Hierarchical Alignment Decomposition Labels for Hiero Grammar Rules

Gideon Maillette de Buy Wenniger
Institute for Logic,
Language and Computation
University of Amsterdam
Science Park 904, 1098 XH Amsterdam
The Netherlands
gemdbw AT gmail.com

Khalil Sima'an
Institute for Logic,
Language and Computation
University of Amsterdam
Science Park 904, 1098 XH Amsterdam
The Netherlands
k.simaan AT uva.nl

Abstract

Selecting a set of nonterminals for the synchronous CFGs underlying the hierarchical phrase-based models is usually done on the basis of a monolingual resource (like a syntactic parser). However, a standard bilingual resource like word alignments is itself rich with reordering patterns that, if clustered somehow, might provide labels of different (possibly complementary) nature to monolingual labels. In this paper we explore a first version of this idea based on a hierarchical decomposition of word alignments into recursive tree representations. We identify five clusters of alignment patterns in which the children of a node in a decomposition tree are found and employ these five as nonterminal labels for the Hiero productions. Although this is our first non-optimized instantiation of the idea, our experiments show competitive performance with the Hiero baseline, exemplifying certain merits of this novel approach.

1 Introduction

The Hiero model (Chiang, 2007; Chiang, 2005) formulates phrase-based translation in terms of a synchronous context-free grammar (SCFG) limited to the inversion transduction grammar (ITG) (Wu, 1997) family. While the original Hiero approach works with a single nonterminal label (X) (besides the start nonterminal S), more recent work is dedicated to devising methods for extracting more elaborate labels for the phrase-pairs and their abstractions into SCFG productions, e.g., (Zollmann and Venugopal, 2006; Li et al., 2012; Almaghout et al., 2011). All labeling approaches exploit monolingual parsers of some kind, e.g., syntactic, semantic or sense-oriented. The rationale behind monolingual labeling is often to make the probability distributions over alternative synchronous derivations of the Hiero model more sensitive to linguistically justified monolingual phrase context. For example, syntactic target-language labels in many approaches are aimed at improved target language modeling (fluency, cf. Hassan et al. (2007); Zollmann and Venugopal (2006)), whereas source-language labels provide suitable context for reordering (see Mylonakis and Sima'an (2011)). It is usually believed that the monolingual labels tend to stand for clusters of phrase pairs that are expected to be intersubstitutable, either syntactically or semantically (see Marton et al. (2012) for an illuminating discussion).

While we believe that monolingual labeling strategies are sound, in this paper we explore the complementary idea that the nonterminal labels could also signify *bilingual properties of the phrase pair*, particularly its characteristic *word alignment patterns*. Intuitively, an SCFG with nonterminal labels standing for alignment patterns should put more preference on synchronous derivations that mimic the word alignment patterns found in the training corpus, and thus, possibly allow for better reordering. It is important to stress the fact that these word alignment patterns are complementary to the monolingual linguistic patterns and it is conceivable that the two can be combined effectively, but this remains beyond the scope of this article.

The question addressed in this paper is how to select word alignment patterns and cluster them into bilingual nonterminal labels? In this paper we explore a first instantiation of this idea starting out from the following simplifying assumptions:

Proceedings of the 7th Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 19–28,
Atlanta, Georgia, 13 June 2013. ©2013 Association for Computational Linguistics

- The labels come from the word alignments only,
- The labels are coarse-grained, pre-defined clusters and not optimized for performance,
- The labels extend the binary set of ITG operators (monotone and inverted) into five such labels in order to cover non-binarizable alignment patterns.

Our labels are based on our own tree decompositions of word alignments (Sima'an and Maillette de Buy Wenniger, 2011), akin to Normalized Decomposition Trees (NDTs) (Zhang et al., 2008). In this first attempt we explore a set of five nonterminal labels that characterize alignment patterns found directly under nodes in the NDT projected for every word alignment in the parallel corpus during training. There is a range of work that exploits the monotone and inverted orientations of binary ITG within hierarchical phrase-based models, either as feature functions of lexicalized Hiero productions (Chiang, 2007; Zollmann and Venugopal, 2006), or as labels on non-lexicalized ITG productions, e.g., (Mylonakis and Sima'an, 2011). As far as we are aware, this is the first attempt at exploring a larger set of such word alignment derived labels in hierarchical SMT. Therefore, we expect that there are many variants that could improve substantially on our strong set of assumptions.

2 Hierarchical SMT models

Hierarchical SMT usually works with *weighted* instantiations of Synchronous Context-Free Grammars (SCFGs) (Aho and Ullman, 1969). SCFGs are defined over a finite set of nonterminals (start included), a finite set of terminals and a finite set of synchronous productions. A synchronous production in an SCFG consists of two context-free productions (source and target) containing the same number of nonterminals on the right-hand side, with a bijective (1-to-1 and onto) function between the source and target nonterminals. Like the standard Hiero model (Chiang, 2007), we constrain our work to SCFGs which involve at most two nonterminals in every lexicalized production.

Given an SCFG G, a source sentence $\mathbf{s}$ is translated into a target sentence $\mathbf{t}$ by synchronous derivations $\mathbf{d}$, each is a finite sequence of well-formed substitutions of synchronous productions from G, see (Chiang, 2006). Standardly, for complexity reasons, most models used make the assumption that the probability $P(\mathbf{t} \mid \mathbf{s})$ can be optimized through as single best derivation as follows:

$$\begin{aligned} \arg\max_{\mathbf{t}} P(\mathbf{t} \mid \mathbf{s}) &= \arg\max_{\mathbf{t}} \sum_{\mathbf{d} \in G} P(\mathbf{t}, \mathbf{d} \mid \mathbf{s}) && (1) \\ &\approx \arg\max_{\mathbf{d} \in G} P(\mathbf{t}, \mathbf{d} \mid \mathbf{s}) && (2) \end{aligned}$$

This approximation can be notoriously problematic for labelled Hiero models because the labels tend to lead to many more derivations than in the original model, thereby aggravating the effects of this assumption. This problem is relevant for our work and approaches to deal with it are Minimum Bayes-Risk decoding (Kumar and Byrne, 2004; Tromble et al., 2008), Variational Decoding (Li et al., 2009) and soft labeling (Venugopal et al., 2009; Marton et al., 2012; Chiang, 2010).

Given a derivation $\mathbf{d}$, most existing phrase-based models approximate the derivation probability through a linear interpolation of a finite set of feature functions ($\Phi(\mathbf{d})$) of the derivation $\mathbf{d}$, mostly working with local feature functions ϕ_i of individual productions, the target side yield string t of $\mathbf{d}$ (target language model features) and other heuristic features discussed in the experimental section:

$$\arg\max_{\mathbf{d} \in G} P(\mathbf{t}, \mathbf{d} \mid \mathbf{s}) \approx \arg\max_{\mathbf{d} \in G} \sum_{i=1}^{|\Phi(\mathbf{d})|} \lambda_i \times \phi_i \quad (3)$$

Where λ_i is the weight of feature ϕ_i optimized over a held-out parallel corpus by some direct error-minimization procedure like MERT (Och, 2003).

3 Baseline: Hiero Grammars (single label)

Hiero Grammars (Chiang, 2005; Chiang, 2007) are a particular form of SCFGs that generalize phrase-based translation models to hierarchical phrase-based Translation models. They allow only up to two (pairs of) nonterminals on the right-hand-side of rules. Hierarchical rules are formed from fully lexicalized base rules (i.e. phrase pairs) by replacing a sub-span of the phrase pair that corresponds itself to a valid phrase pair with variable X called "gap". Two gaps may be maximally introduced in this way[1], labeled as $X_{\boxed{1}}$ and $X_{\boxed{2}}$ respectively for distinction. The types of permissible Hiero rules are:

$$X \rightarrow \langle \alpha,\ \gamma \rangle \tag{4a}$$

$$X \rightarrow \langle \alpha\ X_{\boxed{1}} \beta,\ \delta\ X_{\boxed{1}} \zeta \rangle \tag{4b}$$

$$X \rightarrow \langle \alpha\ X_{\boxed{1}} \beta\ X_{\boxed{2}} \gamma\ ,\ \delta\ X_{\boxed{1}}\ \zeta\ X_{\boxed{2}} \eta\ \rangle \tag{4c}$$

$$X \rightarrow \langle \alpha\ X_{\boxed{1}} \beta\ X_{\boxed{2}} \gamma\ ,\ \delta\ X_{\boxed{2}}\ \zeta\ X_{\boxed{1}} \eta\ \rangle \tag{4d}$$

Here $\alpha, \beta, \gamma, \delta, \zeta, \eta$ are terminal sequences, possibly empty. Equation 4a corresponds to a normal phrase pair, 4b to a rule with one gap and 4c and 4d to the monotone- and inverting rules respectively.

An important extra constraint used in the original Hiero model is that rules must have at least one pair of aligned words, so that translation decisions are always based on some lexical evidence. Furthermore the sum of terminals and nonterminals on the source side may not be greater than five, and nonterminals are not allowed to be adjacent on the source side.

4 Alignment Labeled Grammars

Labeling the Hiero grammar productions makes rules with gaps more restricted about what broad categories of rules are allowed to substitute for the gaps. In the best case this prevents overgeneralization, and makes the translation distributions more accurate. In the worst case, however, it can also lead to too restrictive rules, as well as sparse translation distributions. Despite these inherent risks, a number of approaches based on syntactically inspired labels has succeeded to improve the state of the art by using monolingual labels, e.g., (Zollmann and Venugopal, 2006; Zollmann, 2011; Almaghout et al., 2011; Chiang, 2010; Li et al., 2012).

Unlabeled Hiero derivations can be seen as recursive compositions of phrase pairs. A single translation may be generated by different derivations (see equation 1), each standing for a choice of composition rules over a choice of a segmentation of the source-target sentence pair into a bag of phrase pairs. However, a synchronous derivation also induces an alignment between the different segments that it composes together. Our goal here is to label the Hiero rules in order to exploit aspects of the alignment that a synchronous derivation induces.

We exploit the idea that phrase pairs can be efficiently grouped into maximally decomposed trees (normalized decomposition trees – NDTs) (Zhang et al., 2008). In an NDT every phrase pair is recursively decomposed at every level into the *minimum number* of its phrase constituents, so that the resulting structure is maximal in that it contains the largest number of nodes. In Figure 1 left we show an example alignment and in Figure 1 right its associated NDT. The NDT shows pairs of source and target spans of (sub-) phrase pairs, governed at different levels of the tree by their parent node. In our example the root node splits into three phrase pairs, but these three phrase pairs together do not manage to cover the entire phrase pair of the parent because of the discontinuous translation structure ⟨owe, sind ... schuldig⟩. Consequently, a partially lexicalized structure with three children corresponding to phrase pairs and lexical items covering the words left by these phrase pairs is required.

During grammar extraction we determine an Alignment Label for every left-hand-side and gap of every rule we extract. This is done by looking at the NDT that decomposes their corresponding phrase pairs, and determining the complexity of the relation with their direct children in this tree. Complexity cases are ordered by preference, where the more simple label corresponding to the choice of maximal decomposition is preferred. We distinguish the following five cases, ordered by increasing complexity:

1. *Monotonic*: If the alignment can be split into two monotonically ordered parts.
2. *Inverted*: If the alignment can be split into two inverted parts.
3. *Permutation*: If the alignment can be factored as a permutation of more than 3 parts.[2]
4. *Complex*: If the alignment cannot be factored as a permutation of parts, but the phrase does contain at least one smaller phrase pair (i.e., it is composite).
5. *Atomic*: If the alignment does not allow the existence of smaller (child) phrase pairs.

[1] The motivation for this restriction to two gaps is mainly a practical computational one, as it can be shown that translation complexity grows exponentially with the number of gaps.

[2] Permutations of just 3 parts never occur in a NDT, as they can always be further decomposed as a sequence of two binary nodes.

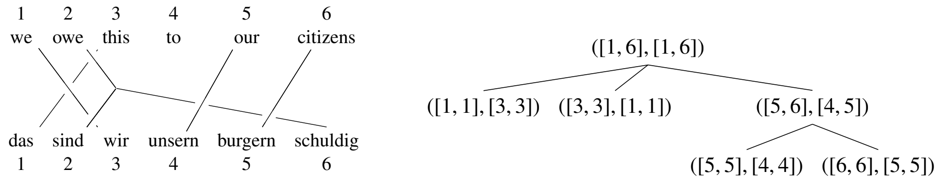

Figure 1: Example of complex word alignment, taken from Europarl data English-German (left) and its associated Normalized Decomposition Tree (Zhang et al., 2008) (right).

We show examples of each of these cases in Figure 2. Furthermore, in Figure 3 we show an example of an alignment labeled Hiero rule based on one of these alignment examples.

Our kind of labels has a completely different flavor from monolingual labels in that they cannot be seen as identifying linguistically meaningful clusters of phrase pairs. These labels are mere latent bilingual clusters and the translation model must marginalize over them (equation 1) or use Minimum Bayes-Risk decoding.

4.1 Features : Relations over labels

In this section we describe the features we use in our experiments. To be unambiguous we first need to introduce some terminology. Let r be a translation rule. We use $\hat{p}$ to denote probabilities estimated using simple relative frequency estimation from the word aligned sentence pairs of the training corpus. Then $src(r)$ is the source side of the rule, including the source side of the left-hand-side label. Similarly $tgt(r)$ is the target side of the rule, including the target side of the left-hand-side label. Furthermore $un(src(r))$ is the source side without any nonterminal labels, and analogous for $un(tgt(r))$.

4.1.1 Basic Features

We use the following phrase probability features:

- $\hat{p}(tgt(r)|src(r))$: Phrase probability target side given source side
- $\hat{p}(src(r)|tgt(r))$: Phrase probability source side given target side

We reinforce those by the following phrase probability smoothing features:

- $\hat{p}(tgt(r)|un(src(r)))$
- $\hat{p}(un(src(r))|tgt(r))$
- $\hat{p}(un(tgt(r))|src(r))$
- $\hat{p}(src(r)|un(tgt(r)))$
- $\hat{p}(un(tgt(r))|un(src(r)))$
- $\hat{p}(un(src(r))|un(tgt(r)))$

We also add the following features:

- $\hat{p}_w(tgt(r)|src(r))$, $\hat{p}_w(src(r)|tgt(r))$: Lexical weights based on terminal symbols as for phrase-based and hierarchical phrase-based MT.
- $\hat{p}(r|lhs(r))$: Generative probability of a rule given its left-hand-side label

We use the following set of basic binary features, with 1 values by default, and a value $exp(1)$ if the corresponding condition holds:

- $\varphi_{Glue}(r)$: $exp(1)$ if rule is a glue rule
- $\varphi_{lex}(r)$: $exp(1)$ if rule has only terminals on right-hand side
- $\varphi_{abs}(r)$: $exp(1)$ if rule has only nonterminals on right-hand side
- $\varphi_{st\_w\_tt}(r)$: $exp(1)$ if rule has terminals on the source side but not on the target side
- $\varphi_{tt\_w\_st}(r)$: $exp(1)$ if rule has terminals on the target side but not on the source side
- $\varphi_{mono}(r)$: $exp(1)$ if rule has no inverted pair of nonterminals

Furthermore we use the :

- $\varphi_{ra}(r)$: Phrase penalty, $exp(1)$ for all rules.
- $exp(\varphi_{wp}(r))$: Word penalty, exponent of the number of terminals on the target side
- $\varphi_{rare}(r)$: $exp(\frac{1}{\#(\sum_{r'\in C}\delta_{rr'})})$: Rarity penalty, with $\#(\sum_{r'\in C}\delta_{rr'})$ being the count of rule r in the corpus.

4.1.2 Binary Reordering Features

Besides the basic features we want to use extra sets of binary features that are specially designed to directly learn the desirability of certain broad classes of reordering patterns, beyond the way this is already implicitly learned for particular lexicalized rules by the introduction of reordering labels.[3] These features can be seen as generalizations of the most simple feature that penalizes/rewards mono-

[3] We did some initial experiments with such features in Joshua, but haven't managed yet to get them working in Moses with MBR. Since these experiments are inconclusive without MBR we leave them out here.

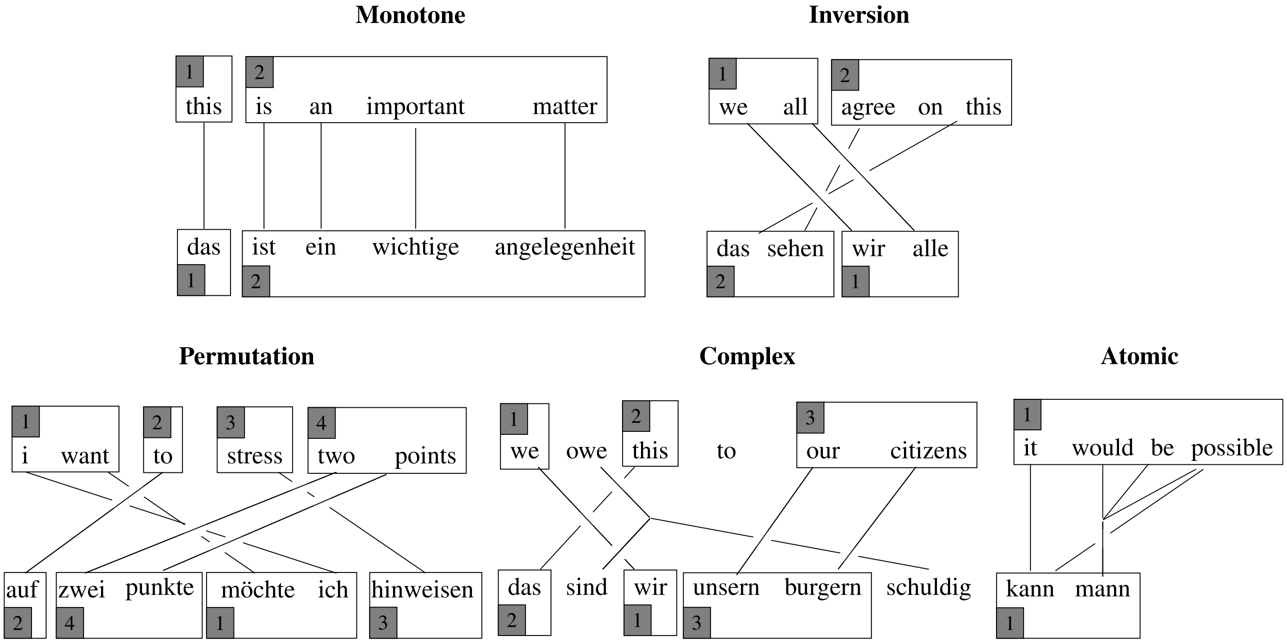

Figure 2: Different types of Alignment Labels

tone order $\varphi_{mono}(r)$ from our basic feature set. The new features we want to introduce "fire" for a specific combination of reordering labels on the left hand side and one or both gaps, plus optionally the information whether the rule itself invert its gaps and whether or not it is abstract.

5 Experiments

We evaluate our method on one language pair using German as source and English as target. The data is derived from parliament proceedings sourced from the Europarl corpus (Koehn, 2005), with WMT-07 development and test data. We used a maximum sentence length of 40 for filtering. We employ either 200K or (approximately) 1000K sentence pairs for training, 1K for development and 2K for testing (single reference per source sentence). Both the baseline and our method decode with a 3-gram language model smoothed with modified Knesser-Ney discounting (Chen and Goodman, 1998), trained on the target side of the full original training set (approximately 1000K sentences).

We compare against state-of-the-art hierarchical translation (Chiang, 2005) baselines, based on the Joshua (Ganitkevitch et al., 2012) and Moses (Hoang et al., 2007) translation systems with default decoding settings. We use our own grammar extractor for the generation of all grammars, including the baseline Hiero grammars. This enables us to use the same features (as far as applicable given the grammar formalism) and assure true comparability of the grammars under comparison.

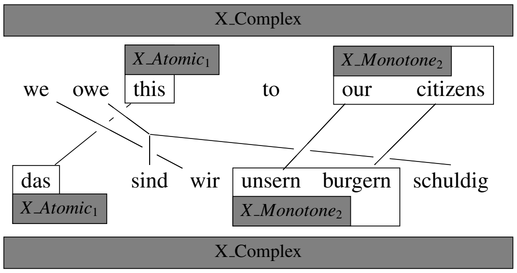

Figure 3: Example of a labeled Hiero rule $X_Complex \rightarrow \langle$we owe $X\_Atomic_{\boxed{1}}$ to $X\_Monotone_{\boxed{2}}$, $X\_Atomic_{\boxed{1}}$ sind wir $X\_Monotone_{\boxed{2}}$ schuldig $\rangle$ extracted from the *Complex* example in Figure 2 by replacing the phrase pairs ⟨this, das⟩ and ⟨our citizens , unsern burgern⟩ with (labeled) variables.

5.1 Training and Decoding Details

In this section we discuss the choices and settings we used in our experiments. Our initial experiments

[4] We later discovered we needed to add the flag "–return-best-dev" in Moses to actually get the weights from the best development run, our initial experiments had not used this. This explains the somewhat unfortunate drop in performance in our Analysis Experiments.

Decoding Type	System Name	**200K**
Lattice MBR	Hiero	26.44
	Hiero-RL	**26.72**
Viterbi	Hiero	26.23
	Hiero-RL-PPL	26.16

Table 1: Initial Results. Lowercase BLEU results for German-English trained on 200K sentence pairs.[4]
Top rows display results for our experiments using Moses (Hoang et al., 2007) with Lattice Minimum Bayes-Risk Decoding[5] (Tromble et al., 2008) in combination with Batch Mira (Cherry and Foster, 2012) for tuning. Below are results for experiments with Joshua (Ganitkevitch et al., 2012) using Viterbi decoding (i.e. no MBR) and PRO (Hopkins and May, 2011) for tuning.

were done on Joshua (Ganitkevitch et al., 2012), using the Viterbi best derivation. The second set of experiments was done on Moses (Hoang et al., 2007) using Lattice Minimum Bayes-Risk Decoding[5] (Tromble et al., 2008) to sum over derivations.

5.1.1 General Settings

To train our system we use the following settings. We use the standard Hiero grammar extraction constraints (Chiang, 2007) but for our reordering labeled grammars we use them with some modifications. In particular, while for basic Hiero only phrase pairs with source spans up to 10 are allowed, and abstract rules are forbidden, we allow extraction of fully abstract rules, without length constraints. Furthermore we allow their application without length constraints during decoding. Following common practice, we use simple relative frequency estimation to estimate the phrase probabilities, lexical probabilities and generative rule probability respectively.[6]

5.1.2 Specific choices and settings Joshua Viterbi experiments

Based on experiments reported in (Mylonakis and Sima'an, 2011; Mylonakis, 2012) we opted to not label the (fully lexicalized) phrase pairs, but instead label them with a generic *PhrasePair* label and use a set of switch rules from all other labels to the *PhrasePair* label to enable transition between Hiero rules and phrase pairs.

We train our systems using PRO (Hopkins and May, 2011) implemented in Joshua by Ganitkevitch et al. (2012). We use the standard tuning, where all features are treated as dense features.We allow up to 30 tuning iterations. We further follow the PRO settings introduced in (Ganitkevitch et al., 2012) but use 0.5 for the coefficient Ψ that interpolates the weights learned at the current with those from the previous iteration. We use the final learned weights for decoding with the log-linear model and report Lowercase BLEU scores for the tuned test set.

5.1.3 Specific choices and settings Moses Lattice MBR experiments

As mentioned before we use Moses (Hoang et al., 2007) for our second experiment, in combination with Lattice Minimum Bayes-Risk Decoding[5] (Tromble et al., 2008). Furthermore we use Batch Mira (Cherry and Foster, 2012) for tuning with maximum 10 tuning iterations of the 200K training set, and 30 for the 1000K training set.[7]

For our Moses experiments we mainly worked with a uniform labeling policy, labeling phrase pairs in the same way with alignment labels as normal rules. This is motivated by the fact that since we are using Minimum Bayes-Risk decoding, the risks of sparsity from labeling are much reduced. And labeling everything does have the advantage that reorder-

[5] After submission we were told by Moses support that in fact neither normal Minimum Bayes-Risk (MBR) nor Lattice MBR are operational in Moses Chart.

[6] Personal correspondence with Andreas Zollmann further reinforced the authors appreciation of the importance of this feature introduced in (Zollmann and Venugopal, 2006; Zollmann, 2011). Strangely enough this feature seems to be unavailable in the standard Moses (Hoang et al., 2007) and Joshua (Ganitkevitch et al., 2012) grammar extractors, that also implement SAMT grammar extraction

[7] We are mostly interested in the relative performance of our system in comparison to the baseline for the same settings. Nevertheless, it might be that the labeled systems, which have more smoothing features, are relatively suffering more from too little tuning iterations than the baseline which does not have these extra features and thus may be easier to tune. This was one of the reasons to increase the number of tuning iterations from 10 to 30 in our later experiments on 1000K. Usage of Minimum Bayes-Risk decoding or not is crucial as we have explained before in section 1. The main reason we opted for Batch Mira over PRO is that it is more commonly used in Moses systems, and in any case at least superior to MERT (Och, 2003) in most cases.

ing information can be fully propagated in derivations starting from the lowest (phrase) level. We also ran experiments with the generic phrase pair labeling, since there were reasons to believe this could decrease sparsity and potentially lead to better results.[8]

5.2 Initial Results

We report Lowercase BLEU scores for experiments with and without Lattice Minimum Bayes-Risk (MBR) decoding (Tromble et al., 2008). Table 1 bottom shows the results of our first experiments with Joshua, using the Viterbi derivation and no MBR decoding to sum over derivations. We display scores for the Hiero baseline (Hiero) and the (partially) alignment labeled system (Hiero-AL-PPL) which uses alignment labels for Hiero rules and PhrasePair to label all phrase pairs. Scores are around 26.25 BLEU for both systems, with only marginal differences. In summary our labeled systems are at best comparable to the Hiero baseline.

Table 1 top shows the results of our second experiments with Moses and Lattice MBR[5]. Here our (fully) alignment labeled system (Hiero-AL) achieves a score of 26.72 BLEU, in comparison to 26.44 BLEU for the Hiero baseline (Hiero). A small improvement of 0.28 BLEU point.

5.3 Advanced experiments

We now report Lowercase BLEU scores for more detailed analysis experiments with and without Lattice Minimum Bayes-Risk[5] (MBR) decoding, where we varied other training and decoding parameters in the Moses environment. Particularly, in this set of experiments we choose the *best tuning parameter settings* over 30 Batch Mira iterations (as opposed to the weights returned by default – used in the previous experiments). We also explore varieties in tuning with a decoder that works with Viterbi/MBR, and final testing with Viterbi/MBR.

Decoding Type	System Name	**200K**	**1000K**
Lattice MBR	Hiero	27.19	28.39
	Hiero-AL	26.61	28.32
	Hiero-AL-PPL	26.89	28.41
Viterbi	Hiero	26.80	**28.57**
	Hiero-AL	———	28.36

Table 2: Analysis Results. Lowercase BLEU results for German-English trained on 200K and 1000K sentence pairs using Moses (Hoang et al., 2007) in combination with Batch Mira (Cherry and Foster, 2012) for tuning. Top rows display results for our experiments with Lattice Minimum Bayes-Risk Decoding[5] (Tromble et al., 2008). Below are results for experiments using Viterbi decoding (i.e. no MBR) for tuning. Results on 200K were run with 10 tuning iterations, results on 1000K with 30 tuning iterations.

In Table 2, the top rows show the results of our experiments using MBR decoding. We display scores for the Hiero baseline (Hiero) and the fully/partially alignment labeled systems Hiero-AL and Hiero-AL-PPL. In the preceding set of experiments MBR decoding clearly showed improved performance over Viterbi, particularly for our labelled system.

On the small training set of 200K we observe that the Hiero baseline achieves 27.19 BLEU and thus beats the labeled systems Hiero-AL with 26.61 BLEU and 26.89 BLEU by a good margin. On the bigger dataset of 1000K and with more tuning iterations (3), all systems perform better. When using Lattice MBR Hiero achieving 28.39 BLEU, Hiero-AL 28.32 BLEU and finally Hiero-AL-PPL achieves 28.41. These are insignificant differences in performance between the labelled and unlabeled systems.

Table 1 bottom also shows the results of our second set of experiments with *Viterbi decoding*. Here, the baseline Hiero system for 200K training set achieves a score of 26.80 BLEU on the small training set. We also conducted another set of experiments on the larger training set of 1000K, this time with Viterbi decoding. The Hiero baseline with Viterbi scores 28.57 BLEU while Hiero-AL scores 28.36 BLEU under the same conditions.

It is puzzling that Hiero Viterbi (for 1000k) performs better than the same system with MBR decoding systems. But after submission we were told by Moses support that neither normal MBR nor Lattice MBR are operational in Moses Chart. This means that in fact the effect of MBR on our labels remains still undecided, and more work is still needed in this direction. The small decrease in performance for the

[8] We discovered that the Moses chart decoder does not allow fully abstract unary rules in the current implementation, which makes direct usage of unary (switch) rules not possible. Switch rules and other unaries can still be emulated though, by adapting the grammar, using multiple copies of rules with different labels. This blows up the grammar a bit, but at least works in practice.

labelled system relative to Hiero (in Viterbi) is possibly the result of the labelled system being more brittle and harder to tune than the Hiero system. This hypothesis needs further exploration.

While a whole set of experimental questions remains open, we think that based on this preliminary but nevertheless considerable set of experiments, it seems that our labels do not always improve performance compared with the Hiero baseline. It is possible that these labels, under a more advanced implementation via soft constraints (as opposed to hard labeling), could provide the empirical evidence to our theoretical choices. A further concern regarding the labels is that our current choice (5 labels) is heuristic and not optimized for the training data. It remains to be seen in the future if proper learning of these labels as latent variables optimized for the training data or the use of soft constraints can shed more light on the use of alignment labels in hierarchical SMT.

5.4 Analysis

While we did not have time to do a deep comparative analysis of the properties of the grammars, a few things can be said based on the results. First of all we have seen that alignment labels do not always improve over the Hiero baseline. In earlier experiments we observed some improvement when the labelled grammar was used in combination with Minimum Bayes-Risk Decoding but not without it. In later experiments with different tuning settings (Mira), the improvements evaporated and in fact, the Viterbi Hiero baseline turned out, surprisingly, the best of all systems.

Our use of MBR is theoretically justified by the importance of aggregating over the derivations of the output translations when labeling Hiero variables: statistically speaking, if the labels split the occurrences of the phrase pairs, they will lead to multiple derivations per Hiero derivation with fractions of the scores. This is in line with earlier work on the effect of spurious ambiguity, e.g. Variational Decoding (Li et al., 2009). Yet, in the case of our model, there is also a conceptual explanation for the need to aggregate over different derivations of the same sentence pair. The decomposition of a word alignment into hierarchical decomposition trees has a interesting property: the simpler (less reordering) a word alignment, the more (binary) decomposition trees – and in our model derivations – it will have. Hence, aggregating over the derivations is a way to gather evidence for the complexity of alignment patterns that our model can fit in between a given source-target sentence pair. However, in the current experimental setting, where final tuning with Mira is crucial, and where the use of MBR within Moses is still not standard, we cannot reap full benefit of our theoretical analysis concerning the fit of MBR for our models' alignment labels.

6 Conclusion

We presented a novel method for labeling Hiero variables with nonterminals derived from the hierarchical patterns found in recursive decompositions of word alignments into tree representations. Our experiments based on a first instantiation of this idea with a fixed set of labels, not optimized to the training data, show promising performance. Our early experiments suggested that these labels have merit, whereas later experiments with more varied training and decoder settings showed these results to be unstable.

Empirical results aside, our approach opens up a whole new line of research to improve the state of the art of hierarchical SMT by learning these latent alignment labels directly from standard word alignments without special use of syntactic or other parsers. The fact that such labels are in principle complementary with monolingual information is an exciting perspective which we might explore in future work.

Acknowledgements

This work is supported by The Netherlands Organization for Scientific Research (NWO) under grant nr. 612.066.929. This work was sponsored by the BIG Grid project for the use of the computing and storage facilities, with financial support from the Netherlands Organization of Scientific Research (NWO) under grant BG-087-12. The authors would like to thank the people from the Joshua team at John Hopkins University, in particular Yuan Cao, Jonathan Weese, Matt Post and Juri Ganitkevitch, for their helpful replies to questions regarding Joshua and its PRO and Packing implementations.

References

Alfred V. Aho and Jeffrey D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci.*, 3(1):37–56.

Hala Almaghout, Jie Jiang, and Andy Way. 2011. Ccg contextual labels in hierarchical phrase-based smt. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation (EAMT-2011)*, May.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *HLT-NAACL*, pages 427–436.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270, June.

David Chiang. 2006. An introduction to synchronous grammars.

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

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch. 2012. Joshua 4.0: Packing, pro, and paraphrases. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 283–291, Montréal, Canada, June. Association for Computational Linguistics.

Hany Hassan, Khalil Sima'an, and Andy Way. 2007. Supertagged phrase-based statistical machine translation. In *Proceedings of ACL 2007*, page 288295.

Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, and Ondrej Bojar. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 177–180.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *HLT-NAACL*, page 16917.

Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009. Variational decoding for statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, pages 593–601.

Junhui Li, Zhaopeng Tu, Guodong Zhou, and Josef van Genabith. 2012. Using syntactic head information in hierarchical phrase-based translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 232–242.

Yuval Marton, David Chiang, and Philip Resnik. 2012. Soft syntactic constraints for arabic—english hierarchical phrase-based translation. *Machine Translation*, 26(1-2):137–157.

Markos Mylonakis and Khalil Sima'an. 2011. Learning hierarchical translation structure with linguistic annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–652.

Markos Mylonakis. 2012. *Learning the Latent Structure of Translation*. Ph.D. thesis, University of Amsterdam.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167.

Khalil Sima'an and Gideon Maillette de Buy Wenniger. 2011. Hierarchical translation equivalence over word alignments. Technical Report PP-2011-38, Institute for Logic, Language and Computation.

Roy W. Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 620–629.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 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.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–404.

Hao Zhang, Daniel Gildea, and David Chiang. 2008. Extracting synchronous grammar rules from word-level alignments in linear time. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 1081–1088.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In

NAACL 2006 - Workshop on statistical machine translation, June.

Andreas Zollmann. 2011. *Learning Multiple-Nonterminal Synchronous Grammars for Statistical Machine Translation*. Ph.D. thesis, Carnegie Mellon University.