transtype2 - an innovative computer-assisted translation system. In *Companion Volume to the Proceedings of the 42nd Annual Meeting of the Association for Computational Linguitics*, pages 94–97, Barcelona, Spain.

[F. Pereira, 1992] F. Pereira, Y. S. (1992). Inside-outside reestimation from partially bracketed corpora. In *Proceedints of the 30th annual meeting of the association for computational linguistics*, Newark, Deleware, USA.

[Fordyce, 2007] Fordyce, C. S. (2007). Overview of the IWSLT 2007 evaluation campagin. In *International Workshop on Spoken Language Translation*, Trento, Italy.

[Foster, 2002] Foster, G. (2002). *Text Prediction for Translators*. PhD thesis, Université de Montréal.

[Foster et al., 2002] Foster, G., Langlais, P., and Lapalme, G. (2002). User-friendly text prediction for translators. In *Proc. of EMNLP'02*, pages 148–155.

[Germann et al., 2001] Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proceeding of the 39th. Annual Meeting of the ACL*, pages 228–235, Toulouse, France.

[Hutchins, 1999] Hutchins, J. (1999). Retrospect and prospect in computer-based translation. In *Proceedings of MT Summit VII*, pages 30–44.

[Johnson et al., 2007] Johnson, J., Martin, J., Foster, G., and Kuhn, R. (2007). Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic.

[Kanthak et al., 2005] Kanthak, S., Vilar, D., Matusov, E., Zens, R., and Ney, H. (2005). Novel reordering approaches in phrase-based statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 167–174, Ann Arbor, Michigan.

[Kay, 1997] Kay, M. (1997). It's still the proper place. *Machine Translation*, 12(1–2):35–38.

[Klein and Manning, 2003] Klein, D. and Manning, C. D. (2003). Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, pages 3–10.

[Knight, 1999] Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

[Koehn, 2004] Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP'04*, pages 388–395, Barcelona, Spain.

[Koehn, 2005] Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit X*, pages 79–86.

[Koehn et al., 2005] Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

[Koehn et al., 2007] Koehn, P. et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL'07*.

[Koehn and Monz, 2006a] Koehn, P. and Monz, C. (2006a). Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the NAACL 2006, Workshop on SMT*, pages 102–121, New York City.

[Koehn and Monz, 2006b] Koehn, P. and Monz, C., editors (2006b). *Proceedings of the Workshop on Statistical Machine Translation*.

[Koehn et al., 2003] Koehn, P., Och, F., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conf. of the NAACL on Human Language Technology*, volume 1, pages 48–54, Edmonton, Canada.

[Kumar and Byrne, 2003] Kumar, S. and Byrne, W. (2003). A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proceedings of the 2003 Conference of the NAACL on Human Language Technology*, volume 1, pages 63–70, Edmonton, Canada.

[Kumar and Byrne, 2005] Kumar, S. and Byrne, W. (2005). Local phrase reordering models for statistical machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 161–168, Vancouver, Canada.

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

[Marcu and Wong, 2002] Marcu, D. and Wong, W. (2002). Joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP02)*, Pennsylvania, Philadelphia, USA.

[Matusov et al., 2005] Matusov, E., Kanthak, S., and Ney, H. (2005). Efficient statistical machine translation with constrained reordering. In *In Proceedings of EAMT 2005 (10th Annual Conference of the European Association for MT)*, pages 181–188, Budapest, Hungary.

[Och, 2003] Och, F. (2003). Minimum error rate training for statistical machine translation. In *Proc. of ACL'03*, pages 160–167.

[Och and Ney, 2002] Och, F. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the ACL'02*, pages 295–302.

[Och and Ney, 2004] Och, F. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449.

[Och et al., 1999] Och, F., Tillmann, C., and Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proc. of EMNLP/WVLC'99*, pages 20–28.

[Och et al., 2003] Och, F., Zens, R., and Ney, H. (2003). Efficient search for interactive statistical machine translation. In *Proc. of EACL'03*, pages 387–393.

[Och and Ney, 2000] Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *ACL 2000*, pages 440–447, Hongkong, China.

[Ortiz et al., 2003] Ortiz, D., García-Varea, I., and Casacuberta, F. (2003). An empirical comparison of stack-based decoding algorithms for statistical machine translation. In *New Advance in Computer Vision, Springer-Verlag, Lecture Notes in Computer Science, 1st Iberian Conference on Pattern Recongnition and Image Analysis (IbPRIA2003)*, Mallorca, Spain.

[Ortiz et al., 2005] Ortiz, D., García-Varea, I., and Casacuberta, F. (2005). Thot: a toolkit to train phrase-based statistical translation models. In *Tenth Machine Translation Summit*, pages 141–148. Asia-Pacific Association for MT, Phuket, Thailand.

[Ortiz-Martínez et al., 2005] Ortiz-Martínez, D., García-Varea, I., and Casacuberta, F. (2005). Thot: a toolkit to train phrase-based statistical translation models. In *Proc. of the MT Summit X*, pages 141–148.

[Papineni et al., 1998] Papineni, K., Roukos, S., and Ward, T. (1998). Maximum likelihood and discriminative training of direct translation models. In *Proc. of ICASSP'98*, pages 189–192.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proc. of ACL'02*.

[Pérez et al., 2005] Pérez, A., Casacuberta, F., Torres, M., and Guijarrubia, V. (2005). Finite-state transducers based on k-tss grammars for speech translation. In *Proceedings of Finite-State Methods and Natural Language Processing (FSMNLP 2005)*, pages 270–272, Helsinki, Finland.

[Pérez et al., 2006] Pérez, A., Torres, I., Casacuberta, F., and Guijarrubia, V. (2006). A Spanish-Basque weather forecast corpus for probabilistic speech translation. In *Advances in Natural Language Processing, Proceedings of 5th International Conference on NLP, FinTAL 2006*, volume 4139 of *Lecture Notes in Computer Science*, pages 716–725. Springer, Turku, Finland.

[Sánchez and Benedí, 2006a] Sánchez, J. and Benedí, J. (2006a). Obtaining word phrases with stochastic inversion transduction grammars for phrase-based statistical machine translation. In *Proc. 11th Annual conference of the European Association for Machine Translation*, pages 179–186, Oslo, Norway.

[Sánchez and Benedí, 2006b] Sánchez, J. and Benedí, J. (2006b). Stochastic inversion transduction grammars for obtaining word phrases for phrase-based statistical machine translation. In *Proceedings of the Workshop on SMT*, pages 130–133, New York City.

[Snover et al., 2006] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA.

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

[Ueffing et al., 2002] Ueffing, N., Och, F., and Ney, H. (2002). Generation of word graphs in statistical machine translation. In *Proc. of EMNLP'02*, pages 156–163.

[Vidal, 1997] Vidal, E. (1997). Finite-state speech-to-speech translation. In *Proceedings of ICASSP-97*, volume 1, pages 111–114, Munich, Germany.

[Vidal et al., 2007] Vidal, E., Rodríguez, L., Casacuberta, F., and García-Varea, I. (2007). Interactive pattern recognition. In *Proceedings of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Volume 4892 of LNCS*, pages 60–71.

[Vilar et al., 1996] Vilar, J., Vidal, E., and Amengual, J. (1996). Learning extended finite-state models for language translation. In *Proc. of Extended Finite State Models Workshop (of ECAI'96)*, pages 92–96, Budapest.

[Viterbi, 1967] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. 13:260 – 269.

[Watanabe et al., 2003] Watanabe, T., Sumita, E., and Okuno, H. (2003). Chunk-based statistical translation. In *Proceedings of the 41st. Annual Meeting of the ACL*, Sapporo, Japan.

[Wu, 1997] Wu, D. (1997). Stochastic iversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

[Zens and Ney, 2003] Zens, R. and Ney, H. (2003). A comparative study on reordering constraints in statistical machine translation. In *Annual Meeting of the Assoc. for Computational Linguistics*, pages 144–151, Sapporo, Japan.

[Zens and Ney, 2004] Zens, R. and Ney, H. (2004). Improvements in phrase-based statistical machine translation. In *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, pages 257–264, Boston, USA.

[Zens et al., 2004] Zens, R., Ney, H., Watanabe, T., and Sumita, E. (2004). Reordering constraints for phrase-based statistical machine translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pages 205–211, Geneva, Switzerland.

[Zens et al., 2002] Zens, R., Och, F., and Ney, H. (2002). Phrase-based statistical machine translation. In *Advances in artificial intelligence. 25. Annual German Conference on AI. Lecture Notes in Computer Science*, volume 2479, pages 18–32.