Learning the latent structure of translation
Mylonakis, M.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
In this chapter we build on the results of Chapter 4 to extend our method to the estimation of syntactically driven translation models, using binary Synchronous Context-Free Grammars (SCFGs) as our formalism of choice. As part of our work on estimating the parameters of phrase-based translation models, we already focused on the space of binary phrase re orderings to constrain successfully the segmentation space and define a prior over it. Now, we move further to bring the recursive nature of binary SCFGs at the centre of our attention, formulating a probabilistic SCFG joint model to describe translation.

Increasing the modelling stress on the latent translation structure necessitates a closer examination of its role and possible deficiencies. For this reason, we empirically consider alternatives in order to identify a translation structure strong enough to function as the backbone holding together the source and target sides of our modelling problem. Noting certain deficiencies of the independence assumptions behind SCFGs, we contribute a lexically sensitive reordering structure which propagates reordering decisions to higher and lower levels of a derivation, in order to widen the role of the abstract recursive translation structure past the rudimentary use that it finds in (Chiang, 2005a).

In comparison with the state-of-the-art, this work as first presented in (Mylonakis and Sima’an, 2010) contributes a method to learn phrase-based synchronous grammars for machine translation, aiming to discover reusable lexical and structural translation patterns which generalise well. We further contribute a particular grammar formalism which puts the focus on orchestrating phrase reordering across the full length of the sentence-pair.

We do not explore synchronous grammars which enrich synchronous productions with lexical context and which allow modelling translation with discontiguous phrases. While these grammars have been shown to offer competitive translation performance (Chiang, 2005a), in this chapter we choose to focus on the implications of learning the unlexicalised recursive structure of synchronous
Chapter 5. Learning Stochastic Synchronous Grammars

grammars. Nevertheless, our learning objective and implementation based on CV-EM, together with the grammar design we contribute, allows us to reach the same level of performance as our hierarchical phrase-based translation baseline which does use lexicalised recursive reordering.

In the context of the thesis, in this chapter we move further than merely estimating phrase translation probabilities as we did in Chapter 4, and integrate them as part of a more comprehensive model which handles all other aspects of translation such as reordering. We proceed towards learning a full translation model capturing both phrase translation and reordering patterns, which will be the key component taking care of the lexical and structural transfer from source to target during decoding. We consider the implications of describing the latent translation structure using the synchronous grammar formalism and take the first step in formulating a learning environment for a relatively simple grammar design. Our findings form a crucial step before proceeding in the following chapter towards discovering intricate, linguistically motivated grammatical structures capturing the translation process.

5.1 Focusing on Translation Structure

Probabilistic phrase-based synchronous grammars are currently considered promising devices for Statistical Machine Translation (SMT), with systems based on this formalism achieving state-of-the-art translation performance. This is especially true when these models are applied to translate between languages with significant differences in their syntax such as Chinese-English. Modelling translation using phrase-based Synchronous Context-Free Grammars (Wu, 1997; Chiang, 2005a) builds upon the strengths of Phrase-Based SMT (PBSMT) (Och et al., 1999; Koehn et al., 2003), while bringing together and extending the different components of a phrase-based system under a single modelling component.

On the one hand, probabilistic SCFGs inherit from the PBSMT models the ability to build models that can reuse memorised multi-word fragments and their translations. This is a powerful feature that essentially allows forfeiting for certain translation patterns the strong independence assumptions posed by word-based models. On the other hand, using synchronous grammars for SMT recasts the reordering problem in terms of establishing a syntactic correspondence between the two languages, unifying the usually separately conceived phrase translation and reordering components of PBSMT systems in a single grammatical formalism (Wu, 1997). In addition, the recursive nature of SCFGs coupled with the concept of modelling translation on the phrase level allows the formulation of hierarchical phrase-based models (Chiang, 2005a) making use of recursive lexical translation patterns, sometimes colloquially referred to as phrase-pairs with ‘gaps’.

In general, discontiguous phrase translation patterns need not necessarily be modelled through a synchronous grammar formalism. This is exhibited by
(Simard et al., 2005), where a standard contiguous PBSMT framework is extended to allow non-contiguous phrase-pairs with missing word placeholders. Still, the recursive nature of SCFGs, apart from the mechanics to allow modelling translation with discontiguous phrase-pairs, puts in place the necessary descriptive power to capture the impact of the hierarchical nature of language in translation.

Nevertheless, following their introduction, the focus on applying hierarchical SCFG-based models was mostly concentrated on the ability to model translation using discontiguous phrase-pairs, leaving the capacity of the model to handle short and long-range syntactic constraints in abstract terms mostly unexploited. This has been largely handled in lexical terms, through the use of recursive phrases with gaps which can trigger certain reorderings in relation with lexical context patterns. However, developments over the last few years have shifted the attention of the research community towards the ability of SCFGs to describe the structural aspects of translation on a level further afield than the lexical surface.

Crucially, the transition from PBSMT to SCFG-based translation was not marked by a similar step towards a better-founded learning framework, leaving the stochastic part of hierarchical models to be founded on the same heuristic methods used in PBSMT. Estimation based on the extraction counts of phrase-pairs was extended to their discontiguous counterparts (Chiang, 2005a), sometimes reaching past the lexical surface and up to the structural part of SCFG analyses (Zollmann and Venugopal, 2006). Some of the reasons behind opting for heuristic estimation in syntax-driven, phrase-based SMT approaches are similar to those encountered in their Phrase-Based SMT forerunners. Embedding the concept of modelling with multi-word fragments of arbitrary lengths in a syntactic framework does not make us any less liable to the same estimation challenges related to contiguous phrase-based models.

Even though the issues related to learning phrase-based models alone are daunting enough, learning synchronous grammars brings in additional aspects to the learning problem. Apart from lexical choice, it also involves training a structural component which takes over the reordering task from the reordering models of PBSMT. This modelling component is concerned with the syntactic well-formedness of the whole sentence, matching long-range syntactic preferences that the reordering models of PBSMT do not usually consider. In addition, the lexical and the structural parts of synchronous grammars can be tightly interlocked together, with the syntactic structure affecting the corresponding lexical choice and vice versa.

As a result of this interplay between the lexical and the structural aspect of synchronous grammars, the estimation challenges of phrase-based models reach out to the structural part as well. The tendency to overfit guides the estimator towards hypotheses translating as much of the source sentence as possible as part of long discontiguous phrase-pairs. This prohibits learning how to combine smaller fragments together and results in models which support only a trivial structure reaching up to the largest fragments allowed by the training constraints.
Avoiding such degenerate hypotheses will allow the estimator to discover not only reasonable phrase correspondences which we hope will be useful to analyse yet unseen data, but also to learn how to combine together these reusable building blocks recursively. The learning environment we will use to work towards this aim will again be the Cross-Validated EM algorithm of section 3.2.

5.2 Synchronous Grammars for SMT

Synchronous grammars extend the descriptive power of formal grammars from single strings to tuples of strings. They can be used to define a language over pairs of strings and are highly interesting for machine translation, as they can capture the correspondences between source sentences and their target language translations. Furthermore, each particular grammar formalism may offer an explanation of the compositional mechanics of translation which allows us to describe compactly the correspondences between a countably infinite set of sentence-pairs. While we may consider a wide range of such formalisms\(^1\), the one which enjoys the widest acceptance in the MT community are the Synchronous Context-Free Grammars (SCFGs) of (Wu, 1997) and (Chiang, 2005a).

As we discuss in more detail in section 2.4, an SCFG defines a language over string-pairs by means of a recursive rewrite process. In a monolingual Context-Free Grammar, starting from a start symbol \(S\) we recursively expand left-hand side non-terminals according to the right-hand side of grammar production rules, rewriting each non-terminal as a string of terminals and novel non-terminals which need to be further expanded. This process continues until we end up with a string of terminal symbols, which then by definition belongs to the language of the grammar. In SCFGs, this rewrite process is \textit{synchronous}, operating on a pair of strings of terminals and pair-wise linked non-terminals, expanding at every rewrite step a single such pair of non-terminals in both sides of the string-pair according to the grammar rules. These rules map a left-hand side of a single non-terminal pair towards a right-hand side consisting of a pair of strings of terminals and non-terminals, with the latter paired together across both sides of the right-hand side expansion.

An example on how the synchronous rewrite process can be employed to capture translation phenomena between language-pairs, repeated here from Chapter 2 for the reader’s convenience, is presented in Figure 5.1. There, we denote linked non-terminals across both parts of the right-hand sides of the synchronous rules by attaching the same subscript indexes, while, to simplify notation, we assume without loss of generality that each linked pair involves non-terminals of the same type. The handful of rules in this small grammar already showcases how SCFG rewrite rules have the potential to encode the abstract syntactical transformations

\(^1\)Some of the other formalisms to describe bilingual data are listed in section 2.4.1.
5.2. Synchronous Grammars for SMT

Figure 5.1: An SCFG rule set for SVO to SOV reordering and question construction from English to (romanised) Japanese, adapted from (Chiang, 2005b)

$S \rightarrow X \, / \, X$

$S \rightarrow \text{Do} \, X \, ? \, / \, X \, \text{ka} \,$

$X \rightarrow \text{NP} \, \text{VB} \, \text{NP} \, / \, \text{NP} \, \text{NP} \, \text{VB} \,$

$NP \rightarrow \text{I} \, / \, \text{watashi ga}$

$VB \rightarrow \text{open} \, / \, \text{akemasu}$

$NP \rightarrow \text{the book} \, / \, \text{hako o}$

and reordering patterns as well as the lexical correspondences between the language pair, possibly combining both abstract and lexical aspects in single rewrite operations (rule 2).

5.2.1 Grammar Design

Approaches considering the use of recursive structure and formal grammars for MT draw inspiration from the related monolingual task of natural language parsing. Superficially, the flavour of syntactic MT that is relevant for this thesis seems highly related to the majority of research on natural language parsing, as they are both occupied with analysing human language manifestations drawing from a common pool of resources, i.e. formal grammars and the related algorithms and learning frameworks. Nevertheless, while a certain link and influence between the two fields undeniably exists, there are fundamental differences with respect to the role of syntax in the learning problems behind syntactic MT and parsing.

Firstly, the structure of Machine Translation is latent, rendering the problem of identifying it as an instance of unsupervised learning. On the contrary, the majority of research on natural language parsing is occupied with the supervised learning of a predefined flavour of language structure, using labelled corpora such as the Penn Treebank (Marcus et al., 1993). While all kinds of learning share common problems such as overfitting and treating yet unseen instances, the unsupervised nature of learning syntactic models for MT brings in the novel challenge of learning from incomplete data, in comparison to supervised monolingual parsing.

Still, one could argue perhaps that syntactic MT is more reminiscent then of the field of unsupervised parsing (van Zaanen, 2000; Clark, 2001; Klein and Manning, 2004; Bod, 2007), which considers the unsupervised learning of lan-
Chapter 5. Learning Stochastic Synchronous Grammars

guage structure from unlabelled corpora. However, while in both cases the task
is to learn latent natural language structure training merely on the lexical sur-
face, there is a crucial difference between the two in relation to the role of the
latent structure in respect to the overall NLP system. It has been recognised (e.g.
(Bod, 2007)) that, in the long term, attention in evaluating unsupervised pars-
ing must be shifted towards more high-level tasks taking advantage of syntactic
analyses, such as sense disambiguation and MT itself. Nevertheless, for the time
being, most work on unsupervised parsing is fixated and evaluated on replicating
certain linguistic annotations, like those (derived from) the aforementioned Penn
Treebank. As long as discovering the syntactic structure of language remains the
final aim of unsupervised parsing, it will boil down to discovering a certain kind
of syntax, as otherwise a meaningful comparison between the different approaches
seems impossible.

In contrast, in Machine Translation any latent variable assumed by a model is
usually not interesting on its own, and is evaluated instead in the context of how
well it captures the correspondence between the sentences of the language-pair.
This extends to the syntactic variables used in MT models, such as those backed
by SCFGs. Our aim is to raise the translation performance, by integrating as
part of an MT model syntactic formalisms and annotations. The extent to which
we will be successfull in this relies on our capacity to learn these latent variables
from the incomplete parallel corpus and subsequently translate better employing
them. While towards this end features of linguistic syntax as they evolved for
monolingual parsing can be useful, overall syntax-based MT is not bound to a
particular annotation scheme.

This leaves substantial space to consider different synchronous grammar de-
signs to explain the translation process, and we venture to explore part of this
space in this thesis. Figures 5.2 and 5.3 showcase two different views on a syntac-
tic analysis of the translation of secondary clauses between English and German,
using the sentence fragment pair ‘which is the solution / der die Lösung ist’ as
a particular example. The grammar of Figure 5.2 uses the linguistic structure
of the English sentence to pivot between the two languages, while that of Figure
5.3 focuses on lexical cues to signal the characteristic reordering of verbs in these
sentence-pairs. Finally, Figure 5.4 takes a hybrid approach, reducing the ambiguity
when applying the lexically grounded rules of Figure 5.3 using linguistic
constituency information.

One may argue about the merits of each grammar design on linguistic, cogni-
tive or other grounds and these arguments are valid as long as we wish to move
further than the machine translation task, e.g. by aiming to discover how the
human brain translates and so forth. Still, as long as translating automatically is
what we aim for and systems are evaluated on the quality of translations that they
offer, establishing different grammar designs and choosing between them remains
an empirical task. It involves assessing not only the descriptive powers of each
synchronous grammar family, but also our ability to learn an effective synchronous
5.2. Synchronous Grammars for SMT

\[
\begin{align*}
SBAR & \rightarrow WHNP_1 V P_2 / WHNP_1 V P_2 \\
VP & \rightarrow VBZ_1 NP_2 / NP_2 VBZ_1 \\
WHNP & \rightarrow \text{which} / \text{der} \\
VBZ & \rightarrow \text{is} / \text{ist} \\
NP & \rightarrow \text{the solution} / \text{die Lösung}
\end{align*}
\]

Figure 5.2: An SCFG rule set for secondary clause verb reordering between English to German based on abstract linguistic structure.

\[
\begin{align*}
X & \rightarrow \text{which} \text{ is } X_1 / \text{der } X_1 \text{ ist} \\
X & \rightarrow \text{the solution} / \text{die Lösung}
\end{align*}
\]

Figure 5.3: An SCFG rule set for secondary clause verb reordering between English to German based on lexical context.

\[
\begin{align*}
SBAR & \rightarrow \text{which is } NP_1 / \text{der } NP_1 \text{ ist} \\
NP & \rightarrow \text{the solution} / \text{die Lösung}
\end{align*}
\]

Figure 5.4: An SCFG hybrid rule set for secondary clause verb reordering between English to German, combining lexical context with linguistic constituency information.
grammar belonging to it from the available training data and determining the extent to which the grammars induced actually lead to strong translations during decoding.

In the end, a strong synchronous grammar design is the one which pairs well with the learning approach that we employ to learn from data and the decoding schemes in which we embed our grammars to translate. Evaluating the strengths and weaknesses of a grammar design should only be performed within the context of a specific MT system implementation, or even a particular language pair or training and test data domain. The synchronous grammar formalisms we employ here, when trained with plain MLE lead to degenerate models that translate extremely poorly yet unseen source sentences, as we discussed in the wider context of Fragment Models in section 3.1.5. The exact same synchronous grammar designs provide state-of-the-art results when trained with CV-MLE as we show later in this chapter.

5.2.2 SCFG Modelling & Its Pitfalls

Probabilistic SCFGs extend the synchronous grammar rules to arrive at a stochastic joint model over string pairs. A probability value is attached to every grammar rule, so that these probabilities sum up to one for all rules having the same left-hand side. The key assumption behind this SCFG model is, similarly to the case of monolingual CFGs, that each non-terminal pair rewrite operation is independent of the rest of the derivation of the string-pair, given this non-terminal pair that we currently expand. The probability of a derivation \( D \) of a string-pair \( \langle e, f \rangle \) is then the product of the probabilities of all rules \( r \) used in \( D \), and the probability of the string-pair itself is the sum of the probabilities of all derivations \( D \Rightarrow \langle e, f \rangle \) leading to it.

\[
p(D) = \prod_{r \in D} p(r) \\
p(e, f) = \sum_{D \Rightarrow \langle e, f \rangle} p(D)
\]

This basic independence assumption behind SCFG models generalises the concept of a constituent from monolingual CFGs to their bilingual version. Right-hand side expansions covered by the same left-hand side non-terminal pair can be considered interchangeable, with the rule probabilities indicating how probable it is that they can be applied to rewrite this left-hand side non-terminal pair. For every right-hand side taking part in a synchronous rule, it is solely the left-hand side of the rule that will determine how the bilingual span covered by it will combine with the higher levels of the derivation. Accordingly, further expansions of the still abstract parts of the right-hand side are conditioned only on the non-terminal pairs that still need to be rewritten.
5.2. Synchronous Grammars for SMT

\[
\begin{align*}
NP & \rightarrow JJ^{1} NN^{2} / NN^{2} JJ^{1} & r_1 : p(r_1) \\
NP & \rightarrow JJ^{1} NN^{2} / JJ^{1} NN^{2} & r_2 : p(r_2) \\
NN & \rightarrow box / boîte & r_3 : p(r_3) \\
JJ & \rightarrow blue / bleue & r_4 : p(r_4) \\
JJ & \rightarrow beautiful / belle & r_5 : p(r_5)
\end{align*}
\]

Figure 5.5: An SCFG grammar rule-set categorising both word-pairs ‘blue / bleue’ and ‘beautiful / belle’ under the same non-terminal JJ, failing to take into account the different reordering patterns that these participate in. For \( p(r_1) > p(r_2) \), the model will prefer to translate the input ‘beautiful box’ wrongly as ‘boîte belle’.

Crucially, this leaves one of the most important components of the synchronous rule unaccounted for, when SCFG rules are combined to form a derivation. The reordering pattern between the non-terminals of the right-hand side is not an explicit part of the conditioning context in SCFG models, which is limited to the identity of the non-terminal pairs that function as left-hand sides. This may constitute a modelling pitfall that has received surprisingly little attention in the syntax-based MT community.

The concept of a constituent in monolingual CFGs describes strings of terminals and non-terminals which can substitute for each other as alternative expansions of the same covering left-hand side. This implied interchangeability in the monolingual case is justified by the ability of expansions covered by the same non-terminal to combine with similar surrounding contexts. For example, in English, noun phrases can combine to the left or to the right with verb clusters as subjects or objects of a sentence respectively, and nouns occur frequently after determiners or close to adjectives.

Importantly, when we move from monolingual to bilingual (or multi-lingual) grammars, the concept of a synchronous constituent and that of substitution must move further than taking into account the surrounding context in the two languages being modelled, for each of the two parts of the right-hand side expansions. An SCFG non-terminal pair must cover not only bilingual constituents whose two parts combine together similarly within each of the two languages of the language-pair, but they must also take part in similar reordering patterns.

A simple example illustrating this for translation between English and French can be seen in Figure 5.5. The word-pairs ‘blue / bleue’ and ‘beautiful / belle’ are both assigned the same non-terminal category JJ. This decision can be based on the observation that ‘blue’ and ‘beautiful’ can frequently substitute syntactically each other in English sentences and similarly ‘bleue’ and ‘belle’ do
so in French sentences. However, this fails to take under account that while the monolingual parts of the two bilingual constituents behave similarly in each of the two languages, when joined as word-pairs they combine quite differently in regard to how they reorder in sentence-pairs. The result is that according to the SCFG model, translations of adjective-noun English phrases will always be swapped if the first more common reordering pattern in rule $r_1$ is correctly assigned a larger probability than the more infrequent $r_2$, even when encountering exceptions such as ‘beautiful / belle’.

As we see next, in practice SCFG-based models of translation are complemented in state-of-the-art syntax-based SMT systems such as (Chiang, 2005a) with an array of additional features including a target language model, which can counter to a certain extent this modelling weakness. However, these systems make limited use of the abstract recursive structure offered by SCFGs, based mostly on reordering based on lexical context. We believe that, when learning SCFG grammars which rely on a syntactical bilingual analysis of the sentence-pairs which investigates the structural aspects of the translation process, it is important to take the issues highlighted in this section into consideration. Later in this chapter we do so, by evaluating a grammar design which uses non-terminals which relate to the reordering behaviour of the string-pairs that they cover, propagating reordering decisions across the synchronous derivation.

### 5.2.3 The Hiero Baseline

The Hiero SMT system (Chiang, 2007) significantly popularised syntax-based MT and remains the yardstick that most other syntax-based models and implementations compare to. Hiero, which we introduce in detail in section 2.4.2, employs an SCFG as the backbone of a log-linear conditional translation model. The SCFG score is combined together with multiple other features $\phi$, using weights $\lambda$ to evaluate the quality of SCFG derivations $D$ employing rules $r \in D$ and leading to translations $e$ for input $f$.

$$p(D \Rightarrow \langle e, f \rangle) \propto \phi_{LM}^{\lambda_M}(e) \times \prod_{r \in D} \prod_{i \neq LM} \phi_i^{\lambda_i}(r) \quad (5.3)$$

The SCFG grammar that Hiero employs treats translation as a hierarchical process, similar to the example of Figure 5.3. Namely, it focuses on lexicalised recursive translation rules, each of which translates a discontiguous source phrase-pair with ‘gaps’, while at the same time indicating the reordering pattern between the gaps on each side. However, as it covers all such discontiguous phrase-pairs under a single non-terminal $X$, the grammar employed by Hiero does not offer an abstract recursive explanation of the translation process and remains itself indiscriminate against the strings that each gap will be filled with.

An example of such rules can be seen in Figure 5.6. These rules are in practice allowed to recursively build sentence-segments up to a certain cut-off length (usu-
5.2. Synchronous Grammars for SMT

\[ X \rightarrow \text{do not } X \mathbf{1} / \text{ne } X \mathbf{1} \text{ pas} \]
\[ X \rightarrow \text{financial } X \mathbf{1} / X \mathbf{1} \text{ éconомiques} \]
\[ X \rightarrow \text{this } X \mathbf{1} X \mathbf{2} / \text{cette } X \mathbf{1} \text{ de } X \mathbf{2} \]
\[ X \rightarrow X \mathbf{1} \text{'s common } X \mathbf{2} \text{ policy} / \]
\[ \text{politique } X \mathbf{2} \text{ commune de } X \mathbf{1} \]

Figure 5.6: Hiero SCFG rules for English and French.

\[ S \rightarrow S \mathbf{1} X \mathbf{2} / S \mathbf{1} X \mathbf{2} \]
\[ S \rightarrow X \mathbf{1} / X \mathbf{1} \]

Figure 5.7: Hiero SCFG glue rules.

ally 10), which are later combined monotonically using the glue rules of Figure 5.7. This constraint together with the complementing features \( \phi \) in the model of equation (5.3), and most importantly the target language model feature \( \phi_{LM}(e) \), aid in avoiding errors due to the absence of a less ambiguous recursive structure than that offered by the rules of Figure 5.6.

The list of features \( \phi \) includes lexical translation scores judging translation on the word level, as well as scores considering the number of words in the target language output and the number of discontiguous phrase-pairs used in the SCFG derivation. Nevertheless, the core modelling elements related to the hierarchical phrase-based interpretation of translation assumed by the model are those employing conditional \textit{discontiguous} phrase translation probabilities. These extend the similar concept of features based on conditional phrase translation probabilities, from the PBSMT models which employ contiguous phrases, to the Hiero models which use phrase-pairs with gaps.

Crucially, these discontiguous phrase translation probabilities for the rules like those in (5.6) are estimated with a heuristic rule of thumb, similarly to how the probabilities for the contiguous phrase-pairs in PBSMT models are set. Namely, they are set based on the extraction counts of contiguous phrase translation patterns from a training word-aligned parallel corpus. These extraction counts are distributed evenly across all discontiguous phrase translation patterns that can be formed by substituting aligned subphrase-pairs of a contiguous phrase-pair for
Chapter 5. Learning Stochastic Synchronous Grammars

the non-terminal $X$. The rule weights are computed after normalising these assigned extraction counts for each target part of each rule right-hand side. These scores do provide relatively strong translation performance for some language pairs. However, like their PBSMT analogues, their relation in statistical terms with the training corpus remains obscure.

Even though the impact of depending on surface extraction counts might be limited when computing such estimates for the largely lexically-grounded rules of Hiero, this approach can hardly extend to the estimation of more involved grammars including notions of abstract recursive translation structure. In that case, heuristic estimation would demand counting extraction events on the unobserved latent part of the translation process. This would seem exceedingly arbitrary as the assumed latent structure abstracts more from the observed lexical surface, as we already discussed in section 2.5. While the heuristic estimation of the key parameters of the Hiero translation system might have offered a solid starting point for the emergence of hierarchical translation, it may be a bottleneck in the process of extending syntax-based systems towards grammars abstracting more from the lexical surface.

Zollmann and Venugopal (2006) move in this direction by extending the Hiero system through the introduction of target-side linguistic information in the grammar design along the lines of Figure 5.4. Nevertheless, they also offer no advancements on the learning aspects of the problem, applying instead the same heuristic estimation regime as Hiero, while supporting the simple heuristic estimates with an array of further additional features.

Overall, Hiero introduces the employment of an SCFG as the backbone of a hierarchical translation system focusing on translating with discontiguous phrase-pairs with gaps. However, the SCFG’s main contribution in Hiero implementations is to provide hierarchical derivations of target translations which are then scored by a feature-based model, while the ability of stochastic synchronous grammars to function as probabilistic models of translation as in equations (5.1) and (5.2) is not explored. In the rest of this chapter we follow this direction, and consider the learning of simple stochastic SCFGs as joint translation models. This features as a crucial intermediate step before moving on to induce the much more intricate linguistically motivated grammars of Chapter 6.

5.2.4 The Learning Problem

The Hiero system exemplifies the gains to be had by combining phrase-based translation (Och and Ney, 2004) with the hierarchical reordering capabilities of SCFGs, particularly originating from the binary Inversion Transduction Grammars (ITG) (Wu, 1997). Yet, the bulk of existing empirical work is largely based on the aforementioned heuristic techniques, and the learning of SCFGs remains an unsolved problem.
The difficulty of this problem stems from the need for simultaneous learning of many kinds of preferences under a single stochastic component.

- The translation of the lexical surface, either as part of dedicated phrase-emitting rules as those employed by our grammars later in this chapter or as part of lexicalised reordering rules as in the Hiero SCFGs.
- The reordering of phrase spans between the two languages.
- The overall translation structure as a mapping between the structure of both sides of the language pair.

The phrase-based analysis of the lexical correspondences between the source and target languages and the modelling of the reordering process have already been addressed in PBSMT models and the related translation systems with relative success, even if it was done only in relation to contiguous string elements. This leaves the modelling of the translation structure as the new exciting element in syntax-based MT. While the concept of a translation structure also includes the reordering patterns between segments of translated sentence-pairs, it moves further than this. It also considers the mapping between abstract syntactic elements of the source and target languages that can aid in explaining the translation process, and which do not necessarily coincide with the syntactic elements of monolingual linguistic analyses.

Crucially, it is exactly this novel aspect of syntax-based MT that learning through rule-of-thumb surface heuristics cannot support, due to the latent nature of translation structure. The approach of Chiang (2005a) however continued to base estimation of model parameters on extraction heuristics, mitigating the related issues by relying on the lexical, observable part of SCFGs, shunning at that time a richer syntactic analysis of the translation process while noting its importance as a future development.

Some efforts to learn a synchronous grammar for SMT concentrate on a part of the three translation preferences listed above. The problem of learning the hierarchical, synchronous grammar reordering rules is oftentimes addressed as a learning problem in its own right assuming all the rest is given (Blunsom et al., 2008b). A small number of efforts has been dedicated to the simultaneous learning of the probabilities of phrase translation pairs as well as hierarchical re-ordering, e.g., (DeNero et al., 2008; Zhang et al., 2008a; Blunsom et al., 2009). Of these, some concentrate on evaluating word-alignment, either directly such as (Zhang et al., 2008a), or indirectly by evaluating a heuristically trained hierarchical translation system from sampled phrasal alignments (Blunsom et al., 2009). However, very few evaluate on actual translation performance of induced synchronous grammars (DeNero et al., 2008). In the majority of cases, the Hiero system, which usually provides the baseline against which hierarchical systems are measured, remains superior in translation performance, see e.g. (DeNero et al., 2008).
5.3 Synchronous Grammar Learning

In the rest of the chapter, we tackle the problem of learning *generative phrase-based ITG models* as translation models assuming latent phrase segmentation and latent reordering; this setting is most similar to the training of Hiero. Unlike all other work that heuristically selects a subset of phrase-pairs, we start out from an SCFG that works with *all* phrase-pairs in the training set and concentrate on the aspects of learning. This problem is fraught with the risks of overfitting and can easily result in inadequate reordering preferences (DeNero et al., 2006).

We find that the translation performance of all-phrase probabilistic SCFGs induced in this setting crucially depends on the interplay between two aspects of learning:

- Defining a more constrained parameter space, where the reordering productions are phrase-lexicalised and made sensitive to neighbouring reorderings.
- Defining an objective function that effectively smoothes the maximum-likelihood criterion.

One of our contributions is in deploying the Cross-Validated EM algorithm implementing an effective, data-driven smoothed Maximum-Likelihood, which can cope with a model working with *all* phrase-pair SCFGs, building upon the work presented in Chapter 4. However, on top of the challenges already discussed there in the context of the application of CV-EM on PBSMTs, learning SCFGs poses significant novel challenges, the core of which lies in the hierarchical nature of a stochastic SCFG translation model and the relevant additional layer of latent structure. We address these issues in this chapter. Another important contribution is in defining a lexicalised reordering component within ITG that captures order divergences orthogonal to those tracked by the Hiero SCFG, but somewhat akin to PBSMT ‘monotone-swap-discontinuous’ reordering models (Tillman, 2004). Our best system exhibits Hiero-level performance on French-English Europarl data using an SCFG-based decoder. Our findings should be insightful for others attempting to make the leap from shallow phrase-based systems to hierarchical SCFG-based translation models that use learning methods, as opposed to heuristics.

5.3.1 Fragment Modelling Aspects

The Synchronous Context-Free Grammars which we consider here, both in the case of the Hiero baseline as well as for our own grammar designs presented in the next section, are phrase-based SCFGs. Unlike the Inversion Transduction Grammar as it was originally introduced as a word-based model in (Wu, 1997), these grammars allow synchronous rules with right-hand sides which include a
lexicalised part of arbitrary length. Such rules can describe contiguous or discontiguous aligned sentence-pair fragments of the training word-aligned parallel corpus. Under the assumption that we do not impose any arbitrary constraints on the length of these lexicalised rule segments, such phrase-based SCFGs can then be categorised under the Fragment Model family. In this context, the abstract part of the SCFG together with the rule probabilities provide the necessary stochastic generative machinery combining the lexical (dis)contiguous fragments to form sentence-pairs.

As in the case of PBSMT, this powerful modelling feature exposes the learning of these grammars under a Maximum Likelihood objective to the same overfitting issues as other all-fragment models such as Phrase-Based SMT (Marcu and Wong, 2002; DeNero et al., 2006) and Data-Oriented Parsing (Bod et al., 2003; Zollmann and Sima’an, 2006). Maximum Likelihood Estimation (MLE) returns degenerate grammar estimates that memorise well the parallel training corpus but generalise poorly to unseen data. As for the other fragment models, also in the case of SCFGs, this overfitting tendency leads towards an MLE estimate which effectively memorises whole sentence-pairs, using merely a trivial abstract structure leading directly towards the emission of whole training sentence-pairs, like for example \( S \rightarrow X \rightarrow \langle e, f \rangle \).

Such degenerate MLE estimates essentially memorise the empirical frequency of sentence-pairs in the parallel corpus but generalise extremely poorly as they also predict nothing more past what is included in the training data, as explained in section 3.1.5. The failure of straightforward applications of MLE to arrive at estimates which generalise well can be also attributed to a trade-off effect on the bias-variance decomposition of the expected Generalisation Error. The zero GE due to estimator bias is counter-balanced by a very high GE due to estimate variance, as we discuss for Fragment Models in general in section 3.1.6.

Independently of the aspect that it is being considered, the overfitting tendency of MLE estimators is encumbering the learning of SCFGs in all the aspects of the translation process that they are modelling, posing further challenges than those encountered while learning PBSMT phrase-table parameters in Chapter 4. On one hand, similarly to the estimation of PBSMT models, it does not allow us to identify and shift probability mass towards reusable lexical fragments. In the case of SCFGs however, this is complemented by the inability to learn any non-trivial translation structure, as the MLE solution overfits towards the minimal syntactic elements necessary to construct sentences from the largest memorised bilingual fragments. In order for the learning of SCFGs under a likelihood optimisation objective to arrive at any meaningful results, both in terms of reusable lexical components as well as abstract syntactic constructions, this strong tendency of the MLE estimator to memorise the training parallel corpus must first be addressed.
INPUT: Word-aligned parallel training data \(X\)  
Grammar extractor \(G\)  
The number of parts \(J\) to partition \(X\)  

OUTPUT: SCFG \(G\) with estimates \(\hat{\theta}^{CV} = \{p(r)\}\) for all grammar rules \(r\)

Partition training data in \(J\) equal parts \(X^1, \ldots, X^J\)

For \(1 \leq j \leq J\) do

Extract grammar rules set \(G_j = G(X^j)\)

Initialise \(G = \bigcup_j G_j, \; \hat{\theta}^{CV}_0 = \{p_0(r) : r \in G\}\) uniform per rule LHS

Let \(r = 0\)  // EM iteration counter

Repeat

Let \(r = r + 1\)

E-step:

For \(1 \leq j \leq J\) do

Calculate expected counts given \(G, \hat{\theta}^{CV}_{r-1}\),

for derivations \(D^{-j}\) of \(X^j\)

using rules from \(\bigcup_{k \neq j} G_k\)

M-step: set \(\hat{\theta}^{CV}_r\) to ML estimate given expected counts

Until \(\hat{\theta}^{CV}_r\) has converged

Figure 5.8: The CV Expectation-Maximization algorithm for SCFG learning.

5.3.2 CV-EM SCFG Estimation

In order to avoid the overfitting solution of plain MLE, we opt instead for a Cross-Validated MLE learning objective, which we implement using the Cross-Validated EM algorithm presented in section 3.2. Here we use Cross-Validation to leverage the bias-variance trade-off for learning stochastic all-phrase SCFGs. Given an input all-phrase SCFG grammar with phrase-pairs extracted from the training data, we maximise training data likelihood subject to CV smoothing. Splitting the word-aligned parallel training data \(X\) in \(J\) roughly equally-sized parts \(X^1, \ldots, X^J\), for each data part \(X^j\) we consider only derivations \(D^{-j}\) which employ grammar rules extracted from the rest of the data \(X^{-j}\). An essential part then of the learning process involves choosing the grammar extractor \(G(X)\), a function from data to an all-phrase SCFG under a particular grammar design, which we discuss in section 5.4 below.

A summary of the CV-EM algorithm for the learning of SCFG joint translation models such as those of equations (5.1) and (5.2) can be seen in Figure 5.8. As in all applications of CV-EM, being an EM instance guarantees convergence and a non-decreasing CV-smoothed training data likelihood after each iteration. Our practical implementation is based on a synchronous version of the Inside-Outside algorithm. This takes care during the E-step of the efficient computation
of expected counts of rule applications in derivations according to the current parameter set and is a straightforward adaptation of the monolingual version, considering bilingual instead of monolingual spans. The running time is $O(n^6)$, where $n$ is the input’s length, but by considering only derivation spans which do not cross word-alignment points, our implementation runs in reasonable times for relatively large corpora.

Beside being an estimator of the SCFG probability parameter set $\hat{\theta}$, the CV-MLE learning algorithm has the added value of being a grammar learner focusing on reducing generalisation error, in the sense that probabilities of grammar productions should reflect the frequency with which these productions are expected to be used for translating future data. Since the CV criterion prohibits for every data point derivations that use rules that can only be extracted from the same data part, such rules are assigned zero probabilities in the final estimate and are effectively excluded from the grammar. In this way, the algorithm ‘shapes’ the input grammar, concentrating probability mass on productions that are likely to be used with future data.

In this chapter we do not pursue the use of grammar extractors outputting complex abstract structures, even though we do move further than a plain Hierolike grammar totally lacking this aspect. For this reason, the effect of restricting the grammar mentioned above relates more to the lexical part of the grammar designs that we experiment with. Nevertheless, concentrating on lexical units which are expected to generalise by applying CV-smoothing on the lexical level is crucial in allowing us to estimate the parameters related to the higher-level syntactical components, as well as to learn how to combine these reusable lexical building blocks together.

The number of abstract syntactic rules used in the grammars that we present below is limited and their design is a generic one without any reference to the training data. This allows us to consider these as included in every extracted rule-set $G(\mathcal{X}^j)$ and allow them to survive the CV-smoothing in their entirety. However, as we increase the complexity of the higher-level syntax in the synchronous grammars that we consider and especially if this part of the grammar is constructed in reference to the training data, we believe it is important to also address the possible overfitting of the abstract part of the grammar. We consider this issue in Chapter 6.

5.3.3 Smoothing the Model

The practical application of CV-EM for SCFGs also demands the treatment of boundary cases. There will often be sentence-pairs in $\mathcal{X}^j$, that cannot be fully derived by the grammar extracted from the rest of the data $\mathcal{X}^{\neg j}$. The reason might be: (a) ‘unknown’ words (i.e. not appearing in other parts of the CV partition) or (b) complicated combinations of adjacent word-alignments. To address this, we employ external smoothing of the grammar, prior to learning.
Our solution is to extend the SCFG extracted from $X^{-j}$ with new emission productions deriving the ‘unknown’ phrase-pairs (i.e., found in $X^j$ but not in $X^{-j}$). Crucially, the probabilities of these productions are drawn from a fixed smoothing distribution, i.e. they remain constant throughout estimation. Our smoothing distribution of phrase-pairs $⟨\tilde{e}, \tilde{f}⟩$ for all pre-terminals considers source-target phrase lengths drawn from a Poisson distribution with unit mean, drawing subsequently the words of each of the phrases uniformly from the vocabulary of each language, similar to (Blunsom et al., 2009).

$$p_{\text{smooth}}(⟨\tilde{e}, \tilde{f}⟩) = \frac{p_{\text{poisson}}(|\tilde{f}|; 1) \cdot p_{\text{poisson}}(|\tilde{e}|; 1)}{V_{\tilde{f}} V_{\tilde{e}}}$$  \hspace{1cm} (5.4)$$

Since the smoothing distribution puts stronger preference on shorter phrase-pairs and avoids competing with the ‘known’ phrase-pairs, it leads the learner to prefer using as little as possible such smoothing rules, covering only the phrase-pairs required to complete full derivations.

5.4 Parameter Spaces and Grammar Extractors

As we discussed in 5.2.1, the translation structure is a latent modelling component and an MT practitioner is free to consider it from different perspectives, which may be based on machine learning, linguistic or cognitive grounds. However in the end the synchronous structure is primarily judged empirically, based on the ability to more closely capture the translation process and lead us towards better translations. In the context of the learning framework presented in the previous section, a crucial modelling choice is then establishing the space of latent synchronous grammatical constructions that our learner will consider against the empirical observations in the training data.

In our SCFG learning pipeline, the decisions related to the synchronous grammar design are encoded in the Grammar Extractor (GE). A GE is a function from a word-aligned parallel corpus to a set of Synchronous Context-Free Grammar rules. Together with the constraints that render a proper joint probabilistic SCFG, i.e. the sum of probabilities for productions that have the same left-hand side must be one, the GE also serves to define the parameter space of the stochastic model that we establish by extending every rule in the output of the GE with a probability.

The Grammar Extractors used in this chapter create SCFGs productions of two different kinds:

1. Abstract hierarchical synchronous productions that define the space of possible derivations up to the level of SCFG pre-terminals
2. The phrase-pair emission rules that expand the pre-terminals to phrase-pairs of varying lengths.
5.4. Parameter Spaces and Grammar Extractors

Start $S \rightarrow X / X$

Monotone Expansion $X \rightarrow X \begin{bmatrix} 1 \\ 2 \end{bmatrix} / X \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Switching Expansion $X \rightarrow X \begin{bmatrix} 1 \\ 2 \end{bmatrix} / X \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

Phrase-Pair Emission $X \rightarrow \hat{e} / \hat{f}$

Figure 5.9: Single Phrase-Pair NT Grammar.

Computing the GE’s output begins by extracting phrase-pair emitting rules for the set of all translational equivalents (without length upper-bound) abiding to the word-alignment, according to the rules of (Och and Ney, 2004; Koehn et al., 2003). These phrase-pair emitting rules are complemented by the abstract translation structure rules that cover the distance between the start symbol and the phrase-pairs. However, while the phrase-pairs that will be the right-hand sides of the phrase-pair emission rules depend on the parallel corpus, we cannot extract translation structure rules from it as the latter is not labelled with a synchronous parse. For this, the translation structure part of the grammar output of the GEs that we examine in this chapter, does not depend on their input. Since these rules will be present in the SCFGs extracted from all cross-validation parts, the CV-EM learning algorithm implementation of Figure 5.8 cannot protect against overfitting caused by this part of the grammars. For this reason, for this first examination of SCFG learning with CV-EM discussed in this chapter, we have elected to employ relatively simple translation structures, to mitigate the risk of overfitting due to over-specialised abstract structure rules.

Below we present the two grammar extractors employed in our experiments. We start out from the most generic, ITG-like formulation, and aim at incremental refinement of the hierarchical productions in order to capture relevant, content-based phrase-pair reordering preferences in the training data.

5.4.1 Single Phrase-Pair NT SCFG

This is a phrase-based binary SCFG grammar employing a single non-terminal $X$ covering each extracted phrase-pair. The other two productions consist of monotone and switching expansions of phrase-pair spans covered by $X$. Finally, the whole sentence-pair is considered to be covered by $X$. We will call this the ‘plain SCFG’ extractor and the simple abstract translation structure that it produces serves as a baseline against which more elaborate grammar designs can be empirically compared. The SCFG produced by the plain SCFG extractor, given an input corpus $\mathcal{X}$ from which phrase-pairs $\langle \hat{e}, \hat{f} \rangle$ can be extracted, is listed in Figure 5.9.
Chapter 5. Learning Stochastic Synchronous Grammars

Start $S \rightarrow X_L/X_R$

Monotone Expansion

$X \rightarrow X_1 X_2 / X_1 X_2$

$X^L \rightarrow X_1 X_2 / X_1 X_2$

$X^R \rightarrow X_1 X_2 / X_1 X_2$

Switching Expansion

$X \rightarrow X^L_1 X^R_2 / X^R_2 X^L_1$

$X^L \rightarrow X^L_1 X^R_2 / X^R_2 X^L_1$

$X^R \rightarrow X^L_1 X^R_2 / X^R_2 X^L_1$

Phrase-Pair Emission

$X \rightarrow \tilde{e} / \tilde{f}$

$X^L \rightarrow \tilde{e} / \tilde{f}$

$X^R \rightarrow \tilde{e} / \tilde{f}$

Figure 5.10: Lexicalised-Reordering SCFG

5.4.2 Lexicalised Reordering SCFG

One weakness of the ‘plain SCFG’ is that the reordering decisions in the derivations are made without reference to lexical content of the phrases; this is because all phrase-pairs are covered by the same non-terminal. As a refinement, we propose a grammar extractor that aims at modelling the reordering behaviour of phrase-pairs by taking their content into account. This time, the $X$ non-terminal is reserved for phrase-pairs and spans which will take part in monotonic productions only. Two fresh non-terminals, $X^L$ and $X^R$, are used for covering phrase-pairs that participate in order switching reordering operations with other, adjacent phrase-pairs. The non-terminal $X^L$ covers phrase-pairs which appear first in the source language order, and the latter those which follow them. The grammar rules produced by this GE, dubbed ‘switch grammar’, are listed in Figure 5.10.

The reordering information captured by the switch grammar is in a sense orthogonal to that of Hiero-like systems utilising rules such as those listed in Figure 5.6. Hiero rules encode hierarchical reordering patterns based on surrounding context. In contrast, the switch grammar models the reordering preferences of the phrase-pairs themselves, similarly to the monotone-swap-discontinuous reordering models of Phrase-based SMT models (Tillman, 2004). On top of that, it strives to match pairs of such preferences, combining together phrase-pairs with compatible reordering preferences, as well as conditioning the production of every
which der is X L X R

Figure 5.11: A sub-tree covering a secondary clause between English and German using the switch SCFG. $< >$ indicates a switch reordering operation between the two children of the non-terminal. The application of the rule $X \rightarrow X^L X^R$ indicates that the two children (verb and noun phrase) must switch as we translate between the two languages, while the resulting verb phrase combines monotonically with the context on its left.
non-terminal to the reordering behaviour of the span covered by it. In this way, it addresses the modelling pitfalls described in section 5.2.2. Now the reordering choices of every synchronous derivation expansion are affected both by the preferences of the children as well as the parents of every node in the derivation tree, as encoded by the three specialised non-terminals present in the left and right-hand side of every production rule.

An example of a derivation subtree for a secondary clause between English and German can be seen in Figure 5.11. For the switch SCFG that we employ in this chapter, while the form of the abstract structure in the example can be explained in linguistic terms, identifying it past the pre-terminals makes use of a small set of generic rules and their probabilities, which together represent the overall reordering behaviour of synchronous spans across the training corpus. In Chapter 6 we will enrich the synchronous grammatical constructions to explicitly condition such reordering operations on linguistic cues.

5.5 Experiments

Pairing each Grammar Extractor with the CV-EM implementation of section 5.3.2 allows us to learn probabilistic Synchronous Context-Free Grammars and estimate their parameters from training word-aligned parallel corpora. In this section we proceed to integrate these synchronous grammars within an SCFG-based decoder. We subsequently evaluate our performance in relation to the state-of-the-art Hiero baseline of section 5.2.3 on a French to English translation task.

5.5.1 Decoding

The joint model of bilingual string derivations provided by the learnt SCFG grammar can be used for translation given a input source sentence, since:

$$
\arg\max_{e} p(e|f) = \arg\max_{e} p(e, f)
$$

We use our learnt stochastic SCFG grammar with the decoding component of the Joshua SCFG toolkit (Li et al., 2009). The full translation model interpolates log-linearly the probability of a grammar derivation together with the language model probability of the target string. The model is further smoothed, similarly to phrase-based models and the Hiero system, with smoothing features $\phi$ such as the lexical translation scores of the phrase-pairs involved and rule usage penalties in the same way as our baseline. As usual with statistical translation, we aim for retrieving the target sentence $e$ corresponding to the most probable derivation $D \Rightarrow (e, f)$ for the source side $f$ making use of rules $r$, with:
\[ p(D \Rightarrow (e, f)) \propto \phi_{LM}^\lambda(e) p_{SCFG}(e, f)^\lambda_{SCFG} \times \prod_{r \in D} \prod_{i \neq LM, SCFG} \phi_i^\lambda(r) \]  

The interpolation weights are tuned using Minimum Error Rate Training (Och, 2003).

### 5.5.2 Results

We test empirically the learner’s output grammars for translating from French to English, using \( J = 5 \) for the Cross-Validation data partitioning. The training material is a GIZA++ word-aligned corpus of 200K sentence-pairs from the Europarl corpus (Koehn, 2005), with our development and test parallel corpora of 2K sentence-pairs stemming from the same source. Training the grammar parameters until convergence demands around 6 hours on an 8-core 2.26 GHz Intel Xeon system. Decoding employs a 4-gram language model, trained on English Europarl data of 19.5M words smoothed using modified Kneser-Ney discounting (Chen and Goodman, 1998), and lexical translation smoothing features based on the GIZA++ alignments.

In a sense, from a learning perspective the real baseline that we might compare against should be a system employing the plain MLE estimate for the grammar extracted from the whole training corpus. However, as we have already discussed, this assigns zero probability to all sentence-pairs outside of the training data and is subsequently bound to perform extremely poorly, as decoding would then completely rely on the smoothing features. In addition, we cannot directly compare the CV-EM estimates for the plain and switch SCFGs against estimates that are heuristically trained similarly to those employed in Hiero or PBSMT, as it is not clear how the reordering rule probabilities of a grammar similar to the ones we use could be trained heuristically based on extraction counts, given that the relevant structure is unobserved.

Instead, we opt to compare against a hierarchical translation baseline provided by the Joshua toolkit, trained and tuned on the same data as our learning algorithm. The grammar used by the baseline is much richer than the ones learnt by our algorithm, also employing rules which translate with context, as discussed in section 5.2.3. However, the baseline does not make use of abstract translation rules without a lexical part, relying on the glue grammar of Figure 5.7 to monotonically combine the discontiguous phrase-pairs together after they have been recursively expanded. Relating the performance of our learnt stochastic SCFG grammars to a hierarchical translation baseline such as this, has the added advantage of comparing against a system which remains in the state-of-the-art of SCFG-based translation, evaluating the potential of our approach to deliver real-world competitive translation performance.
Table 5.1 presents the translation performance results of our systems and the baseline. On first observation, it is evident that our learning algorithm outputs stochastic SCFGs which manage to generalise, avoiding the degenerate behaviour of plain MLE training for these models. Given the notoriety of the estimation process, this is noteworthy on its own. Having a learning algorithm at hand which realises to a reasonable extent the potential of each stochastic grammar design (as implemented in the relevant grammar extractors), we can now compare between the two grammar extractors used in our experiments. The results table highlights the importance of conditioning the reordering process on lexical grounds. The plain grammar with the single phrase-pair non-terminal cannot accomplish this and achieves a lower BLEU score. On the other hand, the switch SCFG allows such conditioning. The learner takes advantage of this feature to output a grammar which performs better in taking reordering decisions, something that is reflected in both the actual translations as well as the BLEU score achieved.

Furthermore, our results highlight the importance of the additional smoothing decoding features of equation (5.5). The unsmoothed baseline system itself scores considerably less when employing solely the heuristic translation score. Our unsmoothed switch grammar decoding setup improves on the baseline by a considerable difference of 0.7 BLEU, highlighting the reliance of the heuristic estimates on these additional smoothing features to provide reasonable translations. Subsequently, when adding the smoothing lexical translation features, both systems record a significant increase in performance, reaching comparable levels of performance.

The degenerate behaviour of MLE for SCFGs can be greatly limited by constraining ourselves to grammars employing minimal phrase-pairs: phrase-pairs which cannot be further broken down into smaller ones according to the word-alignment. One could argue that it is enough to perform plain MLE with such minimal phrase-pair SCFGs, instead of using our more elaborate learning algorithm with phrase-pairs of all lengths. To investigate this, for our final experiment we used a plain MLE estimate of the switch grammar to translate, limiting
the grammar’s phrase-pair emission rules to only those which involve minimal phrase-pairs. The very low score of 17.82 BLEU (without lexical smoothing) not only highlights the performance gains of using longer phrase-pairs in hierarchical translation models, but most importantly provides a strong incentive to address the overfitting behaviour of MLE estimators for such models, instead of avoiding it.

5.6 Related Work

Most learning of phrase-based models, e.g. (Marcu and Wong, 2002; DeNero et al., 2006) and the work presented in Chapter 4, works without hierarchical components such as those employed by ITG/SCFG grammars. These learning problems pose other kinds of learning challenges than the ones presented by the explicit learning of SCFGs. While Chiang’s original work (Chiang, 2005a; Chiang, 2007) introduces a particular flavour of phrase-based binary synchronous grammars, his learning approach keeps almost intact the heuristic estimation of PBSMT. The learning problem is not expressed in terms of an explicit objective function and surface heuristic counts are used instead. Nevertheless, it has been very difficult to match the performance of Hiero-like models without use of these heuristic counts.

A somewhat related work, (Blunsom et al., 2008b), attempts learning new non-terminal labels for synchronous productions in order to improve translation. This work differs substantially from our work because it employs a heuristic estimate for the phrase pair probabilities, thereby concentrating on a different learning problem: that of refining the grammar symbols. Our approach might also benefit from such a refinement but we do not attempt this problem here. In contrast, (Blunsom et al., 2008a) works with the expanded phrase pair set of (Chiang, 2005a), formulating an exponential model and concentrating on marginalising out the latent segmentation variables. Again, the learning problem is rather different from ours. Similarly, the work in (Zhang et al., 2008a) reports on a multi-stage model, without a latent segmentation variable, but with a strong prior preferring sparse estimates embedded in a Variational Bayes (VB) estimator. This work concentrates the efforts on pruning both the space of phrase pairs and the space of (ITG) analyses.

To the best of our knowledge, the work presented in this chapter based on the results of (Mylonakis and Sima’an, 2010) was the first to attempt learning probabilistic phrase-based binary SCFGs as translation models, in a setting where both a phrase segmentation component and a hierarchical reordering component are assumed as latent variables. Like our approach, (DeNero et al., 2008) also employ an all-phrases model, however the work presented here complements the results of Chapter 4 in showing that it is possible to train such large-scale grammars under iterative algorithms like CV-EM, without need for sampling or pruning.
5.7 Discussion

Phrase-based stochastic SCFGs provide a rich formalism to express translation phenomena, which has been shown to offer competitive performance in practice. Since learning SCFGs for machine translation has proven notoriously difficult, most successful SCFG models for SMT rely on rules extracted from word-alignment patterns and heuristically computed rule scores, with the impact and the limitations imposed by these choices yet unknown.

Some of the reasons behind the challenges of SCFG learning can be traced back to the introduction of latent variables at different, competing levels: word and phrase-alignment used side by side with hierarchical reordering structure, with larger phrase-pairs reducing the need for extensive reordering structure and vice versa. While imposing priors such as the often used Dirichlet distribution or the Dirichlet Process provides a method to overcome these pitfalls, we believe that the data-driven CV-MLE learning objective and the CV-EM algorithm employed in this chapter provide an effective alternative to them, focusing more on the data instead of importing generic external human knowledge. Our use of CV-EM to learn Synchronous CFGs adds additional evidence to the effectiveness of our algorithm to train models assuming increasingly complex latent variables, moving from the flat segmentation variables of Chapter 4 to the recursive structures of SCFGs.

We believe that the work in this chapter makes a significant step towards learning synchronous grammars for SMT. This is an objective not only worthy because of promises of increased performance, but, most importantly, also because it increases the depth of our understanding of SCFGs as vehicles of latent translation structures. Our usage of the induced grammars directly for translation, instead of an intermediate task such as phrase-alignment, aims exactly at this.

While the latent structures that we explored here were relatively simple in comparison with Hiero-like SCFGs, they take a different, content-driven approach to learning reordering preferences, rather than the context-driven approach of Hiero. We believe that overall these approaches are not merely orthogonal, but could also prove complementary. Taking advantage of the possible synergies between content and context-driven reordering learning is an appealing direction of future research stemming from this thesis. This is particularly promising for other language pairs, such as Chinese to English, where Hiero-like grammars have been shown to perform particularly well.

In the following chapter, we build on the intuitions gained and the results presented above to learn a translation model employing a rich, linguistically motivated latent structure. This moves further than synchronous grammars which use a handful of abstract categories to describe the translation process, like those we employed here. Even though we proceed towards using grammars taking advantage of hundreds of thousands of abstract categories, we retain the design
principles behind the ‘switch SCFG’ presented here. We use its ability to facilitate learning how to combine together the reordering preferences of phrase-pairs and those of abstract categories, within a translation structure robust enough to cover whole sentence-pairs. In this way, the successful deployment of CV-EM as a learning algorithm for the somewhat simpler SCFGs presented in this chapter, as well as our experimentation with different synchronous grammar design principles, pave the way for the work that follows.