# CCG-based Models for Statistical Machine Translation

*Michael Auli*

PhD Proposal
School of Informatics
University of Edinburgh
2009

# Abstract

The arguably best performing statistical machine translation systems are based on context-free formalisms or weakly equivalent ones. These models usually use a synchronous version of a context-free grammar (SCFG) which we argue is too rigid for the highly ambiguous task of human language translation. This is exacerbated by the fact that the imperfect methods available for aligning parallel texts make extracting an efficient grammar very hard. As a result, the context-free grammars extracted are usually very large in size after having already been restricted through a variety of constraints.

We propose to use Combinatorial Categorial Grammar (CCG) for machine translation models. CCG is a lexicalized, mildly-context-sensitive formalism which is very well suited to capture long-distance dependencies that are not addressed very well by most current models. We believe that CCG is very well suited for the task of machine translation due to its ability to represent non-constituents in a syntactic way which frequently occur in parallel texts as well as its high derivational flexibility. This allows us to use some of the advantages of non-syntactic phrase-based approaches within a syntactic framework such as a relatively small grammar size compared to context-free-based machine translation grammars.

A number of models leveraging the advantages of CCG are possible, however, our principal goal is to develop a string-to-tree based model which projects CCG on the target side of a synchronous grammar. We intend to apply the vast progress made in monolingual CCG parsing to machine translation. Additionally, we propose to extend CCG to a synchronous grammar (SCCG) as it has been done for other related formalisms such as tree adjoining grammars. We hope that a SCCG may provide similar derivational flexibility to monolingual CCG which may result in a better model for translational equivalence.

# Table of Contents

# Chapter 1

# Introduction

Statistical Machine Translation (SMT) uses probabilistic methods to estimate models from parallel text which can be used to fully automatically translate human languages. The striking advantage of statistical approaches is their generic applicability to any pair of languages. This and the availability of the necessary software and data allows virtually everyone to create machine translation systems without any knowledge of the languages involved.

Until up to very recently, non-syntactic phrase-based translation models (Koehn et al., 2003) have been dominant. These finite-state-based models use a base-formalism which is very well understood and allows for easy integration of n-gram language models. The strength of phrase-based models is their ability to memorize entire phrases which makes translation of idioms possible. Recently, linguistically oriented models (Chiang, 2007) have demonstrated comparable or even better performance.

Syntax-based methods relying on more powerful grammar formalisms promise to model translation in a more natural way. Many attempts in developing syntax-based models have been made, ranging from using simple unlexicalized synchronous context-free grammars (Wu, 1997), string-to-tree transducers (Galley et al., 2004, 2006) up to dependency tree-to-tree translation (Quirk and Menezes, 2006) to name a few.

One of the problems associated with machine translation is the intractability for all but very simple models. Search is about finding a good translation according to our model. There is a trade-off in between the complexity of the model and the amount of search errors in finding the best translation one is willing to accept.

# Chapter 2

# Related Work Review

## 2.1 Syntax in Phrase-based Machine Translation

The basic unit of phrase-based translation are pairs of phrases which are extracted from word-aligned bilingual corpora. Experiments in restricting phrase-pairs to syntactic constituents demonstrated lower translation performance (Koehn et al., 2003) and warrants the claim that translation of non-constituent phrases seems to be critical for this model. The strengths of the phrase-based model are the translation of multiword units and the modelling of local reordering, however, long-distance reordering is a big challenge for the finite-state based approach as well as the utilized n-gram models.

The most successful ways in dealing with long-distance reordering are based on reordering source language sentences to assimilate their word-order to the target language in a preprocessing step before actual translation is carried out (Wang et al., 2007; Collins et al., 2005). However, these approaches lack generalization to other language-pairs since they employ hand-crafted rules for the preprocessing step.

Another issue is the translation into morphologically richer languages such as from English into Greek for which all statistical approaches generally perform poorly. Avramidis and Koehn (2008) deals with this problem via extracting gender and case information from source side parse trees to better inform the translation into a morphologically richer language.

Factored models (Koehn and Hoang, 2007) are a framework to integrate arbitrary syntactic information such as lemmas, number or case information into the source and target side of phrasal translation. Most performance gains have been reported when using language models over target-side syntactic annotation. In particular, experiments have shown better noun phrase agreement which can be captured by the target language models, however, improvements in subject verb agreement were marginal due to long-distance dependencies which are hard to capture with n-gram based language models.

## 2.2 Approaches to Syntax-based Machine Translation

Most current syntax-based approaches use parse trees to guide the translation process. These trees can occur on both the target and source side or on either side. We refer to the former as *tree-to-tree* models and the latter as either *string-to-tree* or *tree-to-string* models.

### 2.2.1 String-to-Tree Models

Galley et al. (2006) use a string-to-tree model which encodes syntactic knowledge of the target side to create more fluent translations. The employed grammar formalism (Galley et al., 2004) is a tree transducer which projects target language syntax onto the source language strings. The rules are extracted via cutting the target side trees into subtrees with respect to the word alignment. The productions are then binarized to yield context-free rules with no more than two nonterminal symbols (Zhang et al., 2006). This theoretically allows parsing in $O(n^3)$ where $n$ is the number of foreign words, however, binarization transforms a single production into a number of context-free productions via introducing a large number of new non-terminals which in turn adversely effect parsing complexity. An example production mapping a sub-tree to a string of (Galley et al., 2004) may look like the following

$$\text{NP(DET: } x_0 \text{ ADJ: } x_1 \text{ NN: } x_2) \longrightarrow x_0\ x_2\ x_1$$

which models an adjective-noun swap within a noun phrase. This rule can be converted into the following a synchronous context-free grammar (Zhang et al., 2006)

$$\begin{gathered} \text{NP} \longrightarrow T_{572} \,/\, T_{572} \\ T_{572} \longrightarrow DET_0\ ADJ_1\ NN_2 \,/\, DET_0\ ADJ_1\ NN_2 \end{gathered}$$

via introducing a new nonterminal. The binarization breaks the rule then up into

$$\begin{gathered} \text{NP} \longrightarrow T_{572} \,/\, T_{572} \\ T_{572} \longrightarrow DET_0\ V_0 \,/\, DET_0\ V_0 \\ V_0 \longrightarrow ADJ_1\ NN_2 \,/\, ADJ_1\ NN_2 \end{gathered}$$

Marcu et al. (2006) propose a phrase-based model which utilizes the rules from Galley et al. (2004). Non-syntactic constituents are handled via introducing pseudo non-terminals and pseudo productions. For example, this allows to deal with translations of only partially realized noun phrases such as “the blue” via assigning it a nonterminal (*NN*_NP) that marks that the noun is missing

$$\text{*NN*\_NP(DET(the) JJ(blue))} \rightarrow \text{das blaue}$$

In a sense, this method attempts to represent contextual information with a context-free formalism. Another disadvantage is that this approach also significantly increases the number of nonterminal symbols. The same can be much more naturally achieved with CCG as we will see below.

Another example is the tree transducer model of Yamada (2003) which assumes that an English parse tree is fed into a noisy channel which translates it into a foreign sentence. The model is based on applying stochastic operations to reorder the English parse tree, insert new words and to finally translate the English-leafs into the foreign language. One reason on why this approach did not show any performance increases may be the use of context-free grammars which are too rigid to model isomorphism at the production level (Fox, 2002) as we will discuss in the next section.

The approaches described so far use syntax in a linguistic sense, however, the first syntactic system to demonstrate better performance than a phrase-based system utilized a lexicalized SCFG without using any linguistic information (Chiang, 2007). Follow on work by Venugopal and Zollmann (2006) builds on this work via introducing linguistically motivated nonterminal labels obtained from a Penn Treebank (Marcus et al., 1993) trained parser. Unfortunately, doing so brings with it a number of problems: Firstly, similarily to Marcu et al. (2006), the model has to deal with the problem that initial phrase-pairs do not always follow linguistic boundaries as demonstrated by (Fox, 2002). The authors deal with this problem in similar ways as Marcu et al. (2006) via introducing composed non-terminals labels such as $C_1 + C_2 + C_3$ for a word sequence headed by three non-terminals or via categories similar to CCG complex categories.

Secondly, the integration of a n-gram language model has been shown to be possible for an induced grammar using only a single nonterminal symbol (Chiang, 2007) which can be parsed with acceptable complexity and search errors. However, doing so for a large set of non-terminals increases complexity dramatically which requires a very tight beam search in order to make the model tractable.

A proposed solution is to perform decoding in two passes: In a first coarse step, the hypergraph is constructed without using language model contexts, followed by a detailed exploration in which certain parts of the hypergraph are rescored using a full language model (Venugopal et al., 2007). A similar approach is proposed by Zhang and Gildea (2008) which uses more and more complex language models at each decoding step in order to keep complexity manageable.

### 2.2.2 Tree-to-String Models

Tree-to-string models use their syntactic knowledge of the source side to constrain rule selection. An example for such a model is Liu et al. (2006) which parses inputs and then performs stack decoding via combining translations of sub-tree nodes as opposed to Huang et al. (2006) who use a dynamic programming algorithm to search for the best translation. Subsequent work utilizes parse-forests rather than only the 1-best parse of the source side and presents methods to discount the weight of rules extracted from non-1-best parses (Mi and Huang, 2008).

Gimpel and Smith (2008) reports small performance increases when integrating source-side syntactic features into the log-linear framework of a standard phrase-based system. The authors observe the highest gains for languages for which high-accuracy annotation tools are available. Another way to integrate source side information is via using a word sense disambiguation (WSD) system which exploits the fact that depending on the sense of a word, a different translation may be more appropriate such as demonstrated by Chan et al. (2007).

Discriminative models (Blunsom et al., 2008; Blunsom and Osborne, 2008) can use a plethora of features conditioned on the entire source side tree to make decisions about which rules to choose. Using source side features rather than encoding syntax into the grammar can avoid problems such as rule-sparsity as well as too strict model constraints. However, the scalability issues of discriminative models restrict their usage to very simple sentences.

### 2.2.3 Tree-to-Tree models

Tree-to-Tree models use syntactic trees on both the source and target side in the hope of combining the advantages of string-to-tree and tree-to-string models. Notably, translational equivalence studies (Wellington et al., 2006) have found that syntactic constraints on both sides are sometimes too restrictive to model certain word-alignments.

Most approaches use single trees on both the target and source side of a production (Cowan et al., 2006; Riezler and Maxwell, 2006), however, there are also attempts to use multiple trees on either side (Zhang et al., 2008).

Quirk and Menezes (2006) proposes to use dependency trees on both the target and source side which allow to better capture semantically related constituents (e.g. verbs are surrounded by all their objects) compared to constituency focused phrase-based approaches. The model has been shown to perform well in limited data domains but the problem of ordering the translated treelets presents a large challenge for the target language ordering model. This is despite the finding that the largest cohesion between constituents exists when dependency structures are used (Fox, 2002). In subsequent work, automatically extracted order templates have been proposed in order to mitigate this problem (Menezes and Quirk, 2007).

### 2.2.4 Syntactic Language Models

Most language models used in speech recognition and statistical machine translation are finite state-based n-gram language models such as SRILM (Stolcke, 2002). Unfortunately, such models only have a limited context of a few words which disallows them to determine if a given translation is well formed. Several attempts have been made to overcome this issue via using probabilistic grammars (Charniak et al., 2003; Post and Gildea, 2008; Shen et al., 2008) which promise to give a sense of grammaticality based on the entire context rather then only a small word-window.

From a computational point of view, n-gram based language models are easier to intersect with the translation model, be it a finite-state-based or grammar-based model. However, if the language model uses, for example, a context-free grammar and the translation model as well, then one has to deal with intersecting two context-free grammars which is undecidable (Hopcroft and Ullman, 1979). Practically doing so requires severe pruning of the search space which is likely to result in many search errors. This may be one reason why previous attempts to use syntactic language models were mostly unsuccessful.

Nesson et al. (2006) suspect that the failure of context-free-based language models lies in their inability to state lexical dependencies which is traded in for their ability to substitute abstract categories. We believe that this is one of the reasons since work using context-free grammar-based language models (Charniak et al., 2003; Post and Gildea, 2008) has not shown improvements as opposed to dependency-grammar-based language models (Shen et al., 2008) which may be because dependency grammars are very well suited to stating lexical dependencies. Post and Gildea (2008) argue on a similar line, stating that parsers could perform better when their independence assumptions are removed. Another reason may be that the used parsers are trained on well-formed domain-specific data such as the Penn Treebank corpus as well as their inability to reliably score the quality of very noisy translation candidate-sentences.

*Incremental parsing* is a computational efficient approach used in the speech recognition community to make parsing more suitable for the task of language modelling. It is based on creating larger analysis based on previous structures in a left to right manner without changing the previous structure. Roark (2001) proposes a top-down left-corner parser which has been successfully used as a language model for speech recognition. Although, it has also been noted that finding a good parse this way is much harder (Xu et al., 2001) because the parser effectively has to commit itself to subderivations before it has seen the full sentence.

Hassan et al. (2008); Hassan (2008) present a linear-time incremental CCG parser which is used to discover predicate-argument dependencies in CCGbank. Their approach requires a transformation of CCGbank so that all derivations are left-to-right. While the method has the computational properties required for machine translation, it lacks qualitative performance. In their experiments, the F-Score on dependency relations drops from 87 to 59 when disallowing lookahead. Notably, these results are for well-formed in-domain test-data and also, lookahead is not available during translation. When tested in a machine translation setting, their approach fails to outperform the baseline in terms of BLEU.

Recent work (Post and Gildea, 2008) investigating context-free parsers as language models in machine translation found that standard parsing models such as the one of Collins (2003) require modification, in order to be useable in the noisy machine translation domain. It is argued that removing *independence assumptions* from parsing models may result in better performance. For example, context-free parsing models cannot capture the fact that the expansion of a noun phrase depends on if it is subject or object position Manning and Schütze (1999). We will discuss more powerful formalisms (Section 2.3.3) which are able to overcome these problems.

In summary, syntax-based language models are suffering under the severe pruning which is required to integrate them and their independence assumptions which disallows them to capture lexical relationships.

## 2.3 Syntactic Formalisms for Machine Translation

### 2.3.1 Synchronous Grammars

A synchronous formalism allows to model the generation of string-pairs simultaneously which can be used to capture the correspondence between parallel text. A synchronous grammar can both be used to assign syntactic structure to parallel text, or, in a translation setting, to parse a text in one language and to simultaneously generate text in another. It is very common to use synchronous grammars which is based on a projection of a monolingual grammar from one side to the other, effectively removing the burden to mode the correspondence of syntactic structure itself.

Rambow and Satta (1996) describe a general recipe for creating synchronous grammars. Each grammar is seen as an individual generative device which should be combined. This is achieved via pairing their productions in such a way that right-hand side non-terminals are linked. Derivations proceed via applying the paired grammar rules to previously linked non-terminals. Synchronous versions exist for many formalisms designed for monolingual parsing, below we will discuss the most commonly used in machine translation in more detail.

### 2.3.2 Context-Free Formalisms

A widely used formalism in machine translation is synchronous context free grammar (SCFG) (Lewis and Stearns, 1968). Productions of context-free grammars can be seen as one-level trees, in the synchronous case this restricts productions to the translation and permutation of sibling nodes. This is called the *child-reordering constraint* which effectively prohibits the formalism to capture relationships ranging beyond the one-level tree a single production can model. For example, the mechanism fails to capture a relationship between a child of the left argument of a verb and another verb argument (Rozenberg and Salomaa, 1997). Indeed, it has been shown that this constraint does not allow the representation of certain reorderings occurring in parallel text (Fox, 2002).

This rigidity does not fully describe real data (Eisner, 2003) and therefore attempts to use more powerful formalisms based on tree-transducers (§2.2.1) have been made. These formalisms have an *extended domain of locality* compared to SCFGs meaning that dependencies which range beyond a one-level tree can be captured. The advantage of these extensions is that they can be converted into weakly equivalent SCFGs (Zhang et al., 2006), although, this increases grammar size dramatically.

One of the earliest formalisms used in syntax-based machine translation was inversion transduction grammar (ITG) which is based on the straight or inverted synchronous rewriting of two nonterminal symbols (Wu, 1997). A very simple variant is the so called bracketing transduction grammar (BTG) which is a two-nonterminal SCFG (Aho and Ullman, 1969) using a single undifferentiated category that can match any word sequence. The grammar can be compactly described as

$$\begin{gathered} \mathrm{A} \rightarrow [\mathrm{A}\ \mathrm{A}] \\ \mathrm{A} \rightarrow \langle \mathrm{A}\ \mathrm{A} \rangle \\ \mathrm{A} \rightarrow \mathrm{f}\,/\,\mathrm{e} \end{gathered}$$

where the first two productions are for straight or inverted rewriting respectively and the last production is for general lexical rules where f and e are foreign or English words or phrases. The binary nature of BTG allows for efficient parsing (Wu, 1996) in polynomial time. Despite the simplicity of the formalism it has been found by translational equivalence studies (Wellington et al., 2006) that 95% of alignments in a test corpus could be modeled making it a simple, yet, reasonably expressive formalism.

### 2.3.3 CCG and Mildly Context-Sensitive Formalisms

Mildly context-sensitive grammars are claimed to be adequate to capture certain aspects of human language structure which cannot be modelled by weaker formalisms like CFGs (Joshi et al., 1991) while still being parseable in polynomial time. In this section we will describe synchronous tree adjoining grammars (STAG) and combinatorial categorial grammars (CCG) which belong to the family of mildly-context sensitive grammars.

Tree adjoining grammar (TAG) or the lexicalized version (LTAG) (Schabes, 1992) is based on the combination of elementary tree fragments. The formalism provides an an extended domain of locality compared to CFG which allows to model more complex translations (Joshi, 2004) via being able to capture dependencies within tree fragments rather than only single level trees. However, this comes at the cost of a much higher parsing complexity of $O(n^6)$ compared to the cubic complexity of CFG. Further, the training of machine translation systems requires to parse bilingual corpora, however, synchronous TAG (STAG) (Shieber and Schabes, 1990) has a complexity of even $O(n^{12})$ which is very prohibitive (Nesson et al., 2006). Shieber (2007) argues that the adjoining operation increases the generalization capability of STAG, however, practical statistical implementations (Nesson et al., 2006) removed this operation for efficiency and still needed to restrict the terminal vocabulary severely to carry out practical experiments.

Combinatorial Categorial Grammar (CCG) (Steedman, 2000) is a lexicalized grammar, meaning that the rules of the grammar are completely general and that all language-specific information is given in the lexicon. The formalisms is weakly equivalent to TAG, thus allowing to model the same string languages (Joshi et al., 1991). CCG is based on combining categories which encode valency and directionality. For example, the category for a transitive verb is (S\NP)/NP which encodes the knowledge that there are two arguments to the right and left of the verb. CCG category sets are much larger than the ones used in CFG grammars since they encode richer information. As a result, lexicons can contain as many as 1200 categories, compared to 48 POS-tags of the Penn Treebank (Hockenmaier and Steedman, 2002) usually used in CFG grammars. However, practical implementations of CCG parsers (see Section 2.5) limit the category set and also restrict the categories which are allowed to combine, yet, this has not been found to decrease coverage nor accuracy (Clark and Curran, 2007b).

One of the strengths of CCG is the ability to model long-range dependencies which is problematic with CFG because of the independence assumptions it makes. CCG categories can also be augmented with semantic representations which can be composed synchronously with syntactic derivations. Words are associated with either an atomic (e.g. N for a noun) or a complex category (e.g. S\NP for an intransitive verb). Most importantly, compared to TAG, CCG can represent any leftmost string as a constituent even if its not a syntactic constituent in the sense of CFG. Previous work using context-free grammars required to introduce large numbers of pseudo nonterminal symbols to achieve the same as mentioned in Section 2.2.1.

CCG uses the operators of combinatorial logic which equip it with the expressive power of context-free phrase structure grammar as well as extensions such as functional composition and type raising extend its power. An issue associated with CCG is its susceptibility to *spurious ambiguity* referring to the possibility that there could be multiple derivations with the same semantic interpretation. Steedman (2000) argues that spurious ambiguity is need to capture coordination and other linguistic phenomena. However, it presents a large computational burden and so solutions to restrict the derivations have been sought which we shall describe below.

One way of dealing with spurious ambiguity in CCG is to only consider normal-form derivations which is a derivation using typeraising and composition only when necessary. Eisner (1996) suggests a method to eliminate spurious ambiguity entirely which only allows one normal-form derivation out of all derivations that have the same semantic interpretation. This is achieved via restriction the combination of categories resulting from composition. These restrictions are known as the *Eisner constraints* Notably, they do not restrict the spurious ambiguity resulting from type-raising.

Interestingly, the lexical description of the described formalisms are equivalent meaning that the elementary trees of LTAG describe the same dependency information as CCG categories. Doran and Srinivas (1994) note that LTAG derived trees have are a more rigid structure compared CCG derived trees which allows the latter to better handle *non-constituency* which can make it easy to integrate it into translation models as well as language models.

Another potential application of CCG stems from its use of features which allow to refine categories for example by count, case or gender information. This information together with the long-range dependency capability can be leveraged to facilitate better grammaticality in machine translation in order to tackle issues such as subject-verb agreement. For English a CCG version of the Penn Treebank called CCGbank (Hockenmaier and Steedman, 2007) exists, there is also a German CCG corpus (Hockenmaier, 2006) which allows to train parsing models.

CCG does not require binarization to make it suitable for efficient parsing since it is already binary. This allows to use standard chart parsing techniques such as CKY. Theoretically, CCG has a parsing complexity of $O(n^6)$ like TAG, however, in practice $O(n^3)$ can be achieved (Hockenmaier, 2003a). A number of parsing models (see section 2.5) have been developed making it a formalism for with which wide-coverage can be achieved. Despite those advantages applications of CCG to machine translation have been very limited as we will detail in Section 2.4.

## 2.4 Applications of CCG in Machine translation

Previous work in using CCG in machine translation concentrated on using supertags (see Section 2.5.1) to syntactically inform phrase-based translation. Both Birch et al. (2007) and Hassan et al. (2007) integrate CCG supertags into phrase-based models and use n-gram language models over target-side supertags to improve fluency. The former uses the general factored approach of Koehn and Hoang (2007) whereas the latter directly modifies the translation model. Birch et al. (2007) reports that their performance increase stems mostly from better local reordering.

These approaches did not make use of the following advantages CCG offers: Firstly, the ability to represent phrasal non-constituents, since supertags are only word-level descriptors. Secondly, the flexibility CCG parsing provides over more rigid formalisms such as CFG. And thirdly, the ability to capture long-range dependencies which cannot be realized with supertag-based n-gram models.

## 2.5 CCG Parsing

CCG parsers can be used in a machine translation setting to annotate parallel training data. The presented models may serve as a starting point for a CCG parsing-based machine translation model which can generate more grammatical output.

### 2.5.1 Supertagging

Most parsers for LTAG and CCG perform *supertagging* (Bangalore and Joshi, 1999) before the actual parsing step. A supertagger assigns elementary trees (LTAG) or lexical categories (CCG) to each word in the sentence using statistical sequence tagging techniques based on Maximum Entropy or Hidden Markov Models. The supertags contain already very much syntactical information which makes parsing faster and more accurate, therefore the TAG-community refers to supertagging as almost parsing (Bangalore and Joshi, 1999). It has been demonstrated for both CCG and LTAG that using good supertags can dramatically increase parsing speed as well as accuracy (Clark and Curran, 2007b; Nasr and Rambow, 2004).

Supertaggers can assign single or multiple tags to a word, too few may prevent the parser from discovering the correct derivation and too many may lead the parser astray, or at least slow it down. *Adaptive supertagging* (Clark and Curran, 2007b) is a term referring to the integration of the supertagger and the parser, the idea is to start off with a small set of tags and request more if no analysis for the sentence can be obtained. This has been shown to increase parsing speed without decreasing accuracy or coverage. Clark and Curran (2007b) notes that the performance of LTAG supertaggers is lower than those of CCG equivalents and suspects that this could be because of the nature of the formalisms or the properties of the extracted grammars.

### 2.5.2 Generative Models

In the literature both generative (Hockenmaier, 2003a; Hockenmaier and Steedman, 2002) and discriminative parsing models (Clark and Curran, 2007b) have been proposed. In early work, Hockenmaier (2001) describes a very simple generative model based on top-down generation of binary and unary trees. The model has no notion of combinatory rules and is estimated on rule instantiations of CCGbank, restricting it to the normal-form derivations of the corpus. The defined probability distribution conditions only on a very local context such as the category of the current node, the parent and a potential sibling node. Subsequent work (Hockenmaier and Steedman, 2002) extends the model via additionally conditioning on lexical dependencies or grandparent categories which improves performance for some features. The authors conclude that data sparseness is a problem when trying to integrate many features into the generative model.

The approach described above is only concerned with modelling normal-form derivations which occur in the training corpus. However, one of the strengths of CCG is its ability to model dependencies for which predicate-argument structures are used. These structures define the dependencies between lexical items in a derivation and dependency-based parsing models (Hockenmaier, 2003b) try to recover them.

Generative models also try to model the probability of generating words which is a difficult task with the limited amount of available training data. These models have sparse lexical probabilities and a large problem are words which had not been observed with the necessary category in the training data. This adversely effects coverage of the parser. A workaround proposed by (Hockenmaier and Steedman, 2002) is to replace rare words in the training corpus with part-of-speech tags and to estimate their lexical probabilities based on the tag.

### 2.5.3 Discriminative Models

A more recent approach (Clark and Curran, 2007b) based on discriminative methods outperforms the generative models presented above. The problem of sparse lexical probabilities and their effect on coverage is solved via integrating a supertagger with the parser which they refer to as *adaptive supertagging* (see Section 2.5.1). This results in a coverage of over 99% and increases parsing speed significantly. Their approach is based on a log-linear model which combines a plethora of features (over 500k) defined over combinations of words, part-of-speech tags, lexical-categories and rule instantiations. The authors describe both a normal-form model and a dependency model whose feature weights are estimated on packed charts of training-sentence derivations.

The authors place a number of restrictions on the grammar to keep the size of the charts manageable. First, only lexical categories which occur at least ten times in CCGbank are allowed which results in 425 types instead of 1207. Second, grammar rules are restricted by the Eisner constraints (see Section 2.3.3) in their normal-form model. Thirdly, and similar to the generative models before, two categories are only allowed to combine if they have been seen to combine in the training data. The authors report no detrimental effects on parsing accuracy and coverage when restricting rule combinations to observed instantiations (Clark and Curran, 2007b).

Experimental results show that both their normal-form and dependency-model outperform previous generative approaches. Interestingly, the normal-form model outperforms the dependency-model with an F-Score of 86.73 versus 85.08 despite being simpler and easier to train. Although, the best performing model is the dependency-model with Eisner-constraints which achieves an F-Score on predicate-argument structure recovery of 87.24 Clark and Curran (2007b). From machine translation perspective it might be more useful to model derivations in a first step since a dependency-model is computationally much more expensive.

Clark and Curran (2007a) proposes a very efficient perceptron training regime which requires significantly less memory (20MB instead of 20GB) than the log-linear model of Clark and Curran (2007b) while maintaining the same accuracy and coverage. Decoding is identical to the log-linear model and training is effectively decoding: The Viterbi-decoding algorithm finds the features which correspond to the highest scoring derivation and then updates the current feature weights accordingly. This is much simpler than the inside-outside algorithm used in the log-linear model. Being able to search the space of *all derivations* is only possible because of the lexicalized nature of CCG: The key idea is to use the supertagger to restrict the chart-sizes which makes training manageable via limiting the number of supertags assigned to each word which in turn determines the number of possible derivations.

Most strikingly, the training regimes described here allow to *enumerate all derivations* of a sentence exhaustively. This compares very favorably to other approaches which can do so only for sentences with a limited length of about 15 words (Taskar et al., 2004) or had to use a heuristic beam-search for training (Collins and Roark, 2004).

### 2.5.4 Parsing and Domain-Adaption

A key question when using CCG parsing techniques in machine translation is on how to adapt models to new domains. Clark and Curran (2007b) mentions evidence that the performance of parsers trained on newspaper text drops significantly in other domains. Developing training data for each domain is infeasible and impractical.

One idea is to adapt only the supertagger to a new domain via creating training data with gold-standard supertags (Clark et al., 2004). This is more efficient than creating a new treebank and makes use of the fact that already a lot of syntactic information is contained in the supertags. This assumes that the parsing model itself is transferable, making it very attractive for machine translation.

# Chapter 3

# Thesis Proposal

## 3.1 Preliminary Work

In preliminary work (Auli et al., 2009) we found that the pre-dominant machine translation models are searching roughly the same space. These search spaces are very large and include up to 37% of the desired translations exactly and very likely many more with only slight differences compared to the reference translation. This finding suggests that the decisions these models make during decoding are often suboptimal, offering a large potential for improvement.

In the same study we found that the performance of these models is fairly equal on a range of language pairs when evaluated on a standardized setup. The results of another publication (Zollmann et al., 2008) show that this similarity persists even between syntactic and non-syntactic models. In a more qualitative analysis we have found that the majority of translation errors are due to the failure of translation models to produce grammatical output. Table 3.1. shows a summary of the qualitative errors observed in a small set of sentences translated by Moses (Koehn et al., 2007), a phrase-based decoder, and Hiero (Chiang, 2005), a hierarchical decoder.

1. A notion of case, number and for some languages, gender, is required. We have described in Section 2.3.3 that CCG allows to subcategorize existing types to achieve this. The model also needs to ensure that agreement is ensured across the sentence and not just in a local context such as n-gram based models do.

2. The model should have a sense of how constituents should be ordered in a translation, for example when translating from an SVO into an SOV language, then the model should be able to capture this difference. Or on a more local level,

| | Hiero | Moses |
|---|---|---|
| Agreement | 36 | 31 |
| Word-Order | 16 | 18 |
| Verb-translation | 43 | 48 |

Table 3.1: Qualitative translation errors for Moses and Hiero on the first 40 sentences of the WMT07 testset for the English to German task.

constituents such as verbs and objects should be ordered in such a way as it is required by the verb. Again, the valency and directionality of CCG categories help in this respect.

These requirements formed the basis for our proposal which we shall describe in the following section.

## 3.2 Proposal Overview

We propose a number of statistical translation models based on fully-fledged CCG parsing which has not been investigated before as described in Section 2.4. These models will benefit from the following advantages the formalism brings with it:

1. **Non-constituency:** Handling non-constituents is crucial in statistical translation and CCG allows to devise models which can deal with non-constituents in a syntactical way. For example, CCG can simply describe the partial noun phrase “the mutual” as NP/N whereas the introduction of non-syntactical pseudo-non-terminals is required for context-free grammars.

2. **Dependency-based:** CCG is very well suited for modelling long-range dependencies which is lacking in current machine translation models. We hope to be able to leverage this ability to mitigate agreement errors which cannot be captured by local n-gram models.

3. **Parsing efficiency:** CCG-based parsing models (Section 2.5) are very efficient, we aim to leverage the insights gained from monolingual parsing in a machine translation setting.

4. **Grammar size:** CCG is based on combining continuous constituents in a binary fashion. The productions used by a CCG-based model are therefore similar to the phrasal translation rules used by a phrase-based system except that they are augmented by CCG categories. We have found in the work done so far (see Section 3.5) that the grammar sizes are in fact comparable to the ones used by phrase-based systems. This is more manageable than the large grammars used by SCFG-based systems whose extraction is subject to many constraints.

This thesis is about exploring CCG in translation and to leverage its advantages to obtain better translation models. In particular, we are proposing the following:

1. A string-to-tree model which uses CCG on the target side to produce more grammatical output. The model uses monolingual parsing techniques and a binary reordering model to create a valid CCG derivation spanning the foreign sentence.

2. A synchronous extension of CCG which can model translational equivalence. This can be used to model the correspondence of syntactical structure in parallel text rather than using a projection of a grammar from one side onto the other.

## 3.3 A String-to-Tree Parsing-based Translation Model

### 3.3.1 Grammar and Rule Extraction

The grammar consists out of lexical translation rules with the following general form

$$C \rightarrow f/e$$

where $C$ is a CCG category label, $f$ is a foreign phrase and $e$ is an English constituent that corresponds to $C$. An example production is `NP/N` → `das blaue` / `the blue` which is a typical example for a non-constituent phrase-pair that can be conveniently represented with CCG. The lexical translation rules are then combined into target language syntactical structure using CCG combinatorial rules.

Lexical rules are extracted from a word-aligned parallel corpus similar to (Koehn et al., 2003). Notably, the CCG category label $C$ is not only restricted to lexical categories of single words, it can be *any category occurring in the parse* of the target-side which spans the phrase $e$. To increase the size of the grammar, not only categories from the 1-best parse of the parallel corpus are extracted but from the entire parse-forest available to the parser. This is similar to forest-based approaches used for SCFGs (Mi and Huang, 2008). We found that the used parser (Clark and Curran, 2007b) provides us under default settings with categories for 52% of all $O(n^2)$ possible word-spans, where $n$ is the length of the foreign sentence. We expect even higher coverage when relaxing some of the restrictions. Currently, we restrict the number of categories to 425 which can achieve a parsing coverage of 99% on newspaper text (Clark and Curran, 2007b).

### 3.3.2 Model

The model translates a foreign sentence with words $f_0, f_1, ..., f_n$ of length $n$ into an English sentence with words $e_0, e_0, ..., e_m$ of length $m$ using a CKY-based parsing algorithm. Notably, no binarization of the grammar in any sense is required, since the formalism is *already binary*. Figure 3.1 illustrates the translation process: In the mapping step, foreign string sequences $f$ are translated into CCG constituents $e$ which are placed into the the parsing chart, word reordering is modelled via constituent-swaps as described in Section 3.3.3. Notably, there can be up to $O(n^2)$ segmentations of the foreign sentence resulting in greater flexibility compared to phrase-based models which allow only a single segmentation of the input. If the number of possible entries within a chart-cell is restricted to $k$, then we may generate up to $O(kn^2)$ entries in this step.

We are planning to deal with spurious ambiguity in the same way as (Clark and Curran, 2007b) via enforcing the Eisner constraints (Section 2.3.3) and only allowing two categories to combine if they have been seen to combine in the training data of the parsing model i.e. CCGbank. The authors report no loss in coverage nor accuracy with the added benefit of a much faster parsing model (Clark and Curran, 2007b).

More formally, and similar to (Koehn et al., 2003; Chiang, 2005) we use a log-linear model over derivations:

$$P(D) \propto \prod_i \phi_i(D)^{\lambda_i} = \prod_i \prod_{(C \to f/e) \in D} \phi_i(C \to f/e)^{\lambda_i} \tag{3.1}$$

which is a product over features $\phi_i$, weighted by $\lambda_i$, defined over productions $C \to f/e$ used in derivations $D$. The set of features includes the standard features of (Chiang, 2005) including an n-gram language model. Our CCG-parsing model is defined as another feature $P_{CCG}$ which gives us a measure on how likely the combination of two

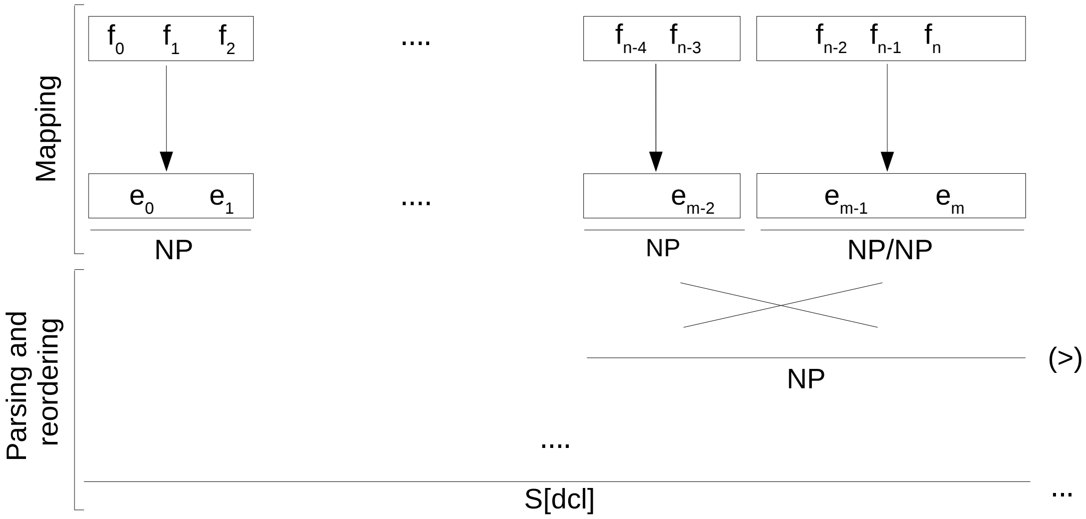

Figure 3.1: Illustration of a string-to-tree parsing-based translation model. The mapping step turns foreign phrases into English CCG-labelled constituents. This is followed by deriving a CCG parse for the sentence and reordering constituents as required.

categories and their associated English constituents is. Only categories which can be combined are considered.

The parsing model can be either the simple generative model (Hockenmaier, 2001) (see Section 2.5.2) or a more elaborate discriminative model (see Section 2.5.3). The former conditions only on the type of the resulting and the combined categories which may be appealing for the resource-intensive domain of machine translation. The log-linear model of (Clark and Curran, 2007b) uses a large number of features which may be too high a computational burden for a parsing-based machine translation model, however, subsequent speed-ups in the training-regime of the parser we intend to use (Clark and Curran, 2007a) may allow to quickly train up less elaborate models which can be used in machine translation.

### 3.3.3 Word-Reordering

The reordering of words is carried out on the constituent level via combining the categories of two sub-derivations in a straight or inverted fashion. The described model allows all binary bracketings of the possible input segmentations. Translational equivalence studies have shown that 95% of word-reorderings can be modelled under these constraints (Wellington et al., 2006). Arguably, considering the rather bad performing reordering capabilities of existing systems, not being able to model a small part of real-world reorderings may be a small loss.

Our model restricts the number of possible permutations further due to the syntactic constraints it imposes. This can be seen as a way to bring syntactic structure into the search space which is still very large. We either combine two categories in a straight or inverted fashion if permitted by the CCG parsing model. This is like two additional rules which place two sub-derivations in straight (S) or inverted (I) order. We define a reordering feature which is integrated into the log-linear model as follows:

$$P_r(D) = p(o|X,Y) \tag{3.2}$$

where $o \in S,I$ and X and Y are sub-derivations. The feature can be either a simple Maximum Likelihood Estimated model based only on the sub-derivation root-categories, or a more flexible discriminative model similar to the parsing model described above. Xiong et al. (2006) describes how a Maximum Entropy based reordering model can be integrated into a BTG decoder which uses up to 100k features. Discriminative reordering models have also been successfully applied to phrase-based translation (Zens and Ney, 2006). In any case, the syntactical constraints enforced by the grammar formalism should already restrict a great number of ungrammatical reorderings. Figure 3.2 and Figure 3.3 illustrate how a sentence requiring substantial reordering can be translated with this model.

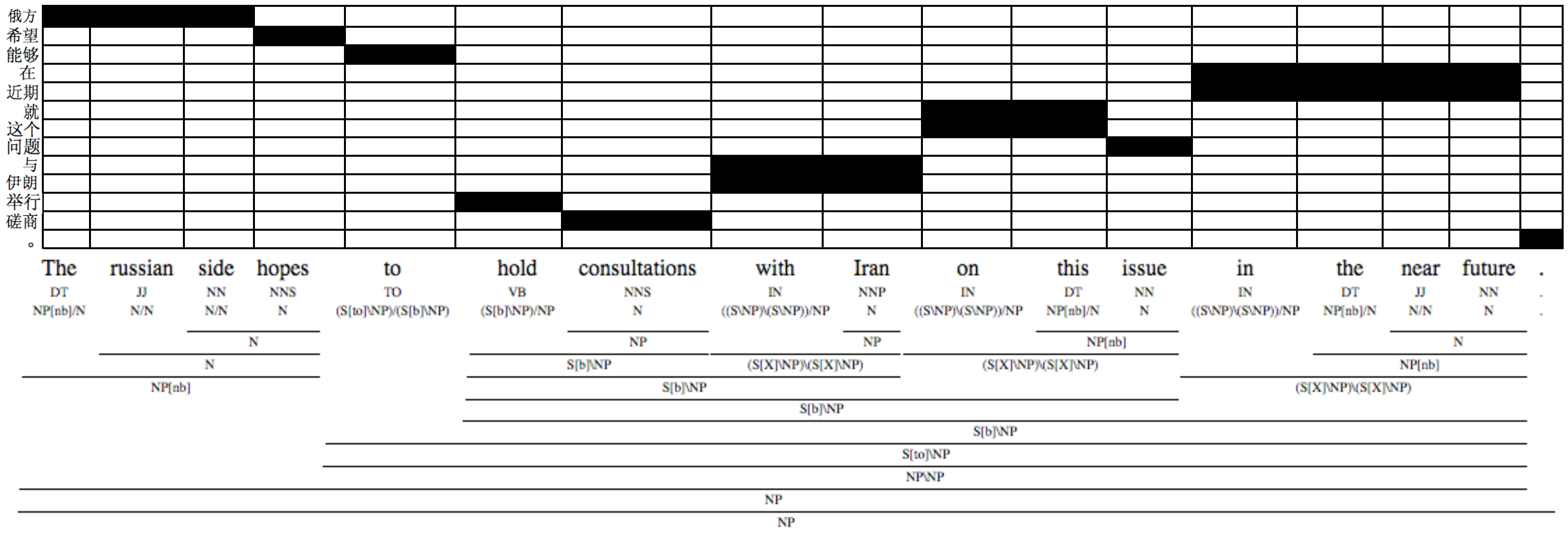

Figure 3.2: Word-aligned Chinese-English sentence (taken from Galley and Manning (2008)) with substantial word-order differences. The English side is augmented by a monolingual CCG-parse.

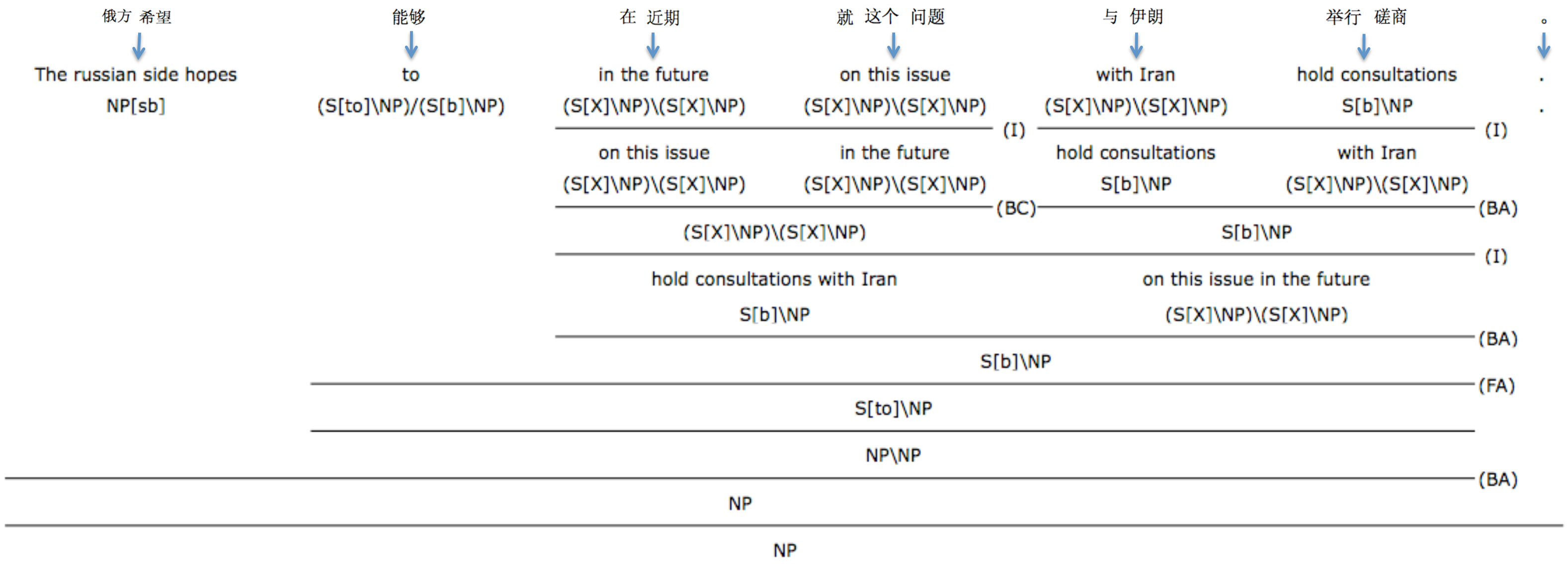

Figure 3.3: Illustration of a derivation by the string-to-tree parsing-based model. The input sentence is segmented and lexical translation rules map the Chinese phrases into English. Reordering of the constituents is achieved through parsing the sentence and applying a reordering model. The derivation involves the forward application (FA) and backward application (BA) as well as the reordering rules, inverted (I) and straight (implicit).

### 3.3.4 Estimation

We reported on perceptron-based training procedures which can enumerate all derivations (Section 2.5.3) for monolingual CCG parsing models. However, we do not believe that doing so will be possible for the described machine translation model despite the restrictions the formalism places on the number possible reorderings. We will therefore follow the machine translation community in using heuristics such as in (Chiang, 2007; Koehn et al., 2003) to hypothesize a distribution of possible rules. Although, the estimation for the distributions of the parsing and reordering models should be possible without heuristics.

### 3.3.5 Search

We described above informally how the model can be used to obtain translations. In this section we want to detail how this can be achieved with respect to the resource limitations we have in finding good translations. We are using a CKY chart-parsing algorithm with a binary grammar to find the highest scoring English derivation. Each derivation-step uses CCG combinatorial rules to create a new item $[C,i,j,e_l,e_r]$ where $C$ is the resulting category of combining two subderivations which cover adjacent source-language strings from $f_i$ to $f_j$, $e_l$ and $e_r$ are the language model contexts which permit the use of a non-local n-gram language model when items are combined. Our goal is to find the highest scoring item $[C,0,n,\perp,\perp]$.

We are only looking for the highest scoring derivation and so items which are identical except for their antecedents and weights can be recombined so that only the highest scoring item kept. The complexity of decoding is as follows:

$$O(n^3[|N||T|^{2(g-1)}]^K) \tag{3.3}$$

where $n$ is the length of the foreign sentence, $N$ and $T$ are the sets of categories and terminals, $g$ is the n-gram model size and $K$ is the maximum number of category pairs per rule. This shows that decoding complexity gets prohibitively large with even small sets of non-terminals or categories. We will therefore experiment with a number of pruning strategies to keep the search space manageable.

The simplest strategy is to use a heuristic pruning technique such as cube pruning (Chiang, 2007) which generates only the $k$ highest scoring items without examining the rest. Doing so has been shown to be effective when using a single undifferentiated non-terminal category. However, our model is based on 425 category-sets and it may easily be the case that the top $k$ items are all of the same category-type. This may lead to a *lack of diversity* in the available categories for a chart-cell which may prohibit to discover a derivation which depends on lower-scoring sub-derivations. A solution for this is to store $k$ items for each category-type regardless of the score per chart-cell in order to preserve diversity. Approaches using multi-pass decoding have been described in Section 2.2.1.

Notably, it has been shown that SCFG-based models using up to 4000 non-terminals (Zollmann and Venugopal, 2006) can outperform state of the art phrase-based and single non-terminal-based hierarchical systems (Zollmann et al., 2008) whereas our approach uses a set which is magnitude lower (425 categories).

### 3.3.6 Unknown words

Unknown words are foreign constituents for which there is no translation rule in the grammar. These words pose a problem during translation since it is not only that they cannot be translated into English but also that no CCG category will available for the corresponding span, effectively prohibiting the discovery of a target-language parse.

We propose an imperfect solution related to an approach successfully used in monolingual CCG parsing (Hockenmaier, 2003a) which is based on predicting supertags from less sparse part-of-speech tags. We are planning to use a distribution $P(C_u|\Theta)$ to obtain a category $C_u$ for an unknown foreign word sequence $f_{u..v}$ which spans the foreign words $f_u$ to $f_v$. The distribution is conditioned on some local context (lexical or syntactical), denoted as $\Theta$. Note, that one of the features of the model is the length of the foreign word sequence $f_{u..v}$ in order to distinguish multiple unknown words from single ones.

A simple generative model similar to (Hockenmaier, 2003a) may be estimated from a target-side CCG annotated parallel corpus as follows: Categories are removed from the phrase-pairs whose foreign side occurs less than $k$ times and the foreign side is part-of-speech tagged. Based on this, we can estimate $P(C_u|\Theta) = P(C_u|POS(f_u))$ from these words using Maximum Likelihood Estimation. A more elaborate discriminative model may condition on a plethora of features around the unknown word sequence $f_{u..v}$.

### 3.3.7 Using CCG Dependencies

Integration of a CCG dependency model would allow us to recover dependencies between constituents and to enforce agreement of number, case or even gender. An additional feature which determines how well constituents agree can further improve the model. The current version of CCGbank does not contain this information and so one of tasks would include to annotate it, potentially with automated tools.

For example, consider the sentence “The book , including all the chapters in the first section , is interesting .” which contains a long-range dependency between the subject “book” and the verb “is”: A dependency-based CCG parsing model can be used us to discover this dependency during translation and to enforce agreement between the two constituents via comparing number features. If they agree, the derivation will receive higher weight compared to other non-grammatical derivations.

### 3.3.8 Discontinuous Rules

Discontinuous rules give a model more generalizational power which is a useful property as discussed in (Melamed et al., 2004). Our model can be too generalized via allowing the right-hand sides of rules to include CCG categories. For example, consider the lexical translation rule based on the phrase ”the russian sides hopes” from the example in Figure 3.2.

$$NP[nb] \quad \rightarrow \quad \text{die russische seite hofft} \quad / \quad \text{the russian side hopes}$$

this rule can be generalized as follows

$$NP[nb] \quad \rightarrow \quad \text{die } N/N_1 \text{ seite } VBZ_2 \quad / \quad \text{the } N/N_1 \text{ side } VBZ_2$$

where subscripts indicate linked non-terminals. The extraction procedure is then the same as the one used by (Chiang, 2007). Translation proceeds then as a combination of CCG parsing and substitution of categories within rules.

## 3.4 A Synchronous Translation Model

A synchronous grammar formalism allows to model the generation of pairs of strings capturing the translational equivalence within parallel texts. We discussed that for many formalisms like TAG and CFG synchronous formalisms exist (see Section 2.3) which are able to model both the translation and permutation of constituents between two languages. A synchronous formalisms can also be used to parse parallel text in order to discover the hidden relationships within in. We propose to extend CCG so that it can model the structure of parallel text.

Synchronization is achieved via combining two generative devices through pairing their productions in such a way that right-hand side non-terminals are linked. The derivation proceeds via applying the paired grammar rules to previously linked non-terminals (Rambow and Satta, 1996). Synchronization in CCG may be achieved through generalizing the existing set of combinatory rules in a similar way. For example, the forward application rule

$$X/Y \quad Y \quad \rightarrow \quad X \quad (FA) \tag{3.4}$$

can be generalized to the synchronous case via introducing two new rules which either apply forward application in both languages or forward application in the first language and backward application (BA) in the second language:

$$X_{\boxed{1}}/Y_{\boxed{2}} \quad Y_{\boxed{3}} \quad : \quad X_{\boxed{1}}/Y_{\boxed{2}} \quad Y_{\boxed{3}} \quad \rightarrow \quad X_{\boxed{1}} \quad : \quad X_{\boxed{1}} \quad (FA,FA) \tag{3.5}$$

$$X_{\boxed{1}}/Y_{\boxed{2}} \quad Y_{\boxed{3}} \quad : \quad Y_{\boxed{3}} \quad X_{\boxed{1}}\backslash Y_{\boxed{2}} \quad \rightarrow \quad X_{\boxed{1}} \quad : \quad X_{\boxed{1}} \quad (FA,BA) \tag{3.6}$$

Using these synchronous operators we can model the translation of a simple clause whose structure is shown in both German and English in Table 3.2.

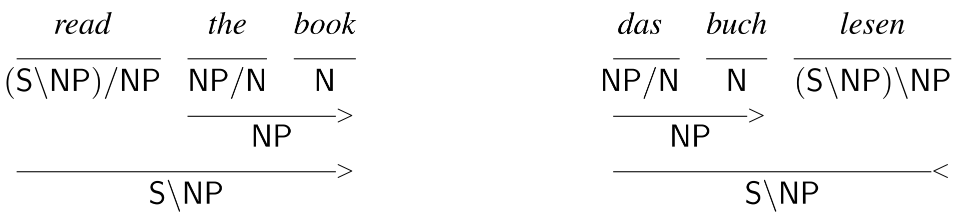

Table 3.2: German and English CCG derivation of a simple clause in both languages.

In this example, the English translation places the verb (”read”) to the right of the object as opposed to the German version which puts it at the left. Using synchronous operators we can model this difference as follows: To translate ”das buch” into ”the book” we apply the forward application in both languages (FA,FA), but to reorder ”lesen” (read), we have to use backward application in German (FA,BA) while forward application in English. This reorders the constituents as necessary.

A synchronous definition of CCG in this manner gives us the capability to model reordering as a binary process. We expect a better translational equivalence model via using discontinuous rules (see Section 3.3.8) and via generalizing the CCG rules which make the formalism mildly context-sensitive.

## 3.5 Work So Far

### 3.5.1 Translation Rule Extraction

We parsed the German-English part of the Europarl corpus (Koehn, 2005) consisting out of one million sentences using the parser of (Clark and Curran, 2007b) and extracted a CCG-informed grammar (see Section 3.3.1).

Single 1-best parse trees do not contain much syntactic material to label the many phrase-pairs which can be extracted from a parallel corpus. We therefore modified the parser to output the highest scoring category for *every span* resulting in categories for nearly 52% of all possible English spans with the standard beam-width of the parser. Note that not all possible $O(n^2)$ spans will be part of a phrase-pair and that higher coverage is expected when increasing the beam-width. For actual phrase-pairs, we obtained categories for nearly 75% of rules. We used the online rule-extraction method of Lopez (2007) as implemented in Li et al. (2009) to generate the grammar.

Interestingly, most phrase-pairs for which no category could be assigned can actually be parsed individually. It is likely that this parse is part of a substructure of the global parse which is pruned during beam-search. This suggests that a far higher coverage is possible if we use a more sophisticated method to extract categories. We intend to use a complete derivation forest such as in the training procedure of the parsing model which is not subject to heavy pruning as it happens during actual parsing.

### 3.5.2 Model Implementation

We modified an existing hierarchical phrase-based decoder (Li et al., 2009) to use our grammar and to parse translations into valid CCG structures as described in Section 3.3. Two types of rules are used:

1. **Translation-rules** which are continuous pairs of phrases like NP → das haus / the house. The local lexical features associated with these rules such as $p_{lex}(e|f)$ and $p_{lex}(f|e)$ are estimated as in Chiang (2005). However, the phrasal translation probability $p(e|f)$ is replaced by $p(e,C|f)$ where $C$ is the CCG-category heading the rule since a pair e,f can have more than one category.

2. **Combination-rules** implement a simple *Maximum Likelihood-based CCG parsing model* within the framework of hierarchical phrase-based parsing. All rules can be enumerated since we restrict our model to rule instantiations which have been observed in CCGbank. Doing so has been shown to result in no loss of accuracy or coverage for monolingual parsing (Clark and Curran, 2007b). We can therefore conveniently implement the combination rules as SCFG-rules which fits the formalism used in our base system. CCGbank has 3500 different rule instantiations, we generalize them as follows to allow *word reordering* via introducing SCFG rules that combines two categories in straight or inverted fashion. For example, given the CCGbank rule-instantiation

   $$S[adj]\backslash NP[conj] \quad \rightarrow \quad \text{conj} \quad S[ng]\backslash NP$$

   we obtain two SCFG rules:

   $$\begin{array}{llllllll} (1)\ S[adj]\backslash NP[conj] & \rightarrow & conj_1 & S[ng\backslash NP_2 & / & conj_1 & S[ng\backslash NP_2 \\ (2)\ S[adj]\backslash NP[conj] & \rightarrow & conj_1 & S[ng\backslash NP_2 & / & S[ng\backslash NP_2 & conj_1 \end{array}$$

   The parsing model is defined through a Maximum Likelihood estimated distribution $p(C_{par}|C_{left},C_{right})$ which solely conditions the likelihood of $C_{par}$ on the categories of the argument-categories $C_{left}$ or $C_{right}$. Reordering is achieved through defining two SCFG rules per CCG rule instantiation. Currently, both rules have the same probability so that straight or inverted combination has the same probability.

### 3.5.3 Preliminary Results

We compared the performance between our model and a BTG baseline which is a reduction of our CCG-implemenation using a non-syntactic grammar. For the baseline the decoder was modified to have two productions which simply rewrite two nonterminals in a straight or inverted order (see BTG in Section 2.3.2) with equal probability such as in our model. Systems were trained on the first 100k sentences of the German-English part of the Europarl corpus (Koehn, 2005) and tested on the 2000 sentences of the WMT08 development set.

| Model | BLEU |
|---|---|
| BTG | 17,22 |
| BTG (restricted) | 16.67 |
| CCG | 16.81 |

Table 3.3: BLEU score results for the pilot system on the WMT08 German-English testset. The BTG (restricted) system uses the phrase-pairs available to the CCG system.

Table 3.3. shows the BLEU scores for the baseline (BTG) and when restricting the baseline to the phrase-pairs available to the CCG system. The CCG system did not produce translations (1635 translations for 2000 sentences) for all sentences and the BLEU score reported is based on the union of the BTG and CCG systems. Table 3.4. shows a number of example translations. In the first example, CCG has a much better word-order, the second translation shows how CCG avoids using a noisy translation rule as opposed to BTG. The last example demonstrates that the mechanism sometimes breaks down.

| | |
|---|---|
| **BTG:** | very well aware that the treaties , is inadequate in the future institutional structure , the union for a more efficient structure ... |
| **CCG:** | we know very well that the treaties are inadequate and a better structure for the union needs a more institutional well-defined structure .... |
| **Ref:** | we know all too well that the present treaties are inadequate and that the union will need a better and different structure in future ... |
| **Src:** | uns ist sehr wohl bewusst , dass die geltenden vertraege unzulaenglich sind und kuenftig eine andere , effizientere struktur fuer die union entwickelt werden muss ... |
| **Gloss:** | we is well aware , that the current treaties inadequate are and in future a different , efficient structure for the union developed will need ... |
| **BTG:** | a picture emerging in the european liberal , democrat and reform party . |
| **CCG:** | it reflects the emerging liberal europe . |
| **Ref:** | it is an accurate reflection of the liberal europe which is being built. |
| **Src:** | sie ist ein abbild des in entstehung begriffenen liberalen europas . |
| **Gloss:** | it is a reflection the in emerging liberal europe . |
| **BTG:** | the situation is unacceptable , particularly for the self-employed hauliers , at all . |
| **CCG:** | the many transport self-employed , it is absolutely unacceptable , needs to be improved . |
| **Ref:** | the current situation, which is intolerable, particularly for many independent haulage firms and for agriculture, does in any case need to be improved . |
| **Src:** | die jetzige situation , die besonders fuer viele selbstaendige transportunternehmen , fuer die landwirtschaft untragbar ist , muss auf alle faelle verbessert werden . |
| **Gloss:** | the current situation , the particular for many self employed transport companies , for the agriculture unbearable is , must in any case improved need . |

Table 3.4: Example translations of the CCG and BTG systems. The reference (Ref), source (Src) and a literal translation of the source (Glooss) is shown.

## 3.6 Future Targets

We are planning the following future work:

- **Pilot-extensions:** The pilot presented above does not distinguish explicitly between the two types of rules outlined above which results in some undesired pruning of combination rules or translation rules. We are planning to introduce separate stacks for each type to avoid this problem. Also, reordering is currently not explicitely modelled, we intend to apply lexical reordeirng models or maximum entropy models used for BTG (Xiong et al., 2006). Additionally, unknown words are not handled effectively restricting translation to sentences containing only known words.

- **Higher coverage:** Currently, CCG-categories are obtained from a pruned parser-chart which is likely the reason why many phrase-pairs cannot be labelled. We want to remedy this via extracting categories from a packed forest which encodes *all* possible derivations. To choose the truly best category for a given span, the marginal probability of categories shall be computed once all derivations are known.

- **Better parsing model:** Our pilot implements a very simplistic model which can be easily improved via additional features such as in Hockenmaier and Steedman (2002) or via moving to a more sophisticated discriminative model similar to Clark and Curran (2007b).

- **Search:** Our model drastically increases the number of non-terminals compared to the simple BTG baseline which results in a much larger search space. We want to experiment with multi-pass decoding techniques (see Section 3.3.5) or heuristics that preserve more diversity in the beam of a single-pass approach.

- **Discontinuous rules and CCG-Dependencies:** These extensions allow our model to use discontinuous translation patterns as well as dependencies which are one of CCG's strengths (see Section 3.3.8 and Section 3.3.7).

- **Synchronous Formalism:** We sketched a possible extension of CCG to the synchronous case and plan to evaluate how well it models translational equivalence on hand-aligned corpora similar to Wellington et al. (2006).

## 3.7 Likely Outcome

The outcome of the proposed work is an implementation of CCG-based statistical machine translation models. Ideally, this framework can leverage the advantages of the formalism to achieve better machine translation performance than the state of the art. The system should be able to translate also into non-English languages for which CCG-resources exist, such as German (Hockenmaier, 2006).

# Bibliography

Aho, A. V. and Ullman, J. D. (1969). Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3:37–57.

Auli, M., Lopez, A., Koehn, P., and Hoang, H. (2009). A systematic analysis of translation model search spaces. In *Proc. of WMT*.

Avramidis, E. and Koehn, P. (2008). Enriching morphologically poor languages for statistical machine translation. In *Proc. of ACL*, pages 763–770, Columbus, Ohio. Association for Computational Linguistics.

Bangalore, S. and Joshi, A. K. (1999). Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):238–265.

Birch, A., Osborne, M., and Koehn, P. (2007). CCG supertags in factored statistical machine translation. In *Proc. of WMT*, pages 9–16, Prague, Czech Republic. Association for Computational Linguistics.

Blunsom, P., Cohn, T., and Osborne, M. (2008). A discriminative latent variable model for statistical machine translation. In *Proc. of ACL-HLT*, Columbus, Ohio. Association for Computational Linguistics.

Blunsom, P. and Osborne, M. (2008). Probabilistic inference for machine translation. In *Proc. of EMNLP*. Association for Computational Linguistics.

Chan, Y. S., Ng, H. T., and Chiang, D. (2007). Word sense disambiguation improves statistical machine translation. In *Proc. of ACL*, pages 33–40.

Charniak, E., Knight, K., and Yamada, K. (2003). Syntax-based language models for statistical machine translation. In *Proc. of MT Summit IX*.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*, pages 263–270.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Clark, S. and Curran, J. R. (2007a). Perceptron training for a wide-coverage lexicalized-grammar parser. In *Proc. of ACL Workshop on Deep Linguistic Processing*, pages 9–16.

Clark, S. and Curran, J. R. (2007b). Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Clark, S., Steedman, M., and Curran, J. R. (2004). Object-extraction and question-parsing using CCG. In *Proc. of EMNLP*, pages 111–118, Barcelona, Spain.

Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Collins, M., Koehn, P., and Cučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proc. of ACL*, pages 531–540.

Collins, M. and Roark, B. (2004). Incremental parsing with the perceptron algorithm. In *Proc. of ACL*, page 111, Morristown, NJ, USA. Association for Computational Linguistics.

Cowan, B., Kučerová, I., and Collins, M. (2006). A discriminative model for tree-to-tree translation. In *Proc. of EMNLP*, pages 232–241.

Doran, C. and Srinivas, B. (1994). Bootstrapping a wide-coverage CCG from FB-LTAG. In *Proc. on TAG*.

Eisner, J. (1996). Efficient normal-form parsing for combinatory categorial grammar. In *Proc. of ACL*, pages 79–86, Santa Cruz.

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *Proc. of ACL (companion volume)*, pages 205–208.

Fox, H. J. (2002). Phrasal cohesion and statistical machine translation. In *Proc. of EMNLP*, pages 304–311.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *Proc. of ACL*, pages 961–968.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In *Proc. of HLT-NAACL*, pages 273–280.

Galley, M. and Manning, C. D. (2008). A simple and effective hierarchical phrase reordering model. In *Proc. of EMNLP*, pages 848–856, Morristown, NJ, USA. Association for Computational Linguistics.

Gimpel, K. and Smith, N. A. (2008). Rich source-side context for statistical machine translation. In *Proc. of WMT*, pages 9–17, Columbus, Ohio. Association for Computational Linguistics.

Hassan, H. (2008). *Lexical Syntax for Statistical Machine Translation*. PhD dissertation, Dublin City University.

Hassan, H., Hearne, M., and Way, A. (2007). Supertagged phrase-based statistical machine translation. In *Proc. of ACL*, pages 288–295, Prague.

Hassan, H., Sima'an, K., and Way, A. (2008). A syntactic language model based on incremental ccg parsing. In *Proc. of IEEE-SLT*, pages 205–208.

Hockenmaier, J. (2001). Statistical parsing for CCG with simple generative models. In *Proc. of ACL Student Research Workshop*, pages 7–12.

Hockenmaier, J. (2003a). *Data and models for statistical parsing with Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh.

Hockenmaier, J. (2003b). Parsing with generative models of predicate-argument structure. In *Proc. of ACL*, pages 359–366, Morristown, NJ, USA. Association for Computational Linguistics.

Hockenmaier, J. (2006). Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proc. of ACL*, pages 505–512, Morristown, NJ, USA. Association for Computational Linguistics.

Hockenmaier, J. and Steedman, M. (2002). Generative models for statistical parsing with combinatory categorial grammar. In *Proc. of ACL*, pages 335–342, Morristown, NJ, USA. Association for Computational Linguistics.

Hockenmaier, J. and Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Huang, L., Knight, K., and Joshi, A. (2006). A syntax-directed translator with extended domain of locality. In *Proc. of Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8.

Joshi, A. K. (2004). Domains of locality. *Data Knowl. Eng.*, 50(3):277–289.

Joshi, A. K., Vijay-Shanker, K., and Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In Sells, P., Shieber, S., and Wasow, T., editors, *Foundational Issues in Natural Language Processing*, chapter 2, pages 31–81. MIT Press, Cambridge, MA.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proc. of MT Summit*.

Koehn, P. and Hoang, H. (2007). Factored translation models. In *Proc. of EMNLP-CoNLL*, pages 868–876.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL Demo and Poster Sessions*, pages 177–180.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proc. of HLT-NAACL*, pages 127–133.

Lewis, P. M. I. and Stearns, R. E. (1968). Syntax-directed transductions. *Journal of the ACM*, 15:465–488.

Li, Z., Callison-Burch, C., Dyer, C., Khudanpur, S., Schwartz, L., Thornton, W., Weese, J., and Zaidan, O. (2009). Joshua: An open source toolkit for parsing-based machine translation. In *Proc. of WMT*, pages 135–139, Athens, Greece. Association for Computational Linguistics.

Liu, Y., Liu, Q., and Lin, S. (2006). Tree-to-string alignment template for statistical machine translation. In *Proc. of ACL-COLING*, pages 609–616.

Lopez, A. (2007). Hierarchical phrase-based translation with suffix arrays. In *Proc. of EMNLP-CoNLL*, pages 976–985.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Marcu, D., Wang, W., Echihabi, A., and Knight, K. (2006). SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. of EMNLP*, pages 44–52.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):314–330.

Melamed, I. D., Satta, G., and Wellington, B. (2004). Generalized multitext grammars. Technical Report 04-003, New York University.

Menezes, A. and Quirk, C. (2007). Using dependency order templates to improve generality in translation. In *Proc. of WMT*, pages 1–8, Prague, Czech Republic. Association for Computational Linguistics.

Mi, H. and Huang, L. (2008). Forest-based Translation Rule Extraction. In *Proc. of EMNLP*, pages 206–214.

Nasr, A. and Rambow, O. (2004). Supertagging and full parsing. In *Proc. of TAG*, pages 56–63, Vancouver, Canada.

Nesson, R., Shieber, S., and Rush, E. (2006). Induction of probabilistic synchronous tree-insertion grammars for machine translation. In *Proc. of AMTA*, pages 128–137, Cambridge.

Post, M. and Gildea, D. (2008). Parsers as language models for statistical machine translation. In *Proc. of AMTA*. AMTA.

Quirk, C. and Menezes, A. (2006). Do we need phrases? Challenging the conventional wisdom in statistical machine translation. In *Proc. of HLT-NAACL*, pages 8–16.

Rambow, O. and Satta, G. (1996). Synchronous models of language. In *Proc. of ACL*, pages 116–123, Morristown, NJ, USA. Association for Computational Linguistics.

Riezler, S. and Maxwell, J. T. (2006). Grammatical machine translation. In *Proc. of HLT-NAACL*, pages 248–255, Morristown, NJ, USA. Association for Computational Linguistics.

Roark, B. (2001). Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Rozenberg, G. and Salomaa, A., editors (1997). *Handbook of formal languages, vol. 3: beyond words*. Springer-Verlag New York, Inc., New York, NY, USA.

Schabes, Y. (1992). Stochastic lexicalized tree-adjoining grammars. In *Proc. of CL*, pages 425–432, Morristown, NJ, USA. Association for Computational Linguistics.

Shen, L., Xu, J., and Weischedel, R. (2008). A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. of ACL-HLT*, pages 577–585, Columbus, Ohio. Association for Computational Linguistics.

Shieber, S. (2007). Probabilistic synchronous tree-adjoining grammars for machine translation: The argument from bilingual dictionaries.

Shieber, S. M. and Schabes, Y. (1990). Synchronous tree-adjoining grammars. In *Proc. of COLING*, pages 253–258.

Steedman, M. (2000). *The syntactic process*. MIT Press, Cambridge, MA.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proc. Int. Conf. Spoken Language Processing (ICSLP 2002)*.

Taskar, B., Klein, D., Collins, M., Koller, D., and Manning, C. (2004). Max-margin parsing. In *Proc. of EMNLP*, pages 1–8.

Venugopal, A. and Zollmann, A. (2006). Syntax augmented machine translation via chart parsing with integrated language modeling. Technical report, Carnegie Mellon University.

Venugopal, A., Zollmann, A., and Vogel, S. (2007). An efficient two-pass approach to synchronous-CFG driven statistical MT. In *Proc. of HLT-NAACL*.

Wang, C., Collins, M., and Koehn, P. (2007). Chinese syntactic reordering for statistical machine translation. In *Proc. of EMNLP-CoNLL*, pages 737–745.

Wellington, B., Waxmonsky, S., and Melamed, I. D. (2006). Empirical lower bounds on the complexity of translational equivalence. In *Proc. of ACL-COLING*, pages 977–984.

Wu, D. (1996). A polynomial-time algorithm for statistical machine translation. In *Proc. of ACL*, pages 152–158.

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

Xiong, D., Liu, Q., and Lin, S. (2006). Maximum entropy based phrase reordering model for statistical machine translation. In *Proc. of ACL-COLING*, pages 521–528.

Xu, P., Chelba, C., and Jelinek, F. (2001). A study on richer syntactic dependencies for structured language modeling. In *Proc. of ACL*, pages 191–198, Morristown, NJ, USA. Association for Computational Linguistics.

Yamada, K. (2003). *A syntax-based statistical translation model.* PhD thesis, University of Southern California, Los Angeles, CA, USA.

Zens, R. and Ney, H. (2006). Discriminative reordering models for statistical machine translation. In *Proc. of WMT*, pages 55–63, New York City. Association for Computational Linguistics.

Zhang, H. and Gildea, D. (2008). Efficient multi-pass decoding for synchronous context free grammars. In *Proc. of ACL-HLT*, pages 209–217, Columbus, Ohio. Association for Computational Linguistics.

Zhang, H., Huang, L., Gildea, D., and Knight, K. (2006). Synchronous binarization for machine translation. In *Proc. of HLT-NAACL*, pages 256–263.

Zhang, M., Jiang, H., Aw, A., Li, H., Tan, C. L., and Li, S. (2008). A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In *Proc. of ACL-HLT*, pages 200–208, Columbus, Ohio. Association for Computational Linguistics.

Zollmann, A. and Venugopal, A. (2006). Syntax augmented machine translation via chart parsing. In *Proc. of NAACL Workshop on Statistical Machine Translation*, pages 138–141, New York, NY, USA.

Zollmann, A., Venugopal, A., Och, F., and Ponte, J. (2008). A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proc. of COLING*, pages 1145–1152, Manchester, UK.