Improved Translation with Source Syntax Labels

Hieu Hoang
School of Informatics
University of Edinburgh
h.hoang@sms.ed.ac.uk

Philipp Koehn
School of Informatics
University of Edinburgh
pkoehn@inf.ed.ac.uk

Abstract

We present a new translation model that include undecorated hierarchical-style phrase rules, decorated source-syntax rules, and partially decorated rules.

Results show an increase in translation performance of up to 0.8% BLEU for German–English translation when trained on the news-commentary corpus, using syntactic annotation from a source language parser. We also experimented with annotation from shallow taggers and found this increased performance by 0.5% BLEU.

1 Introduction

Hierarchical decoding is usually described as a formally syntactic model without linguistic commitments, in contrast with syntactic decoding which constrains rules and production with linguistically motivated labels. However, the decoding mechanism for both hierarchical and syntactic systems are identical and the rule extraction are similar.

Hierarchical and syntax statistical machine translation have made great progress in the last few years and can claim to represent the state of the art in the field. Both use synchronous context free grammar (SCFG) formalism, consisting of rewrite rules which simultaneously parse the input sentence and generate the output sentence. The most common algorithm for decoding with SCFG is currently CKY+ with cube pruning works for both hierarchical and syntactic systems, as implemented in Hiero (Chiang, 2005), Joshua (Li et al., 2009), and Moses (Hoang et al., 2009)

Rewrite rules in hierarchical systems have general applicability as their non-terminals are undecorated, giving hierarchical system broad coverage. However, rules may be used in inappropriate situations without the labeled constraints. The general applicability of undecorated rules create spurious ambiguity which decreases translation performance by causing the decoder to spend more time sifting through duplicate hypotheses. Syntactic systems makes use of linguistically motivated information to bias the search space at the expense of limiting model coverage.

This paper presents work on combining hierarchical and syntax translation, utilizing the high coverage of hierarchical decoding and the insights that syntactic information can bring. We seek to balance the generality of using undecorated non-terminals with the specificity of labeled non-terminals. Specifically, we will use syntactic labels from a source language parser to label non-terminal in production rules. However, other source span information, such as chunk tags, can also be used.

We investigate two methods for combining the hierarchical and syntactic approach. In the first method, syntactic translation rules are used concurrently with a hierarchical phrase rules. Each ruleset is trained independently and used concurrently to decode sentences. However, results for this method do not improve.

The second method uses one translation model containing both hierarchical and syntactic rules. Moreover, an individual rule can contain both decorated syntactic non-terminals, and undecorated hierarchical-style non-terminals (also, the left-hand-side non-terminal may, or may not be decorated). This results in a 0.8% improvement over the hierarchical baseline and analysis suggest that long-range ordering has been improved.

We then applied the same methods but using linguistic annotation from a chunk tagger (Abney, 1991) instead of a parser and obtained an improvement of 0.5% BLEU over the hierarchical baseline, showing that gains with additional source-side annotation can be obtained with simpler tools.

2 Past Work

Hierarchical machine translation (Chiang, 2005) extends the phrase-based model by allowing the use of non-contiguous phrase pairs ('production rules'). It promises better re-ordering of translation as the reordering rules are an implicit part of the translation model. Also, hierarchical rules follow the recursive structure of the sentence, reflecting the linguistic notion of language.

However, the hierarchical model has several limitations. The model makes no use of linguistic information, thus creating a simple model with broad coverage. However, (Chiang, 2005) also describe heuristic constraints that are used during

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 409–417,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

rule extraction to reduce spurious ambiguity. The resulting translation model does reduces spurious ambiguity but also reduces the search space in an arbitrary manner which adversely affects translation quality.

Syntactic labels from parse trees can be used to annotate non-terminals in the translation model. This reduces incorrect rule application by restricting rule extraction and application. However, as noted in (Ambati and Lavie, 2008) and elsewhere,the naïve approach of constraining every non-terminal to a syntactic constituent severely limits the coverage of the resulting grammar, therefore, several approaches have been used to improve coverage when using syntactic information.

Zollmann and Venugopal (2006) allow rules to be extracted where non-terminals do not exactly span a target constituent. The non-terminals are then labeled with complex labels which amalgamates multiple labels in the span. This increase coverage at the expense of increasing data sparsity as the non-terminal symbol set increases dramatically. Huang and Chiang (2008) use parse information of the source language, production rules consists of source tree fragments and target languages strings. During decoding, a packed forest of the source sentence is used as input, the production rule tree fragments are applied to the packed forest. Liu et al. (2009) uses joint decoding with a hierarchical and tree-to-string model and find that translation performance increase for a Chinese-English task. Galley et al. (2004) creates minimal translation rules which can explain a parallel sentence pair but the rules generated are not optimized to produce good translations or coverage in any SMT system. This work was extended and described in (Galley et al., 2006) which creates rules composed of smaller, minimal rules, as well as dealing with unaligned words. These measures are essential for creating good SMT systems, but again, the rules syntax are strictly constrained by a parser.

Others have sought to add soft linguistic constraints to hierarchical models using addition feature functions. Marton and Resnik (2008) add feature functions to penalize or reward non-terminals which cross constituent boundaries of the source sentence. This follows on from earlier work in (Chiang, 2005) but they see gains when finer grain feature functions which different constituency types. The weights for feature function is tuned in batches due to the deficiency of MERT when presented with many features. Chiang et al. (2008) rectified this deficiency by using the MIRA to tune all feature function weights in combination. However, the translation model continues to be hierarchical.

Chiang et al. (2009) added thousands of linguistically-motivated features to hierarchical and syntax systems, however, the source syntax features are derived from the research above. The translation model remain constant but the parameterization changes.

Shen et al. (2009) discusses soft syntax constraints and context features in a dependency tree translation model. The POS tag of the target head word is used as a soft constraint when applying rules. Also, a source context language model and a dependency language model are also used as features.

Most SMT systems uses the Viterbi approximation whereby the derivations in the log-linear model is not marginalized, but the maximum derivation is returned. String-to-tree models build on this so that the most probable derivation, including syntactic labels, is assumed to the most probable translation. This fragments the derivation probability and the further partition the search space, leading to pruning errors. Venugopal et al. (2009) attempts to address this by efficiently estimating the score over an equivalent unlabeled derivation from a target syntax model.

Ambati and Lavie (2008); Ambati et al. (2009) notes that tree-to-tree often underperform models with parse tree only on one side due to the non-isomorphic structure of languages. This motivates the creation of an isomorphic backbone into the target parse tree, while leaving the source parse unchanged.

3 Model

In extending the phrase-based model to the hierarchical model, non-terminals are used in translation rules to denote subphrases. Hierarchical non-terminals are undecorated so are unrestricted to the span they cover. In contrast, SCFG-based syntactic models restrict the extraction and application of non-terminals, typically to constituency spans of a parse tree or forest. Our soft syntax model combine the hierarchical and source-syntactic approaches, allowing translation rules with undecorated and decorated non-terminals with information from a source language tool.

We give an example of the rules extracted from an aligned sentence in Figure 1, with a parse tree on the source side.

Lexicalized rules with decorated non-terminals are extracted, we list five (non-exhaustive) examples below.

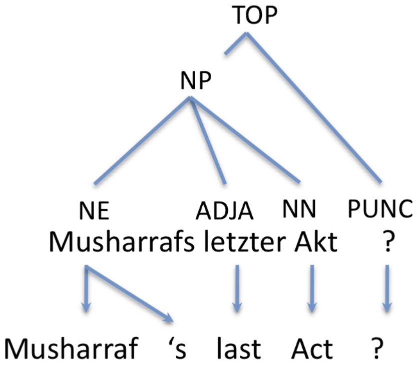

Figure 1: Aligned parsed sentence

$$\begin{array}{lcl} NP & \rightarrow & Musharrafs\ letzter\ Akt \\ & & \#\ Musharraf\,'s\ Last\ Act \\ NP & \rightarrow & NE_1\ letzter\ Akt\ \#\ X_1\ Last\ Act \\ NP & \rightarrow & NE_1\ ADJA_2\ Akt\ \#\ X_1\ X_2\ Act \\ NP & \rightarrow & NE_1\ letzter\ NN_2\ \#\ X_1\ Last\ X_2 \\ TOP & \rightarrow & NE_1\ ADJA_2\ Akt\ ?\ \#\ X_1\ X_2\ Act\ ? \end{array}$$

Hierarchical style rules are also extracted where the span doesn't exactly match a parse constituent. We list 2 below.

$$\begin{array}{lcl} X & \rightarrow & letzter\ Akt\ \#\ Last\ Act \\ X & \rightarrow & letzter\ X_1\ \#\ Last\ X_1 \end{array}$$

Unlexicalized rules with decorated non-terminals are also extracted:

$$\begin{array}{lcl} TOP & \rightarrow & NP_1\ PUNC_2\ \#\ X_1\ X_2 \\ NP & \rightarrow & NE_1\ ADJA_2\ NN_3\ \#\ X_1\ X_2\ X_3 \end{array}$$

Rules are also extracted which contains a mixture of decorated and undecorated non-terminals. These rules can also be lexicalized or unlexicalized. A non-exhaustive sample is given below:

$$\begin{array}{lcl} X & \rightarrow & ADJA_1\ Akt\ \#\ X_1\ Act \\ NP & \rightarrow & NE_1\ X_2\ \#\ X_1\ X_2 \\ TOP & \rightarrow & NE_1\ letzter\ X_2\ \#\ X_1\ Last\ X_2 \end{array}$$

At decoding time, the parse tree of the input sentence is available to the decoder. Decorated non-terminals in rules must match the constituent span in the input sentence but the undecorated X symbol can match any span.

Formally, we model translation as a string-to-string translation using a synchronous CFG that constrain the application of non-terminals to matching source span labels. The source words and span labels are represented as an unweighted word lattice, $< V, E >$, where each edge in the lattice correspond to a word or non-terminal label over the corresponding source span. In the soft syntax experiments, edges with the default source label, X, are also created for all spans. Nodes in the lattice represent word positions in the sentence.

We encode the lattice in a chart, as described in (Dyer et al., 2008). A chart is is a tuple of 2-dimensional matrices $< F, R >$. $F_{i,j}$ is the word or non-terminal label of the j^{th} transition starting word position i. $R_{i,j}$ is the end word position of the node on the right of the j^{th} transition leaving word position i.

The input sentence is decoded with a set of translation rules of the form

$$X \rightarrow < \alpha L_s, \gamma, \sim >$$

where α and γ and strings of terminals and non-terminals. L_s and the string α are drawn from the same source alphabet, Δ_s. γ is the target string, also consisting of terminals and non-terminals. $\sim$ is the one-to-one correspondence between non-terminals in α and γ. L_s is the left-hand-side of the source. As a string-to-string model, the left-hand-side of the target is always the default target non-terminal label, X.

Decoding follows the CKY+ algorithms which process contiguous spans of the source sentence bottom up. We describe the algorithm as inference rules, below, omitting the target side for brevity.

Initialization

$$\frac{}{[X \rightarrow \bullet \alpha L_s, i, i]} \qquad (X \rightarrow \alpha L_s) \in G$$

Terminal Symbol

$$\frac{[X \rightarrow \alpha \bullet F_{j,k} \beta L_s, i, j]}{[X \rightarrow \alpha F_{j,k} \bullet \beta L_s, i, j+1]}$$

Non-Terminal Symbol

$$\frac{[X \rightarrow \alpha \bullet F_{j,k} \beta L_s, i, j] \qquad [X, j, R_{j,k}]}{[X \rightarrow \alpha F_{j,k} \bullet \beta L_s, i, R_{j,k}]}$$

Left Hand Side

$$\frac{[X \rightarrow \alpha \bullet L_s, i, R_{i,j}] \qquad [F_{i,j} = L_s]}{[X \rightarrow \alpha L_s \bullet, i, R_{i,j}]}$$

Goal

$$[X \rightarrow \alpha L_s \bullet, 0, |V| - 1]$$

This model allows translation rules to take advantage of both syntactic label and word context. The presence of default label edges between every node allows undecorated non-terminals to be applied to any span, allowing flexibility in the translation model.

This contrasts with the approach by (Zollmann and Venugopal, 2006) in attempting to improve the coverage of syntactic translation. Rather than creating ad-hoc schemes to categories non-terminals with syntactic labels when they do not span syntactic constituencies, we only use labels that are presented by the parser or shallow tagger. Nor do we try to expand the space where rules can apply by propagating uncertainty from the parser in building input forests, as in (Mi et al., 2008), but we build ambiguity into the translation rule.

The model also differs from (Marton and Resnik, 2008; Chiang et al., 2008, 2009) by adding informative labels to rule non-terminals and requiring them to match the source span label. The soft constraint in our model pertain not to a additional feature functions based on syntactic information, but to the availability of syntactic and non-syntactic informed rules.

4 Parameterization

In common with most current SMT systems, the decoding goal of finding the most probable target language sentence $\hat{\mathbf{t}}$, given a source language sentence **s**

$$\hat{\mathbf{t}} = \text{argmax}_{\mathbf{t}}\ p(\mathbf{t}|\mathbf{s}) \tag{1}$$

The argmax function defines the search objective of the decoder. We estimate $p(\mathbf{t}|\mathbf{s})$ by decomposing it into component models

$$p(\mathbf{t}|\mathbf{s}) = \frac{1}{Z} \prod_m h'_m(\mathbf{t}, \mathbf{s})^{\lambda_m} \tag{2}$$

where $h'_m(\mathbf{t}, \mathbf{s})$ is the feature function for component m and λ_m is the weight given to component m. Z is a normalization factor which is ignored in practice. Components are translation model scoring functions, language model, and other features.

The problem is typically presented in log-space, which simplifies computations, but otherwise does not change the problem due to the monotonicity of the log function ($h_m = \log\, h'_m$)

$$\log p(\mathbf{t}|\mathbf{s}) = \sum_m \lambda_m\, h_m(\mathbf{t}, \mathbf{s}) \tag{3}$$

An advantage of our model over (Marton and Resnik, 2008; Chiang et al., 2008, 2009) is the number of feature functions remains the same, therefore, the tuning algorithm does not need to be replaced; we continue to use MERT (Och, 2003).

5 Rule Extraction

Rule extraction follows the algorithm described in (Chiang, 2005). We note the heuristics used for hierarchical phrases extraction include the following constraints:

1. all rules must be at least partially lexicalized,
2. non-terminals cannot be consecutive,
3. a maximum of two non-terminals per rule,
4. maximum source and target span width of 10 word
5. maximum of 5 source symbols

In the source syntax model, non-terminals are restricted to source spans that are syntactic phrases which severely limits the rules that can be extracted or applied during decoding. Therefore, we can adapt the heuristics, dropping some of the constraints, without introducing too much complexity.

1. consecutive non-terminals are allowed
2. a maximum of three non-terminals,
3. all non-terminals and LHS must span a parse constituent

In the soft syntax model, we relax the constraint of requiring all non-terminals to span parse constituents. Where there is no constituency spans, the default symbol X is used to denote an undecorated non-terminal. This gives rise to rules which mixes decorated and undecorated non-terminals.

To maintain decoding speed and minimize spurious ambiguity, item (1) in the syntactic extraction heuristics is adapted to prohibit consecutive undecorated non-terminals. This combines the strength of syntactic rules but also gives the translation model more flexibility and higher coverage from having undecorated non-terminals. Therefore, the heuristics become:

1. consecutive non-terminals are allowed, but consecutive undecorated non-terminals are prohibited
2. a maximum of three non-terminals,
3. all non-terminals and LHS must span a parse constituent

5.1 Rule probabilities

Maximum likelihood phrase probabilities, $p(\bar{\mathbf{t}}|\bar{\mathbf{s}})$, are calculated for phrase pairs, using fractional counts as described in (Chiang, 2005). The maximum likelihood estimates are smoothed using Good-Turing discounting (Foster et al., 2006). A phrase count feature function is also create for each translation model, however, the lexical and backward probabilities are not used.

6 Decoding

We use the Moses implementation of the SCFG-based approach (Hoang et al., 2009) which support hierarchical and syntactic training and decoding used in this paper. The decoder implements the CKY+ algorithm with cube pruning, as well as histogram and beam pruning, all pruning parameters were identical for all experiments for fairer comparison.

All non-terminals can cover a maximum of 7 source words, similar to the maximum rule span feature other hierarchical decoders to speed up decoding time.

7 Experiments

We trained on the New Commentary 2009 corpus[1], tuning on a hold-out set. Table 1 gives more details on the corpus. *nc_test2007* was used for testing.

		German	English
Train	Sentences	82,306	
	Words	2,034,373	1,965,325
Tune	Sentences	2000	
Test	Sentences	1026	

Table 1: Training, tuning, and test conditions

The training corpus was cleaned and filtered using standard methods found in the Moses toolkit (Koehn et al., 2007) and aligned using GIZA++ (Och and Ney, 2003). Standard MERT weight tuning was used throughout. The English half of the training data was also used to create a trigram language model which was used for each experiment. All experiments use truecase data and results are reported in case-sensitive BLEU scores (Papineni et al., 2001).

The German side was parsed with the Bitpar parser[2]. 2042 sentences in the training corpus failed to parse and were discarded from the training for both hierarchical and syntactic models to ensure that train on identical amounts of data. Similarly, 991 out of 1026 sentences were parsable in the test set. To compare like-for-like, the baseline translates the same 991 sentences, but evaluated over 1026 sentences. (In the experiments with chunk tags below, all 1026 sentences are used).

#	Model	% BLEU
	Using parse tree	
1	Hierarchical	15.9
2	Syntax rules	14.9
3	Joint hier. + syntax rules	16.1
4	Soft syntax rules	16.7
	Using chunk tags	
5	Hierarchical	16.3
6	Soft syntax	16.8

Table 2: German–English results for hierarchical and syntactic models, in %BLEU

We use as a baseline the vanilla hierarchical model which obtained a BLEU score of 15.9% (see Table 2, line 1).

7.1 Syntactic translation

Using the naïve translation model constrained with syntactic non-terminals significantly decreases translation quality, Table 2, line 2. We then ran hierarchical concurrently with the syntactic models, line 3, but see little improvement over the hierarchical baseline. However, we see a gain of 0.8% BLEU when using the soft syntax model.

7.2 Reachability

The increased performance using the soft syntax model can be partially explained by studying the effect of changes to the extraction and decoding algorithms has to the capacity of the translation pipeline. We run some analysis in which we trained the phrase models with a corpus of one sentence and attempt to decode the same sentence. Pruning and recombination were disabled during decoding to negate the effect of language model context and model scores.

The first thousand sentences of the training corpus was analyzed, Table 3. The hierarchical model successfully decode over half of the sentences while a translation model constrained by a source syntax parse tree manages only 113 sentences, illustrating the severe degradation in coverage when a naive syntax model is used.

Decoding with a hierarchical and syntax model jointly (line 3) only decode one extra sentence over the hierarchical model, suggesting that the expressive power of the hierarchical model almost

[1] http://www.statmt.org/wmt09/

[2] http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html

#	Model	Reachable sentences
1	Hierarchical	57.8%
2	Syntax rules	11.3%
3	Joint hier. + syntax rules	57.9%
4	Soft syntax rules	58.5%

Table 3: Reachability of 1000 training sentences: can they be translated with the model?

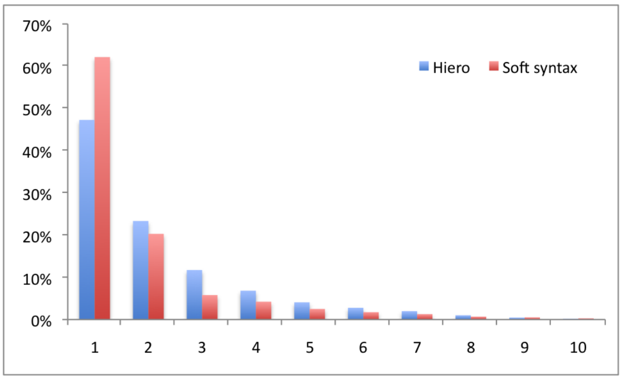

Figure 2: Source span lengths

completely subsumes that of the syntactic model. The MERT tuning adjust the weights so that the syntactic model is very rarely applied during joint decoding, suggesting that the tuning stage prefers the broader coverage of the hierarchical model over the precision of the syntactic model.

However, the soft syntax model slightly increases the reachability of the target sentences, lines 4.

7.3 Rule Span Width

The soft syntactic model contains rules with three non-terminals, as opposed to 2 in the hierarchical model, and consecutive non-terminals in the hope that the rules will have the context and linguistic information to apply over longer spans. Therefore, it is surprising that when decoding with a soft syntactic grammar, significantly more words are translated singularly and the use of long spanning rules is reduced, Figure 2.

However, looking at the usage of the glue rules paints a different picture. There is significantly less usage of the glue rules when decoding with the soft syntax model, Figure 3. The use of the glue rule indicates a failure of the translation model to explain the translation so the decrease in its usage is evidence of the better explanatory power of the soft syntactic model.

An example of an input sentence, and the best translation found by the hierarchical and soft syntax model can be seen in Table 4. Figure 4 is the parse tree given to the soft syntax model.

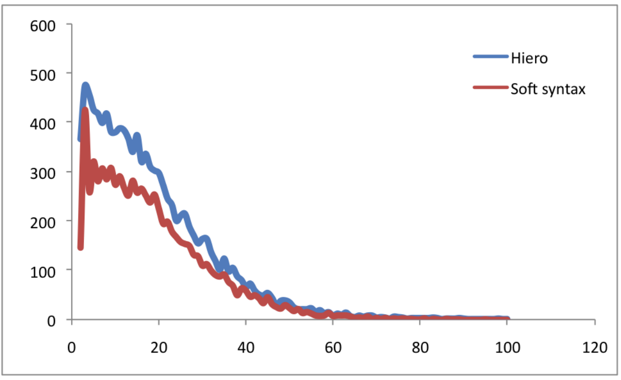

Figure 3: Length and count of glue rules used decoding test set

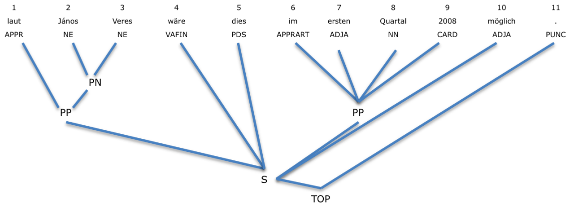

Figure 4: Example input parse tree

Input laut János Veres wäre dies im ersten Quartal 2008 möglich .
Hierarchical output according to János Veres this in the first quarter of 2008 would be possible .
Soft Syntax according to János Veres this would be possible in the first quarter of 2008 .

Table 4: Example input and best output found

Both output are lexically identical but the output of the hierarchical model needs to be reordered to be grammatically correct. Contrast the derivations produced by the hierarchical grammar, Figure 5, with that produced with the soft syntax model, Figure 6. The soft syntax derivation makes use of several non-lexicalized to dictate word order, shown below.

$$\begin{array}{lll} X & \rightarrow & NE_1\ NE_2\ \#\ X_1\ X_2 \\ X & \rightarrow & VAFIN_1\ PDS_2\ \#\ X_1\ X_2 \\ X & \rightarrow & ADJA_1\ NN_2\ \#\ X_1\ X_2 \\ X & \rightarrow & APPRART_1\ X_2\ CARD_3\ \#\ X_1\ X_2\ X_3 \\ X & \rightarrow & PP_1\ X_2\ PUNC_3\ \#\ X_2\ X_1\ X_3 \end{array}$$

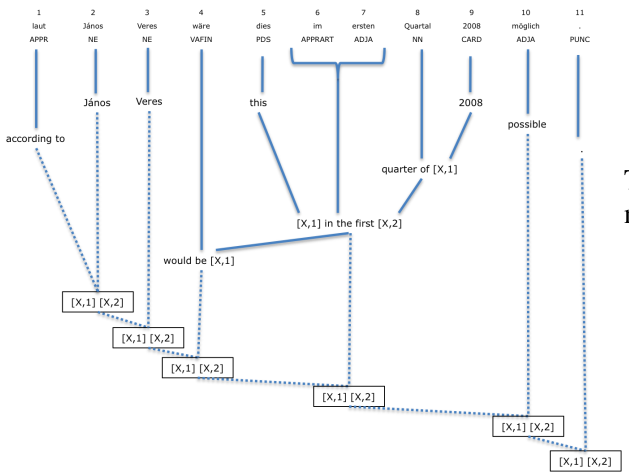

Figure 5: Derivation with Hierarchical model

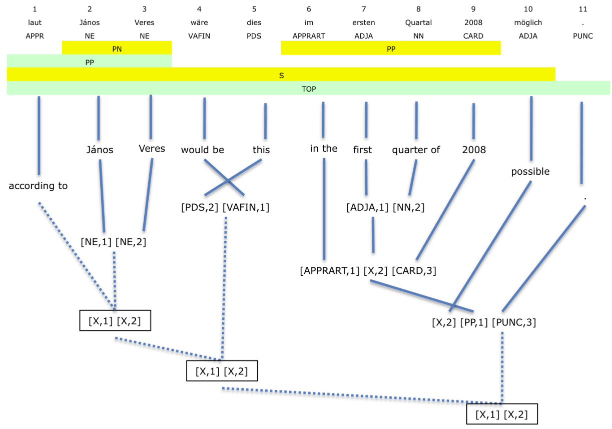

Figure 6: Derivation with soft syntax model

The soft syntax derivation include several rules which are partially decorated. Crucially, the last rule in the list above reorders the *PP* phrase and the non-syntactic phrase *X* to generate the grammatically correct output. The other non-lexicalized rules monotonically concatenate the output. This can be performed by the glue rule, but nevertheless, the use of empirically backed rules allows the decoder to better compare hypotheses. The derivation also rely less on the glue rules than the hierarchical model (shown in solid rectangles).

Reducing the maximum number of non-terminals per rule reduces translation quality but increasing it has little effect on the soft syntax model, Table 5. This seems to indicate that non-terminals are useful as context when applying rules up to a certain extent.

7.4 English to German

We experimented with the reverse language direction to see if the soft syntax model still increased translation quality. The results were positive but less pronounced, Table 6.

# non-terms	% BLEU
2	16.5
3	16.8
5	16.8

Table 5: Effect on %BLEU of varying number of non-terminals

#	Model	% BLEU
1	Hierarchical	10.2
2	Soft syntax	10.6

Table 6: English–German results in %BLEU

7.5 Using Chunk Tags

Parse trees of the source language provide useful information that we have exploited to create a better translation model. However, parsers are an expensive resource as they frequently need manually annotated training treebanks. Parse accuracy is also problematic and particularly brittle when given sentences not in the same domain as the training corpus. This also causes some sentences to be unparseable. For example, our original test corpus of 1026 sentences contained 35 unparsable sentences. Thus, high quality parsers are unavailable for many source languages of interest.

Parse forests can be used to mitigate the accuracy problem, allowing the decoder to choose from many alternative parses, (Mi et al., 2008).

The soft syntax translation model is not dependent on the linguistic information being in a tree structure, only that the labels identify contiguous spans. Chunk taggers (Abney, 1991) does just that. They offer higher accuracy than syntactic parser, are not so brittle to out-of-domain data and identify chunk phrases similar to parser-based syntactic phrases that may be useful in guiding reordering.

We apply the soft syntax approach as in the previous sections but replacing the use of parse constituents with chunk phrases.

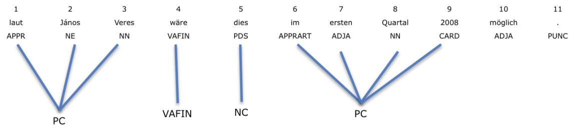

Figure 7: Chunked sentence

7.6 Experiments with Chunk Tags

We use the same data as described earlier in this chapter to train, tune and test our approach. The Treetagger chunker (Schmidt and Schulte im Walde, 2000) was used to tag the source (German) side of the corpus. The chunker successfully processed all sentences in the training and test dataset so no sentences were excluded. The increase training data, as well as the ability to translate all sentences in the test set, explains the higher hierarchical baseline than the previous experiments with parser data. We use the noun, verb and prepositional chunks, as well as part-of-speech tags, emitted by the chunker.

Results are shown in Table 2, line 5 & 6. Using chunk tags, we see a modest gain of 0.5% BLEU.

The same example sentence in Table 4 is shown with chunk tags in Figure 7. The soft syntax model with chunk tags produced the derivation tree shown in Figure 8. The derivation make use of an unlexicalized rule local reordering. In this example, it uses the same number of glue rule as the hierarchical derivation but the output is grammatically correct.

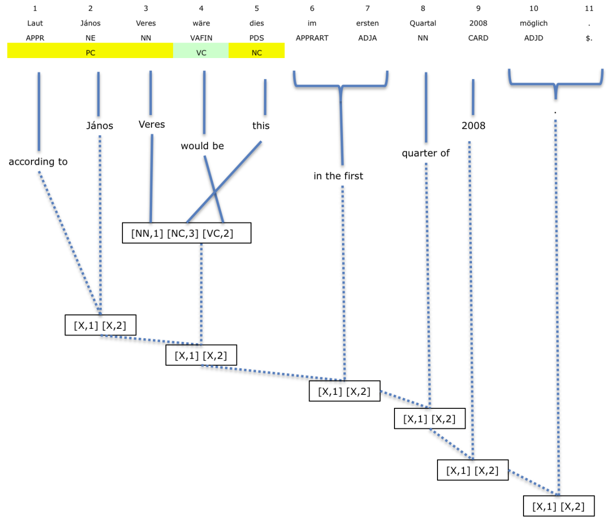

Figure 8: Translated chunked sentence

However, overall, the number of glue rules used shows the same reduction that we saw using soft syntax in the earlier section, as can be seen in Figure 9. Again, the soft syntax model, this time using chunk tags, is able to reduce the use of the glue rule with empirically informed rules.

8 Conclusion

We show in this paper that combining the generality of the hierarchical approach with the specificity of syntactic approach can improve translation. A reason for the improvement is the better long-range reordering made possible by the increase capacity of the translation model.

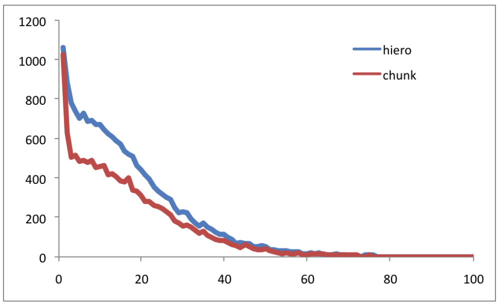

Figure 9: Chunk - Length and count of glue rules used decoding test set

Future work in this direction includes using tree-to-tree approaches, automatically created constituency labels, and back-off methods between decorated and undecorated rules.

9 Acknowledgement

This work was supported in part by the EuroMatrixPlus project funded by the European Commission (7th Framework Programme) and in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

References

Abney, S. (1991). Parsing by chunks. In *Robert Berwick, Steven Abney, and Carol Tenny: Principle-Based Parsing*. Kluwer Academic Publishers.

Ambati, V. and Lavie, A. (2008). Improving syntax driven translation models by re-structuring divergent and non-isomorphic parse tree structures. In *AMTA*.

Ambati, V., Lavie, A., and Carbonell, J. (2009). Extraction of syntactic translation models from parallel data using syntax from source and target languages. In *MT Summit*.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

Chiang, D., Knight, K., and Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado. Association for Computational Linguistics.

Chiang, D., Marton, Y., and Resnik, P. (2008). Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, Hawaii. Association for Computational Linguistics.

Dyer, C., Muresan, S., and Resnik, P. (2008). Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio. Association for Computational Linguistics.

Foster, G., Kuhn, R., and Johnson, H. (2006). Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia. Association for Computational Linguistics.

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 *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia. Association for Computational Linguistics.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Hoang, H., Koehn, P., and Lopez, A. (2009). A Unified Framework for Phrase-Based, Hierarchical, and Syntax-Based Statistical Machine Translation. In *Proc. of the International Workshop on Spoken Language Translation*, pages 152–159, Tokyo, Japan.

Huang, L. and Chiang, D. (2008). Forest-based translation rule extraction. In *EMNLP*, Honolulu, Hawaii.

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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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 *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece. Association for Computational Linguistics.

Liu, Y., Mi, H., Feng, Y., and Liu, Q. (2009). Joint decoding with multiple translation models. In *In Proceedings of ACL/IJCNLP 2009*, pages 576–584, Singapore.

Marton, Y. and Resnik, P. (2008). Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL-08: HLT*, pages 1003–1011, Columbus, Ohio. Association for Computational Linguistics.

Mi, H., Huang, L., and Liu, Q. (2008). Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL)*.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176(W0109-022), IBM Research Report.

Schmidt, H. and Schulte im Walde, S. (2000). Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Shen, L., Xu, J., Zhang, B., Matsoukas, S., and Weischedel, R. (2009). Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 72–80, Singapore. Association for Computational Linguistics.

Venugopal, A., Zollmann, A., Smith, N. A., and Vogel, S. (2009). Preference grammars: Softening syntactic constraints to improve statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–244, Boulder, Colorado. Association for Computational Linguistics.

Zollmann, A. and Venugopal, A. (2006). Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City. Association for Computational Linguistics.