# CCG Syntactic Reordering Models for Phrase-based Machine Translation

**Dennis N. Mehay**
The Ohio State University
Columbus, OH, USA
mehay@ling.ohio-state.edu

**Chris Brew**
Educational Testing Service
Princeton, NJ, USA
cbrew@ets.org

## Abstract

Statistical phrase-based machine translation requires no linguistic information beyond word-aligned parallel corpora (Zens et al., 2002; Koehn et al., 2003). Unfortunately, this linguistic agnosticism often produces ungrammatical translations. *Syntax*, or sentence structure, could provide guidance to phrase-based systems, but the "non-constituent" word strings that phrase-based decoders manipulate complicate the use of most recursive syntactic tools. We address these issues by using Combinatory Categorial Grammar, or CCG, (Steedman, 2000), which has a much more flexible notion of constituency, thereby providing more labels for putative non-constituent multiword translation phrases. Using CCG parse charts, we train a syntactic analogue of a lexicalized reordering model by labelling phrase table entries with multiword labels and demonstrate significant improvements in translating between Urdu and English, two language pairs with divergent sentence structure.

## 1 Introduction

Statistical phrase-based machine translation (PMT) is attractive, as it requires no linguistic information beyond word-aligned parallel corpora (Zens et al., 2002; Koehn et al., 2003). Unfortunately, this linguistic agnosticism leaves phrase-based systems with no precise characterization of the word order relationships between languages, often leading to ungrammatical translations. Syntax could provide guidance to phrase-based systems, by steering them towards reorderings that reflect the structural relationships between languages, but using syntax to guide a phrase-based system is problematic. Phrase-based systems build the result incrementally from the beginning of the target string to the end, and the intermediate strings need not constitute complete traditional syntactic constituents. It is difficult to reconcile traditional recursive syntactic processing with this regime, because not all intermediate strings considered by the decoder would even have a syntactic category to assess. As a result, most phrase-based decoders control reordering using simple distance-based distortion models, which penalize all reordering equally, and lexicalized reordering models (Tillmann, 2004; Axelrod et al., 2005), which probabilistically score various reordering configurations conditioned on specific lexical translations. While undoubtedly better than nothing, these models perform poorly when languages diverge considerably in sentence structure. Distance-based distortion models are too coarse-grained to distinguish correct from incorrect reordering, while lexical reordering models suffer from data sparsity and fail to capture more general patterns. We argue that finding a way to label translation phrases with syntactic labels will abstract over the observed reordering configurations thereby address both all three deficiencies of granularity, data sparsity and lack of generality.

The present work presents a novel syntactic analogue of the lexicalized reordering model that uses multiword syntactic labels to capture the general reordering patterns between two languages with very different word order. We accomplish this by using Combinatory Categorial Grammar, or CCG (Steed-



*Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 210–221,
Montréal, Canada, June 7-8, 2012. 

man, 2000), a word-centered syntax that allows a great deal of flexibility in how sentence analyses are formed. Syntactic derivations in CCG are massively *spuriously ambiguous*, i.e., there are many ways to derive the same semantic analysis of a sentence, similar to how a mathematical equation can be reduced by canceling out variables in different orders. Despite its name, spurious ambiguity is a benefit to us, as it provides many different labelled bracketings for the same dependency graph of the same sentence, thereby increasing the chance that any substring of that sentence will have a syntactic label. Our approach exploits this property of CCG to derive multiword CCG syntactic labels for target translation strings in a phrase table, thus providing a firmer basis on which to collect syntactic reordering statistics. In particular:

- We show how CCG can derive constituent labels for target-side phrase-table entries that are often lamented as "non-constituents" or as "crossing a phrase boundary".

- Our CCG categories are not limited to single-word *supertags*. Rather, as these labels are drawn from CCG parse charts, they can span multiple words. Further, the labels are tailored specifically to each translation constituent's boundaries (Section 2.1). As a consequence, $\approx$70% of phrase table entries receive a single syntactic label (Section 5), largely removing the terminological inconsistency of calling lexical translation constituents "phrases". Now, more of them actually are syntactic phrases.

- We use these labels to train a target-language bidirectional reordering model over CCG syntactic sequences (Section 3), which, when added to the baseline system, is found to be superior to systems that use both lexicalized reordering models and supertag reordering models (Section 5).

With only minor modifications, we incorporate these enhancements into a state-of-the-art PMT decoder (Koehn et al., 2007), achieving significant improvements over two competitive baselines in an Urdu-English translation task (Sections 5). This language pair was chosen to highlight the promise of this approach for languages with considerable, but syntactically governed, word-order differences to one another. Finally, in a small discussion we provide qualitative evidence that the improvements in automatic metric scores correspond to real gains in target language fluency.

## 2 Syntax, Constituency and Phrase-based MT

Consider the following German-English PMT phrase pair that we have extracted from a parallel European parliamentary transcript:[1]

Ich hoffe, daß $\Leftrightarrow$ I hope that

Neither word string is a well-formed constituent in traditional theories of syntax. But tradition is at odds with the intuition that that such "non-constituent" sequences are still well-formed substrings, governed by rules of how they can be combined with other word strings — e.g., declarative sentence translation rules like es möglich sein wird $\Leftrightarrow$ it will be possible can grammatically extend each, but a noun phrase rule cannot.

As Figure 1 illustrates, putative non-constituent word sequences abound in phrase-based MT. Here a translation "phrase" is simply any contiguous word string that is consistent with a word alignment (a relation between source and target words), usually produced by a language-independent alignment procedure (Zens et al., 2002). The figure also highlights the need for linguistic syntax in controlling how translations are assembled; the successful translation is merely one among many possible reorderings, many of which (despite their ungrammaticality) might score well on a word $n$-gram model. But rather than changing the word alignments or PMT "phrase" boundaries to fit a syntactic theory, we choose to use a flexible syntax which can produce a wider range of bracketings to accommodate the results of alignment-derived translations. To this end, we use Combinatory Categorial Grammar, or CCG, (Steedman, 2000). To understand how CCG allows this, we illustrate its use with some simple examples.

[1] Throughout this paper, the term "PMT phrase" refers to an unbroken sequence of words used by a PMT system, whereas "phrase" (without context) refers to a syntactic constituent.

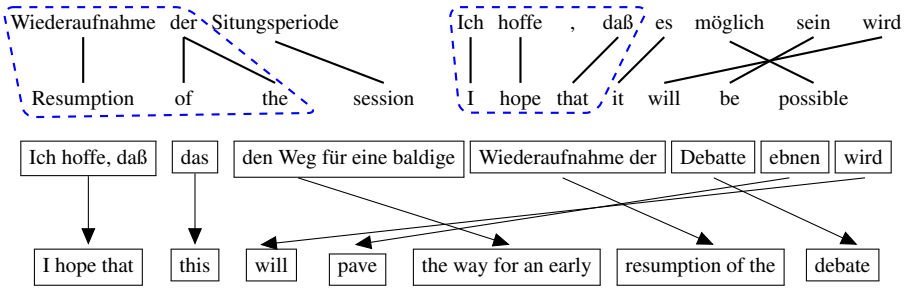



Figure 1: Two phrase-based MT word groups are extracted from aligned words (the dashed outlines) and then used to form a new translation (bottom). [Adapted from parallel sentences in the Europarl German-English corpus, v6.]

### 2.1 CCG, Spurious Ambiguity and PMT: Turning "Phrases" into Phrases

CCG is a derivational syntax, where words are assigned a *lexical category*[2] and sentence structures are then recursively built using a small set of deductive rule schemata known as *combinators* (Steedman, 2000). Lexical syntactic categories can be richly structured in CCG, indicating how words can combine. A syntactic category of the form $X/Y$, e.g., states that a category of type $X$ can be formed if combined with a $Y$ to its right — i.e., a function from rightward $Y$s to $X$. This can be accomplished with the *forward function application* combinator ($>$),[3] which is written in derivational form as follows:[4]

$$\frac{X/Y \quad Y}{X}>$$

This derivation of the symbol $X$ is known as the *normal-form* derivation (Steedman, 2000), since it uses function application whenever possible. But CCG has the ability to construct the same result by using a different, *non-normal-form* sequence of combinatory inferences. For example, by using the *backward type-raising* combinator ($\mathbf{T}_<$) and then *backward function application* ($<$), we can arrive at the same result:

$$\frac{X/Y \qquad \dfrac{Y}{X\backslash(X/Y)}\mathbf{T}_<}{X}<$$

This derivation shows how the argument $Y$ to the functional type $X/Y$[5] can "raise" its type to become a function that consumes that functional type, $X\backslash(X/Y)$, only to produce same result as before, namely $X$. This property of CCG is often referred to as "spurious ambiguity", because there are many ways of reaching the same result as the canonical, normal-form derivation.

Despite the name, this property is useful for our purposes. Considering the target translation in Figure 1, we then observe in Figure 2 how CCG can derive not only a bracketing similar to a more traditional Penn Treebank-style parse, but also a non-normal-form variant that gives us a single category for the English translation string I hope that — namely the category $S[dcl]/S[dcl]$ (a declarative sentence lacking a declarative sentence complement to its right).

We use this fact about CCG to label a wider range of PMT phrases with genuine syntactic constituent labels. First we parse the English sentences in our training data with the C&C parser, a state-of-the-art, treebank-trained CCG parser (Clark and Curran, 2007), producing normal-form CCG derivations. We then enumerate all non-normal-form derivations that result in the same top-level symbol, packing all derivations (normal-form and non-normal-form) into a parse chart (see Figure 4).

[2] When represented by a strings, lexical categories are called *supertags*.

[3] CCG actually respects the rule-to-rule hypothesis (Bach, l976), where, for every syntactic term built, there is a corresponding semantic term, but, for simplicity of exposition, we focus only on syntax here.

[4] The reader will notice that CCG derivations are in fact trees, but that they "grow" in the direction opposite to how parse trees are often depicted in NLP.

[5] Also referred to as a *functor*.

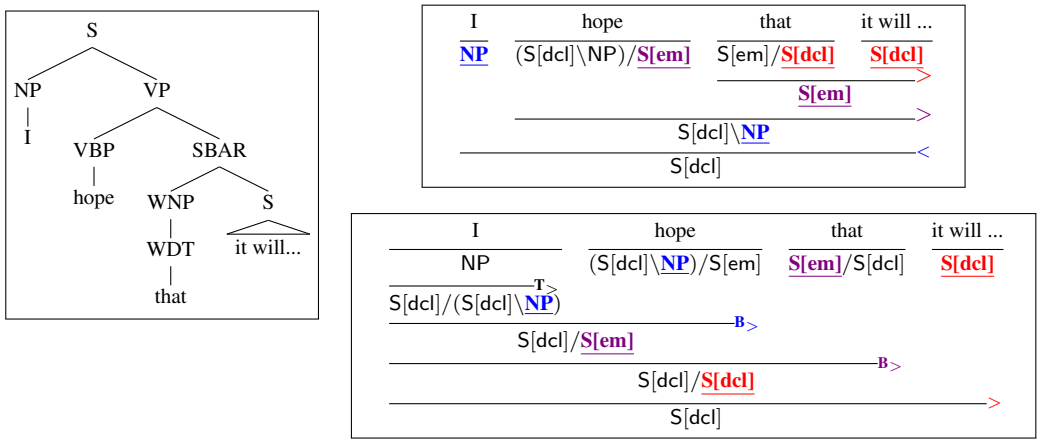



Figure 2: *Left:* a traditional syntactic derivation; *top right:* a normal-form CCG derivation with the same subject+predicate bracketing; *bottom right:* one of many non-normal-form variants. Combinator symbol key: >=forward function application, <=backward function application, $\mathbf{T}_{>}$=forward type-raising, $\mathbf{B}_{>}$=forward composition. Note: the CCG dependencies that are discharged in different orders are indicated by color-coding (if available in your medium) and underlining the appropriate categories (type-raising discharges no dependencies). Both CCG derivations lead to the same symbol (S[dcl]), and dependencies.

| | UR.-EN. |
|---|---|
| SINGLE-LABEL COVERAGE | 69% |
| AVE. EN. PHRASE LEN. | 2.8 wds |
| AVE. CCG LABEL SPAN | 2.3 wds |
| AVE. CCG LABS/ENTRY | 1.4 |

Table 1: Training data statistics (top to bottom): (1) % of single CCG labels spanning entire English translation phrases, (2) average length of English translation phrase, (3) average CCG label span and (4) average CCG labels per English translation phrase. (Maximum translation phrase length is 7 words.)

For the English string of each phrase table entry, we inspect the chart for the English-side sentence that it came from and extract a list of labels as in Figure 3. For each span, this procedure either (lines 5–9) finds the topmost single label, only using type-raised categories when no others exist,[6] or (lines 10–19) recursively and greedily finds the longest spanning labels from left to right, if no single label exists. The degenerate case is the single-word level (supertags). In this way we find single labels for 69% of the English-side phrase training instances. Table 1 gives more details.

[6]Type-raisings are almost always possible, and will always be closer to the top-level symbol. Many type-raisings, however, are superfluous – i.e., produce no novel bracketings. Therefore we only use type-raised symbols to derive a label for a span of words when necessary.

```
GETLABELS(C,s)
1   ▷ C: a packed chart of derivations of E
2   ▷ s = (e_l, e_r): a span in target sentence E
3   ▷ RETURN: a list of labels covering all words
4   ▷         from E in span s
5   if EXISTSSINGLESPANNINGLABEL(C,s)
6     then ▷ Get the topmost label
7          ▷ non-type-raised, if possible
8          lb ← GETTOPMOSTLABEL(C,s)
9          return [ lb ]
10    else ▷ Get the longest label starting at e_l
11         for i ← (e_r − 1) to (e_l + 1)
12         do lbs ← GETLABELS(C,(e_l, i))
13            if LENGTH(lbs)=1
14              then e_l' ← i + 1
15                   lb ← HEAD(lbs)
16                   BREAK
17              else CONTINUE
18         return
19           CONS(lb,GETLABELS(C,(e_l', e_r)))
```

Figure 3: Algorithm for labeling English sides of phrase table instances.

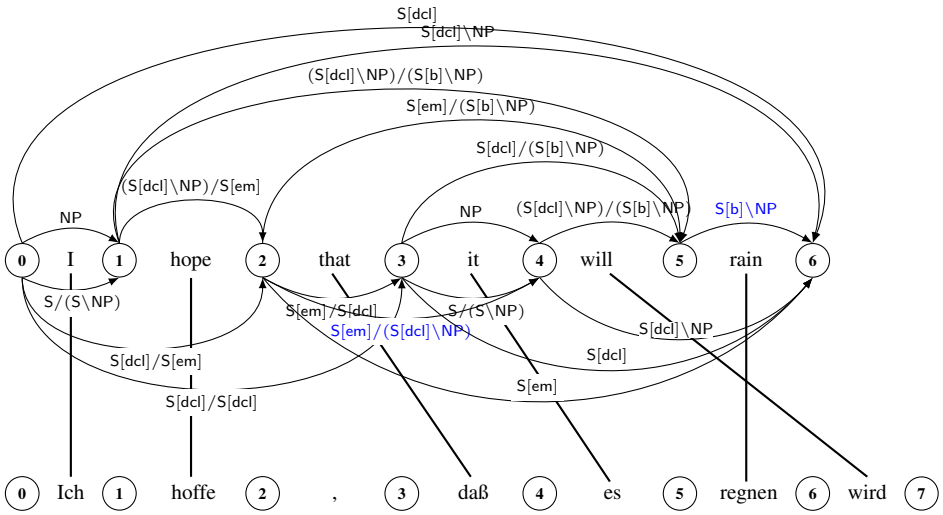



Figure 4: A packed CCG parse chart with multiple semantically equivalent derivations and two word-aligned strings. (Not all derivations are depicted.)

## 3 Reordering Models: from Words to Supertags to Parses

In phrase-based MT systems, the standard reordering model that controls the order in which the source string is translated is the lexicalized reordering model (Tillmann, 2004; Axelrod et al., 2005). In its simplest form, a lexicalized reordering model estimates, for each translation phrase pair $(\mathbf{f}_{i...j}, \mathbf{e}_{k...l})$ (where the indices sit "in-between" words, as in Figure 4), the probability of $p(\text{O} \mid \mathbf{f}_{i...j}, \mathbf{e}_{k...l})$, where $\text{O} \in \{\text{MONO}, \text{SWAP}, \text{DISCONTINUOUS}\}$ (abbreviated M, S and D) is the orientation of the phrase pair $(\mathbf{f}_{i...j}, \mathbf{e}_{k...l})$ w.r.t. the previously translated source phrase $\mathbf{f}_{u...v}$. If $v = i$, then $\text{O} = \text{M}$; if $u = j$, then $\text{O} = \text{S}$; otherwise $\text{O} = \text{D}$. This model, known as a *unidirectional* MSD lexicalized reordering model, can also be enriched with statistics over orientations to the next source phrase translated (i.e., it can be a *bidirectional* model), as well as with more fine-grained distinctions in the third class D (i.e., whether it is $\text{D}_{\text{LEFT}}$ or $\text{D}_{\text{RIGHT}}$). All models in the present work are bidirectional MSD models.

During decoding, orientations are predicted based on previously translated (or following) phrases in the decoder's search state, but, when extracting orientation statistics, there are many different possible phrasal segmentations of both strings. A simple solution, known as *word-based extraction*, is to look for neighboring alignment points that support the various orientations. In Figure 4, e.g., a word-based extraction regime would count the phrase $\boxed{\text{hoffe}} \Leftrightarrow \boxed{\text{hope}}$ as being in orientation D w.r.t. to what follows, because its rightmost index, 2, is discontiguous with the next aligned source point, (3,4). Another approach, known as *phrase-based extraction* aims to remedy this situation by conditioning the extraction of orientations on translation phrases consistent with the alignment. In Figure 4 there is a translation phrase that follows the phrase in question — *viz.*, $\boxed{\text{, daß}} \Leftrightarrow \boxed{\text{that}}$ — and an orientation of M is therefore tallied.

Regardless of the method of extraction, lexicalized reordering model statistics rely on exact word-string pairs, $(\mathbf{f}, \mathbf{e})$, which can lead problems with data sparsity. Moreover, even given ample data, cross-phrasal reordering generalizations will be missed. E.g., the fact that $\boxed{\text{regnen}} \Leftrightarrow \boxed{\text{rain}}$ has orientation S w.r.t. the previous phrase pair does not support the fact that other infinitival German verbs should also behave similarly in relative clausal environments.

To remedy this we might substitute abstract symbols for each word in $\mathbf{e}$, and train a syntactic bidirectional MSD reordering model. For this we use CCG supertags (cf. the single-word labels in the parse

chart in Figure 4), which are richly structured parts of speech that describe their potential to combine with other words (cf. Section 2.1). Given the same phrase from Figure 4, we can estimate the probability of orientation S, given $\boxed{\text{regnen}} \Leftrightarrow \boxed{\text{S[b]\NP}}$. A further level of abstraction is to use CCG parse charts packed with all derivations. The phrase $\boxed{\text{daß es}} \Leftrightarrow \boxed{\text{that it}}$ can therefore be abstracted to $\boxed{\text{daß es}} \Leftrightarrow \boxed{\text{S[em]/(S[dcl]\NP)}}$ (a "that" clause lacking a verb phrase to the right).

Except in cases of high ambiguity, the source phrase effectively encodes the target phrase, meaning that these extensions will suffer from data sparsity similarly to the baseline lexicalized model. We therefore omit the source phrase in our syntactic reordering models, estimating probability distributions $p(\text{O}|\text{LAB}(\mathbf{e}))$ where $\text{LAB}(\mathbf{e})$ is the syntactic label sequence derived from the chart (or supertagged string, as the case may be) using the algorithm in Figure 3.[7] Orientations are determined using the phrase-based extraction regime described in (Tillmann, 2004), but statistics are tallied only for the syntactic label sequence of the target string. More precisely, for phrase pair $(\mathbf{f}_{i...j}, \mathbf{e}_{k...l})$, if a phrase $(\mathbf{f}_{a...i}, \mathbf{e}_{b...k})$ exists in the alignment grid, an orientation of M is assigned to $\text{LAB}(\mathbf{e}_{k...l})$ . Otherwise, if a phrase $(\mathbf{f}_{j...p}, \mathbf{e}_{l...m})$ exists in the alignment grid, an orientation of S is assigned. In all other cases, an orientation of D is assigned.

Using these statistics, we deploy target-side reordering models, as described below.

## 4 Related Work

As noted, lexicalized reordering models can be trained and configured in many different ways. In addition to the standard word-based extraction (Axelrod et al., 2005) and phrase-based extraction (Tillmann, 2004) cases, more recent work has explored using dynamic programming to extract and later score orientations based on *hierarchical configurations* of phrases consistent with an alignment (Galley and Manning, 2008). This means that the reordering model can be conditioned on an unbounded amount of context and can capture the fact that many translations are monotonic w.r.t. the previously translated block, but are mistakenly identified as having orientation S or D.

Su and colleagues (2010) observe that the space of phrase pairs consistent with an alignment can be viewed in its entirety, as a *graph* of phrases, thereby collecting reordering statistics w.r.t. the entire space of surrounding phrases. Ling and colleagues (2011) extend this approach by weighting orientation counts with multiple scored alignments. All of these more sophisticated reordering extraction approaches are compatible with the current approach, and could be straightforwardly applied to our labelled target-side word strings.

Syntax-driven reordering approaches in phrase-based MT abound, but, perhaps due to the incompatibility of phrase table entries and traditional syntactic constituency, most research has avoided using recursive target-side syntax during decoding. Tillmann (2008) presents an algorithm that reorders using part-of-speech based permutation patterns during the decoding process. Others have side-stepped the issue by restructuring the source language *before decoding* to resemble the target language using syntactic rules, either automatically extracted (Xia and McCord, 2004), or hand-crafted (Collins et al., 2005; Wang et al., 2007; Xu and Seneff, 2008).

The flexibility of CCG syntax is also gaining recognition as a useful tool for constraining statistical MT decoders. Hassan (2009) describes an incremental CCG parsing language model, although his model does not beat a supertag factored PMT approach. Almaghout and colleagues (2010) also use a CCG chart to improve translation, augmenting SCFG rules by consulting the multiple derivations in the parse chart of Clark and Curran's (2007) CCG parser. We note two key differences to our use of spurious ambiguity. First, they use a chart packed with *multiple* dependency analyses, unlike our spuriously ambiguous reworkings of the parser's *single*-best analysis. Second, the C&C parser restrains type-raising to a small number of possibilities, thereby blocking many non-normal-form derivations that we do not.

Two SCFG approaches that employ categorial syntax that resembles CCG are the *syntax-augmented MT* (SAMT) system described in (Venugopal et al., 2007), and the target dependency lan-

[7] Note that a tagged string can be viewed as a very impoverished parse chart, and so the algorithm defined in Figure 3 can be applied to the supertagging case as well.

guage model of of (Shen et al., 2008). (Venugopal et al., 2007) uses a Penn Treebank-trained CFG parser to label target strings and then reworks the CFG parse trees, if needed,x to account for non-traditional constituents. This on-demand reworking process, however, is bounded by tree depth, and sometimes produces conjoined categories, rather than consistently produce the functional "slash" categories that a full CCG would — e.g., a $\boxed{\text{subject + transitive verb}}$ string might sometimes be labelled $\boxed{\mathsf{NP}+\mathsf{V}}$ and other times $\boxed{\mathsf{S/NP}}$. The approach in (Shen et al., 2010) uses a simple categorial grammar with only a single atomic symbol — i.e., every functional category has the form $\mathsf{C}\backslash\mathsf{X}$ or $\mathsf{C}/\mathsf{X}$, where $\mathsf{X}$ is either $\mathsf{C}$ or another slash category $\mathsf{C}\backslash\mathsf{X}$ or $\mathsf{C}/\mathsf{X}$. In contrast to these two approaches, the CCG parser we use is trained on a CCG treebank that is the result of a carefully engineered Penn Treebank-to-CCG conversion (Hockenmaier and Steedman, 2007) and we impose no limits on deriving categorial functional categories ($\mathsf{X}/\mathsf{Y}$). We view our reworking of CCG charts as a potentially useful extension to such approaches.

## 5 Experimental Results

We empirically validate our technique by translating from Urdu into English. Urdu has a canonical word order of SOV — subject, object(s), verb — whereas English has SVO, leading to indefinitely long distances between corresponding verbs and objects. This language pair is therefore a strong test case for a reordering model.

For decoding we use Moses (Koehn et al., 2007), a state-of-the-art PMT decoder, with IRST LM (Federico and Cettolo, 2007) for language model inference. For Urdu-English parallel data, we use the OpenMT 2008 training set which consists of 88 thousand sentence-level translations and a translation dictionary of ≈114 thousand word and phrase translations. We use half of the OpenMT 2008 Urdu-English evaluation data for development and perform development testing on the other half. Both halves are ≈900 sentences long and were balanced to contain approximately the same number of tokens. Our blind test set is the entire OpenMT 2009 Urdu-English evaluation set. All evaluation sets had 4 reference translations for each tuning or testing instance. All system component weights were tuned using minimum error-rate training (Och, 2003), with three tuning runs for each condition. The data was normalized, tokenized and the English sentences were lowercased,[8]

As a baseline, we train a standard phrase-based system with a bidirectional MSD lexicalized reordering model using word-based extraction. Our CCG-augmented reordering system has all of the model components of the baseline, as well as a bidirectional orientation reordering model over target-side multiword syntactic labels. To directly test the effect of using CCG parse charts — as opposed to simply using a CCG supertagger — we also added a CCG supertag bidirectional MSD reordering model to the baseline set-up. All systems were tuned and tested with distortion limit of 15 words, and test runs were performed with and without 200-best minimum Bayes' risk (MBR) hypothesis selection (Kumar and Byrne, 2004).

To acquire CCG labels for our English parallel data, we use the C&C CCG toolkit of Clark and Curran (2007). We build CCG parse charts by reworking the normal-form derivations from the C&C parser in all spuriously ambiguous ways, as described in Section 2.1. For supertags, we tag with the C&C supertagger. Rather than training separate phrase tables for our CCG systems, however, we instead decorate the baseline phrase tables with CCG multiword labels or supertags. To smooth over parsing and tagging errors, we only use those labels whose relative frequency (rf) is sufficiently high w.r.t. the most frequent label for that phrase pair $\textsc{Lab}^*_{[\mathrm{f}\Leftrightarrow\mathrm{e}]}$. More precisely, for each phrase pair, we use the set of labels:[9]

$$\{\textsc{Lab}_{[\mathrm{f}\Leftrightarrow\mathrm{e}]} | \mathrm{rf}(\textsc{Lab}_{[\mathrm{f}\Leftrightarrow\mathrm{e}]}) \geq \beta \cdot \mathrm{rf}(\textsc{Lab}^*_{[\mathrm{f}\Leftrightarrow\mathrm{e}]})\}$$

This is reminiscent of the $\beta$-best tagging approach of (Clark and Curran, 2004), but performed in a batch process when creating the syntactic phrase tables (both supertag and CCG chart-derived). We set

[8] N.B. We use Penn Treebank III-compatible tokenization for English and a specially designed tokenization script for Urdu, cf. (Baker et al., 2010), Appendix C

[9] Recalling that ≈31% of the time, a phrase pair might have a list of labels, rather than a single label, the word 'label' here refers to a single token that can be the concatenation of multiple symbols.

| | DEVTEST (NIST-08) (MBR/NON-MBR) | | | | NIST-09 TEST (MBR/NON-MBR) | | | |
|---|---|---|---|---|---|---|---|---|
| | BLEU-4 | METEOR | TER | LENGTH | BLEU-4 | METEOR | TER | LENGTH |
| LR | 25.3/24.7 | 28.3/28.2 | 64.2/64.4 | 98.2/97.6 | 29.1/28.8 | 30.0/28.8 | 60.0/60.1 | **98.2**/97.8 |
| NO-LR | 22.5/22.1 | 27.5/27.3 | 66.3/66.3 | 97.6/97.1 | 26.2/25.8 | 29.2/29.1 | 61.9/62.0 | 97.1/96.6 |
| ST+LR | 24.5/24.2 | 28.4/28.3 | 64.6/64.5 | 97.9/97.3 | 28.5/28.2 | 30.0/**30.0** | 60.3/60.2 | 97.9/97.3 |
| CCG+LR | **25.6/25.2** | **28.7/28.5** | 64.3/64.5 | **98.7/98.1** | 29.1/**29.2** | 30.1/**30.2** | **59.5/59.8** | 97.4/**97.9** |

Table 2: Case-insensitive BLEU-4, METEOR, TER and hypothesis/reference length ratio (LENGTH) for a lexicalized reordering baseline (LR), a system with only a distance-based distortion model (NO-LR), a system with an additional CCG supertag reordering model (ST+LR) and our system with an additional CCG chart-derived reordering model (CCG+LR). Systems were run with (left of slash) and without (right of slash) 200-best-list MBR hypothesis selection. All boldfaced results were found to be significantly better than the baseline at $\geq$ the 95% confidence level using method described in (Clark et al., 2011) with 3 separate MERT tuning runs for each system. Non-boldfaced numbers are statistically indistinguishable from (or worse than) the baseline.

$\beta = 0.5$ in all of our CCG experiments.

To minimize disruption to the Moses decoder (which only supports single-word labels in phrase-based mode), we project multiword labels across the words they label as single-word factors with book-keeping characters, similar to the "microtag" annotations of asynchronous factored translation models (Cettolo et al., 2008). We modified to the decoder to reassemble the multiple single-word factors into a single label before querying the reordering model. As an example, we might have the phrase pair le vélo rouge $\Leftrightarrow$ the|NP( red|NP+ bike|NP). Before querying the reordering model, the factor sequence NP( NP+ NP) is collapsed into the single, multiword label 'NP' by the rule schema `X( ... X+ ... X)` $\rightarrow$ `X`.

We train a language model using all of the WMT 2011 NEWSCRAWL, NEWSCOMENTARY and EUROPARL monolingual data,[10] tokenized and lowercased as above, but de-duplicated to address the redundancy of the Web-crawled portion of that data set. We also train a separate language model on the English portion of the Urdu-English parallel corpus (minus the dictionary entries), and interpolate the two models by optimizing perplexity on our tuning set.

Table 2 lists our results, where we see significant improvement over both of our baselines, lexicalized reordering (LR) and supertag reordering plus lexicalized reordering (ST+LR). To test the effects of the lexicalized reordering model itself, we also evaluate a system with no lexicalized reordering model (only a distance-based distortion model). This last system (a system which almost always prefers not to reorder) is considerably worse than all other systems, demonstrating the need for non-monotonic reordering configurations when accounting for the Urdu-English data.

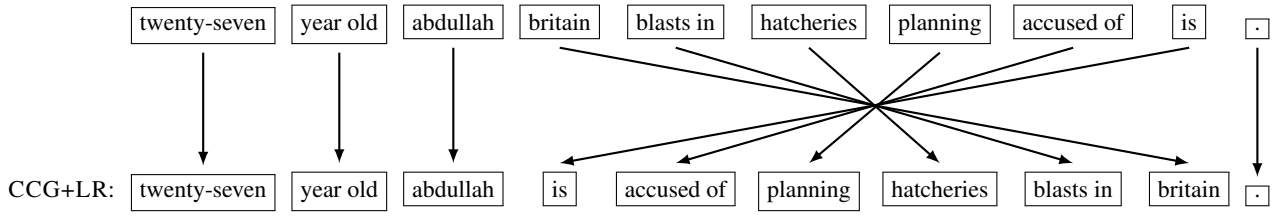
[10] `http://www.statmt.org/wmt11/translation-task.html`

## 6 Analysis and Discussion

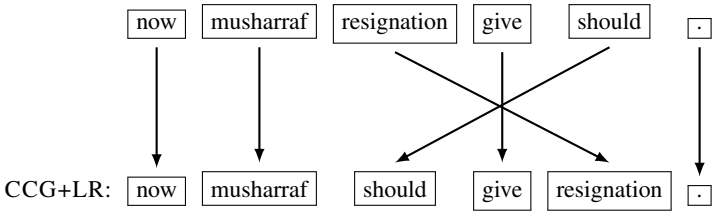
Our CCG system (CCG+LR) outperforms both baseline systems (LR and ST+LR) in a majority of metrics in both MBR and non-MBR conditions. We see that, even though MBR decoding closes the performance gap somewhat, our system continues to match or outperform (if sometimes insignificantly) in all areas. Note that the CCG+LR non-MBR configuration outperforms both LR and ST+LR in MBR and non-MBR decoding conditions in its METEOR score on the NIST-09 test set. We note also that, in the NIST-09 test case, the CCG+LR system's poorer performance is perhaps due to a mismatch in hypothesis length, which could be harming its scores, particularly the BLEU brevity penalty.

### 6.1 Poor Performance of CCG Supertag Model

We have no firm explanation for the poor performance of the CCG supertag model (ST-LR), but it is important to note that the supertag reordering model does not unify statistics across phrases of different lengths, as the CCG chart-derived model does. E.g., the phrase pair den Weg für eine $\Leftrightarrow$ the way for an will query the CCG chart-derived reordering model with the same symbol as the phrase pair den Weg für eine baldige $\Leftrightarrow$ the way for an early




LR: twenty-seven years on charges of planning bombings hatcheries , abdullah in britain .
Reference 1: 27 years old abdullah is accused of planning explosions in britain .
Reference 2: twenty-seven years old abdullah is blamed for planning attacks in britain .
Reference 3: abdullah , 27 , has been blamed for planning the blasts in britain .
Reference 4: abdullah , 27 , has been blamed for planning the blasts in britain .




LR: now musharraf resignation should be given .
Reference 1: now musharraf should resign .
Reference 2: now , musharraf should resign .
Reference 3: now , musharraf should resign .
Reference 4: musharraf should resign .

Figure 5: Sample devtest (NIST-08) translations of the median-performing tuned CCG syntactic reordering model (CCG+LR) compared to the median-performing baseline lexicalized reordering model (LR).

— *viz.*, NP/N. The CCG supertag model, however, will have two distinct label sequences for these phrases — *viz.*, NP/N_N_(NP\NP)/NP_NP/N and NP/N_N_(NP\NP)/NP_NP/N_N/N, resp. — both of which could be reduced to the single label, NP/N, using CCG's syntactic combinators. The supertag system does not have the means of relating the reordering patterns of strings of symbols such as this.[11] Such data fragmentation may be leading to decreased performance, which would indicate the use of *recursive* CCG syntax.

### 6.2 Qualitative Improvements

In addition to improved metric scores, we noted real qualitative improvements in some examples, as Figure 5 shows. These examples demonstrate the ability of the reordering model to navigate the massive, structure-governed reorderings needed to approximate the correct answer with the phrase inventory it is given.

[11] Its reordering table has more than twice as many entries as that of the chart-derived model.

### 6.3 Comparison to the State of the Art

To our knowledge, the state of the art in Urdu-English translation using the OpenMT data is listed in the NIST OpenMT 2009 evaluation results (`http://www.itl.nist.gov/iad/mig/tests/mt/2009/ResultsRelease/currentUrdu.html`). This evaluation accepted only single system outputs, and used cased references. Therefore we had to choose a single system output and recase its text.

For system selection, we picked the tuned system that performed best on the development test set. For recasing, we trained a lowercased-to-cased monolingual phrase-based "translation model" with no reordering and a cased language model, similar to what is described in (Baker et al., 2010). The training text is simply the non-dictionary portion of the Urdu-English parallel corpus, with its lowercased version as the source and the original cased text as the target, both halves tokenized as above. We tuned on a similar version of the English half of our tuning

references. The lowercased output of our system is fed to this model and the first token of each casing "translation" is capitalized (if not already).

The official metric of the NIST 2009 evaluation is BLEU (as implemented in the NIST-distributed `mteval-v13a.pl` script).[12] The best-performing system in the constrained data evaluation scored **0.312** w.r.t. the cased references, with the second and third place systems scoring **0.2395** and **0.2322**, respectively.[13] Our best performing MERT-tuned system (as determined on the devtest data) scores **0.2734** on the test set, putting it between the top two systems. For comparison, our devtest-best baseline LR system scores **0.2683** on the test set.

While is generally not useful to test experimental manipulations based on a single tuning run (Clark et al., 2011) and with different monolingual language modelling data, we note these figures simply to situate our results within the state of the art.

## 7 Conclusion

We have argued for the use of CCG in phrase-based translation, due to its flexibility in providing a wealth of different bracketings that better accommodate lexical translation strings. We have also presented a novel method for using CCG constituent labels in a syntactic reordering model where the syntactic labels span multiple words, do not cross translation constituent boundaries and are tailored specifically to each translation constituent. The result is a significant improvement in Urdu-English (SOV $\rightarrow$ SVO) translation scores over two baselines: a traditional phrase-based baseline with a lexicalized reordering model and a phrase-based baseline with an additional supertag reordering model. Moreover, we have provided qualitative examples that confirm the improvements in automatic metrics.

In future work we would like explore whether further improvements can be gained by using more sophisticated reordering models, such as reordering graphs (Su et al., 2010) and hierarchical reordering models (Galley and Manning, 2008) both for our word-based and syntactic reordering models. Further, as in prior work (Zollmann et al., 2006; Shen et al., 2010; Almaghout et al., 2010), our categorial labels could also be used to derive CCG-augmented SCFG rules, both lexicalized and unlexicalized, cf. (Zhao and Al-onaizan, 2008) — the latter being the SCFG analogue of our current model.

[12] `ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a-20091001.tar.gz`.

[13] We exclude combination entries that are combinations of multiple systems with different algorithmic approaches.


## Acknowledgments

The authors would like to thank Chong Min Lee, Aoife Cahill and Nitin Madnani at ETS for taking the time to read earlier drafts of this (and closely related) work. Their comments and suggestions made this a better paper. We would also like to thank the anonymous reviewers for their very helpful feedback. The views expressed in this paper do not necessarily reflect those of The Ohio State University or of Educational Testing Service.


## References


Hala Almaghout, Jie Jiang, and Andy Way. 2010. CCG Augmented Hierarchical Phrase-based Machine Translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation*, Paris, France.

Amittai Axelrod, Ra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT-05)*, Pittsburgh, PA, USA.

Emmon Bach. l976. An Extension of Classical Transformational Grammar. In *Proceedings of the 1976 Conference on Problems of Linguistic Metatheory*, pages 183–224, East Lansing, MI, USA.

Kathy Baker, Steven Bethard, Michael Bloodgood, Ralf Brown, Chris Callison-Burch, Glen Coppersmith, Bonnie Dorr, Wes Filardo, Kendall Giles, Anni Irvine, Mike Kayser, Lori Levin, Justin Martineau, Jim Mayeld, Scott Miller, Aaron Phillips, Andrew Philpot, Christine Piatko, Lane Schwartz, and David Zajic. 2010. Semantically informed machine translation. Technical Report 002, Johns Hopkins University, Baltimore, MD, Human Language Technology Center of Excellence.

Mauro Cettolo, Marcello Federico, Daniele Pighin, and Nicola Bertoldi. 2008. Shallow-syntax Phrase-based Translation: Joint versus Factored String-to-chunk Models. In *Proceedings of AMTA 2008*, Honolulu, HI, USA.

Stephen Clark and James R. Curran. 2004. The Importance of Supertagging for Wide-Coverage CCG Parsing. In *Proceedings of the 20th International Con-*

*ference on Computational Linguistics (COLING-04)*, Geneva, Switzerland.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493–552.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL-11)*, Portland, OR, USA.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI, USA.

Marcello Federico and Mauro Cettolo. 2007. Efficient Handling of $n$-gram Language Models for Statistical Machine Translation. In *Proceedings of Association for Computational Linguistics*, Prague, The Czech Republic.

Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of EMNLP-08*.

Hany Hassan. 2009. *Lexical Syntax for Statistical Machine Translation*. Ph.D. thesis, Dublin City University, Dublin, Ireland.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of NAACL-HLT*, pages 48–54, Edmonton, Alberta, CA.

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

Shankar Kumar and William Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of HLT-NAACL*.

Wang Ling, Jo ao Graça, David Martins de Matos, Isabel Trancoso, and Alan Black. 2011. Discriminative Phrase-based Lexicalized Reordering Models using Weighted Reordering Graphs. In *Proceedings of the $5^{th}$ International Joint Conference on Natural Language Processing*.

Franz Joseph Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the $41^{st}$ Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A New String-to-dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proceedings of the Joint Meeting of the Association for Computational Linguistics and Human Language Technologies (ACL-08:HLT)*, Columbus, OH, USA.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-Dependency Statistical Machine Translation. *Computational Linguistics*, 36(4):649–671.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Jinsong Su, Yang Liu, Yajuan Lü, Haitao Mi, and Qun Liu. 2010. Learning Lexicalized Reordering Models from Reordering Graphs. In *Proceedings of the ACL 2010; Short Papers*.

Christoph Tillmann. 2004. A Unigram Orientation Model for Statistical Machine Translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04.

Christoph Tillmann. 2008. A Rule-Driven Dynamic Programming Decoder for Statistical MT. In *Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation (SSST-08)*.

Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. 2007. An Efficient Two-pass Approach to Synchronous-CFG Driven Statistical MT. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL-07)*, Rochester, NY.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese Syntactic Reordering for Statistical Machine Translation. In *Proceedings of EMNLP/CoNLL-07*, Prague, The Czech Republic.

Fei Xia and Michael McCord. 2004. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. In *Proceedings of International Conference on Computational Linguistics (COLING-04)*, Geneva, Switzerland.

Yushi Xu and Stephanie Seneff. 2008. Two-stage Translation: A Combined Linguistic and Statistical Machine Translation Framework. In *Proceedings of the $8^{th}$ Conference of the Association for Machine Translation in the Americas (AMTA-08)*, Waikiki, Honolulu, HI, USA.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-Based Statistical Machine Translation. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *KI-2002: Advances in Artificial Intelligence, Proceedings of the $25^{th}$ Annual German Conference on AI, (KI-2002)*, pages 18–32. Springer Verlag, Aachen, Germany.

Bing Zhao and Yaser Al-onaizan. 2008. Generalizing Local and Non-Local Word-Reordering Patterns for Syntax-Based Machine Translation. In *Proceedings of The Conference on Empirical Methods in Natural Language Processig (EMNLP-08)*.

Andreas Zollmann, Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2006. The CMU-UKA Syntax Augmented Machine Translation System for IWSLT-06. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT-06)*, Kyoto, Japan.