Cohesive Phrase-based Decoding for Statistical Machine Translation

Colin Cherry*
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
colinc@microsoft.com

Abstract

Phrase-based decoding produces state-of-the-art translations with no regard for syntax. We add syntax to this process with a cohesion constraint based on a dependency tree for the source sentence. The constraint allows the decoder to employ arbitrary, non-syntactic phrases, but ensures that those phrases are translated in an order that respects the source tree's structure. In this way, we target the phrasal decoder's weakness in order modeling, without affecting its strengths. To further increase flexibility, we incorporate cohesion as a decoder feature, creating a soft constraint. The resulting cohesive, phrase-based decoder is shown to produce translations that are preferred over non-cohesive output in both automatic and human evaluations.

1 Introduction

Statistical machine translation (SMT) is complicated by the fact that words can move during translation. If one assumes arbitrary movement is possible, that alone is sufficient to show the problem to be NP-complete (Knight, 1999). **Syntactic cohesion**[1] is the notion that all movement occurring during translation can be explained by permuting children in a parse tree (Fox, 2002). Equivalently, one can say that phrases in the source, defined by subtrees in its parse, remain contiguous after translation. Early methods for syntactic SMT held to this assumption in its entirety (Wu, 1997; Yamada and Knight, 2001). These approaches were eventually superseded by tree transducers and tree substitution grammars, which allow translation events to span subtree units, providing several advantages, including the ability to selectively produce uncohesive translations (Eisner, 2003; Graehl and Knight, 2004; Quirk et al., 2005). What may have been forgotten during this transition is that there is a reason it was once believed that a cohesive translation model would work: for some language pairs, cohesion explains nearly all translation movement. Fox (2002) showed that cohesion is held in the vast majority of cases for English-French, while Cherry and Lin (2006) have shown it to be a strong feature for word alignment. We attempt to use this strong, but imperfect, characterization of movement to assist a non-syntactic translation method: phrase-based SMT.

Phrase-based decoding (Koehn et al., 2003) is a dominant formalism in statistical machine translation. Contiguous segments of the source are translated and placed in the target, which is constructed from left to right. The process iterates within a beam search until each word from the source has been covered by exactly one phrasal translation. Candidate translations are scored by a linear combination of models, weighted according to Minimum Error Rate Training or MERT (Och, 2003). Phrasal SMT draws strength from being able to memorize non-compositional and context-specific translations, as well as local reorderings. Its primary weakness is in movement modeling; its default distortion model applies a flat penalty to any deviation from source

*Work conducted while at the University of Alberta.

[1]We use the term "syntactic cohesion" throughout this paper to mean what has previously been referred to as "phrasal cohesion", because the non-linguistic sense of "phrase" has become so common in machine translation literature.

Proceedings of ACL-08: HLT, pages 72–80,
Columbus, Ohio, USA, June 2008.

order, forcing the decoder to rely heavily on its language model. Recently, a number of data-driven distortion models, based on lexical features and relative distance, have been proposed to compensate for this weakness (Tillman, 2004; Koehn et al., 2005; Al-Onaizan and Papineni, 2006; Kuhn et al., 2006).

There have been a number of proposals to incorporate syntactic information into phrasal decoding. Early experiments with syntactically-informed phrases (Koehn et al., 2003), and syntactic reranking of K-best lists (Och et al., 2004) produced mostly negative results. The most successful attempts at syntax-enhanced phrasal SMT have directly targeted movement modeling: Zens et al. (2004) modified a phrasal decoder with ITG constraints, while a number of researchers have employed syntax-driven source reordering before decoding begins (Xia and McCord, 2004; Collins et al., 2005; Wang et al., 2007).[2] We attempt something between these two approaches: our constraint is derived from a linguistic parse tree, but it is used inside the decoder, not as a preprocessing step.

We begin in Section 2 by defining syntactic cohesion so it can be applied to phrasal decoder output. Section 3 describes how to add both hard and soft cohesion constraints to a phrasal decoder. Section 4 provides our results from both automatic and human evaluations. Sections 5 and 6 provide a qualitative discussion of cohesive output and conclude.

2 Cohesive Phrasal Output

Previous approaches to measuring the cohesion of a sentence pair have worked with a word alignment (Fox, 2002; Lin and Cherry, 2003). This alignment is used to project the spans of subtrees from the source tree onto the target sentence. If a modifier and its head, or two modifiers of the same head, have overlapping spans in the projection, then this indicates a cohesion violation. To check phrasal translations for cohesion violations, we need a way to project the source tree onto the decoder's output.

Fortunately, each phrase used to create the target sentence can be tracked back to its original source phrase, providing an alignment between source and target phrases. Since each source token is used exactly once during translation, we can transform this phrasal alignment into a word-to-phrase alignment, where each source token is linked to a target phrase. We can then project the source subtree spans onto the target phrase sequence. Note that we never consider individual tokens on the target side, as their connection to the source tree is obscured by the phrasal abstraction that occurred during translation.

Let e_1^m be the input source sentence, and $\bar{f}_1^p$ be the output target phrase sequence. Our word-to-phrase alignment $a_i \in [1,p], 1 \leq i \leq m$, maps a source token position i to a target phrase position a_i. Next, we introduce our source dependency tree T. Each source token e_i is also a node in T. We define $T(e_i)$ to be the subtree of T rooted at e_i. We define a local tree to be a head node and its immediate modifiers. With this notation in place, we can define our projected spans. Following Lin and Cherry (2003), we define a head span to be the projection of a single token e_i onto the target phrase sequence:

$$spanH(e_i, T, a_1^m) = [a_i, a_i]$$

and the subtree span to be the projection of the subtree rooted at e_i:

$$spanS(e_i, T, a_1^m) = \left[\min_{\{j|e_j \in T(e_i)\}} a_j, \max_{\{k|e_k \in T(e_i)\}} a_k \right]$$

Consider the simple phrasal translation shown in Figure 1 along with a dependency tree for the English source. If we examine the local tree rooted at *likes*, we get the following projected spans:

$$\begin{aligned} spanS(nobody, T, a) &= [1,1] \\ spanH(likes, T, a) &= [1,1] \\ spanS(pay, T, a) &= [1,2] \end{aligned}$$

For any local tree, we consider only the head span of the head, and the subtree spans of any modifiers.

Typically, cohesion would be determined by checking these projected spans for intersection. However, at this level of resolution, avoiding intersection becomes highly restrictive. The monotone translation in Figure 1 would become non-cohesive: *nobody* intersects with both its sibling *pay* and with its head *likes* at phrase index 1. This complication stems from the use of multi-word phrases that

[2] While certainly both syntactic and successful, we consider Hiero (Chiang, 2007) to be a distinct approach, and not an extension to phrasal decoding's left-to-right beam search.

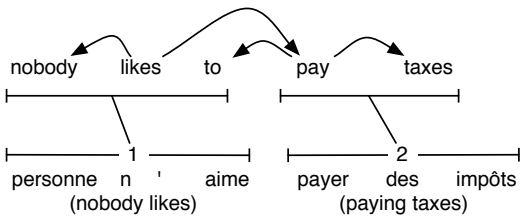

Figure 1: An English source tree with translated French output. Segments are indicated with underlined spans.

Span	(a)	(b)
$spanS(session, T, a)$	[1,3]	[1,3]*
$spanH(begins, T, a)$	[4,4]	[2,2]*
$spanS(tomorrow, T, a)$	[4,4]	[4,4]

Table 1: Spans of the local trees rooted at *begins* from Figures 2 (a) and (b). Innersection is marked with a "*".

do not correspond to syntactic constituents. Restricting phrases to syntactic constituents has been shown to harm performance (Koehn et al., 2003), so we tighten our definition of a violation to disregard cases where the only point of overlap is obscured by our phrasal resolution. To do so, we replace span intersection with a new notion of span **innersection**.

Assume we have two spans $[u, v]$ and $[x, y]$ that have been sorted so that $[u, v] \leq [x, y]$ lexicographically. We say that the two spans **innersect** if and only if $x < v$. So, $[1, 3]$ and $[2, 4]$ innersect, while $[1, 3]$ and $[3, 4]$ do not. One can think of innersection as intersection, minus the cases where the two spans share only a single boundary point, where $x = v$. When two projected spans innersect, it indicates that the second syntactic constituent must begin before the first ends. If the two spans in question correspond to nodes in the same local tree, innersection indicates an unambiguous cohesion violation. Under this definition, the translation in Figure 1 is cohesive, as none of its spans innersect.

Our hope is that syntactic cohesion will help the decoder make smarter distortion decisions. An example with distortion is shown in Figure 2. In this case, we present two candidate French translations of an English sentence, assuming there is no entry in the phrase table for "voting session." Because the proper French construction is "session of voting", the decoder has to move *voting* after *session* using a distortion operation. Figure 2 shows two methods to do so, each using an equal numbers of phrases. The projected spans for the local tree rooted at *begins* in each candidate are shown in Table 1. Note the innersection between the head *begins* and its modifier *session* in (b). Thus, a cohesion-aware system would receive extra guidance to select (a), which maintains the original meaning much better than (b).

2.1 K-best List Filtering

A first attempt at using cohesion to improve SMT output would be to apply our definition as a filter on K-best lists. That is, we could have a phrasal decoder output a 1000-best list, and return the highest-ranked cohesive translation to the user. We tested this approach on our English-French development set, and saw no improvement in BLEU score. Error analysis revealed that only one third of the uncohesive translations had a cohesive alternative in their 1000-best lists. In order to reach the remaining two thirds, we need to constrain the decoder's search space to explore only cohesive translations.

3 Cohesive Decoding

This section describes a modification to standard phrase-based decoding, so that the system is constrained to produce only cohesive output. This will take the form of a check performed each time a hypothesis is extended, similar to the ITG constraint for phrasal SMT (Zens et al., 2004). To create a such a check, we need to detect a cohesion violation inside a partial translation hypothesis. We cannot directly apply our span-based cohesion definition, because our word-to-phrase alignment is not yet complete. However, we can still detect violations, and we can do so before the spans involved are completely translated.

Recall that when two projected spans a and b ($a < b$) innersect, it indicates that b begins before a ends. We can say that the translation of b interrupts the translation of a. We can enforce cohesion by ensuring that these **interruptions** never happen. Because the decoder builds its translations from left to right, eliminating interruptions amounts to enforcing the following rule: once the decoder begins translating any part of a source subtree, it must cover all

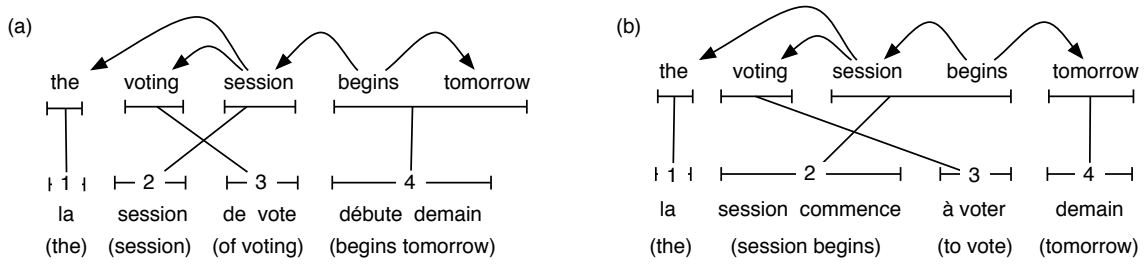

Figure 2: Two candidate translations for the same parsed source. (a) is cohesive, while (b) is not.

the words under that subtree before it can translate anything outside of it.

For example, in Figure 2b, the decoder translates *the*, which is part of $T(session)$ in $\bar{f}_1$. In $\bar{f}_2$, it translates *begins*, which is outside $T(session)$. Since we have yet to cover *voting*, we know that the projected span of $T(session)$ will end at some index $v > 2$, creating an innersection. This eliminates the hypothesis after having proposed only the first two phrases.

3.1 Algorithm

In this section, we formally define an interruption, and present an algorithm to detect one during decoding. During both discussions, we represent each target phrase as a set that contains the English tokens used in its translation: $\bar{f}_j = \{e_i | a_i = j\}$. Formally, an interruption occurs whenever the decoder would add a phrase $\bar{f}_{h+1}$ to the hypothesis $\bar{f}_1^h$, and:

$$\begin{array}{lll} \exists r \in T & \text{such that:} & \\ \exists e \in T(r) & \text{s.t. } e \in \bar{f}_1^h & \text{(a. Started)} \\ \exists e' \notin T(r) & \text{s.t. } e' \in \bar{f}_{h+1} & \text{(b. Interrupted)} \\ \exists e'' \in T(r) & \text{s.t. } e'' \notin \bar{f}_1^{h+1} & \text{(c. Unfinished)} \end{array} \quad (1)$$

The key to checking for interruptions quickly is knowing which subtrees $T(r)$ to check for qualities (1:a,b,c). A naïve approach would check every subtree that has begun translation in $\bar{f}_1^h$. Figure 3a highlights the roots of all such subtrees for a hypothetical T and $\bar{f}_1^h$. Fortunately, with a little analysis that accounts for $\bar{f}_{h+1}$, we can show that at most two subtrees need to be checked.

For a given interruption-free $\bar{f}_1^h$, we call subtrees that have begun translation, but are not yet complete, **open** subtrees. Only open subtrees can lead to interruptions. We can focus our interruption check on $\bar{f}_h$, the last phrase in $\bar{f}_1^h$, as any open subtree $T(r)$ must contain at least one $e \in \bar{f}_h$. If this were not the case, then the open $T(r)$ must have began translation somewhere in $\bar{f}_1^{h-1}$, and $T(r)$ would be interrupted by the placement of $\bar{f}_h$. Since our hypothesis $\bar{f}_1^h$ is interruption-free, this is impossible. This leaves the subtrees highlighted in Figure 3b to be checked. Furthermore, we need only consider subtrees that contain the left and right-most source tokens e_L and e_R translated by $\bar{f}_h$. Since $\bar{f}_h$ was created from a contiguous string of source tokens, any distinct subtree between these two endpoints will be completed within $\bar{f}_h$. Finally, for each of these focus points e_L and e_R, only the highest containing subtree $T(r)$ that does not completely contain $\bar{f}_{h+1}$ needs to be considered. Anything higher would contain all of $\bar{f}_{h+1}$, and would not satisfy requirement (1:b) of our interruption definition. Any lower subtree would be a descendant of r, and therefore the check for the lower subtree is subsumed by the check for $T(r)$. This leaves only two subtrees, highlighted in our running example in Figure 3c.

Algorithm 1 Interruption check.

- Get the left and right-most tokens used to create $\bar{f}_h$, call them e_L and e_R
- For each of $e \in \{e_L, e_R\}$:
  - i. $r' \leftarrow e, r \leftarrow null$
    While $\exists e' \in \bar{f}_{h+1}$ such that $e' \notin T(r')$:
    $r \leftarrow r', r' \leftarrow parent(r)$
  - ii. If $r \neq null$ and $\exists e'' \in T(r)$ such that $e'' \notin \bar{f}_1^{h+1}$, then $\bar{f}_{h+1}$ interrupts $T(r)$.

With this analysis in place, an extension $\bar{f}_{h+1}$ of the hypothesis $\bar{f}_1^h$ can be checked for interruptions with Algorithm 1. Step (i) in this algorithm finds an ancestor r' such that $T(r')$ completely contains

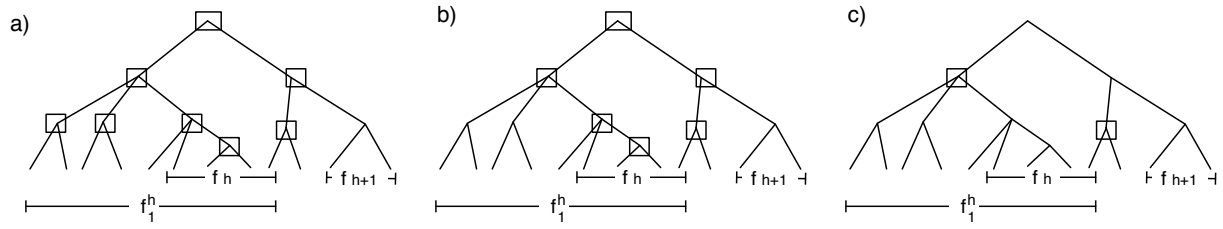

Figure 3: Narrowing down the source subtrees to be checked for completeness.

$\bar{f}_{h+1}$, and then returns r, the highest node that **does not** contain $\bar{f}_{h+1}$. We know this r satisfies requirements (1:a,b). If there is no $T(r)$ that does not contain $\bar{f}_{h+1}$, then e and its ancestors cannot lead to an interruption. Step (ii) then checks the coverage vector of the hypothesis[3] to make sure that $T(r)$ is covered in $\bar{f}_1^{h+1}$. If $T(r)$ is not complete in $\bar{f}_1^{h+1}$, then that satisfies requirement (1:c), which means an interruption has occurred.

For example, in Figure 2b, our first interruption occurs as we add $\bar{f}_{h+1} = \bar{f}_2$ to $\bar{f}_1^h = \bar{f}_1^1$. The detection algorithm would first get the left and right boundaries of $\bar{f}_1$; in this case, *the* is both e_L and e_R. Then, it would climb up the tree from *the* until it reached $r' = \textit{begins}$ and $r = \textit{session}$. It would then check $T(\textit{session})$ for coverage in $\bar{f}_1^2$. Since $\textit{voting} \in T(\textit{session})$ is not covered in $\bar{f}_1^2$, it would detect an interruption.

Walking up the tree takes at most linear time, and each check to see if $T(r)$ contains all of $\bar{f}_{h+1}$ can be performed in constant time, provided the source spans of each subtree have been precomputed. Checking to see if all of $T(r)$ has been covered in Step (ii) takes at most linear time. This makes the entire process linear in the size of the source sentence.

3.2 Soft Constraint

Syntactic cohesion is not a perfect constraint for translation. Parse errors and systematic violations can create cases where cohesion works against the decoder. Fox (2002) demonstrated and counted cases where cohesion was not maintained in hand-aligned sentence-pairs, while Cherry and Lin (2006) showed that a soft cohesion constraint is superior to a hard constraint for word alignment. Therefore, we propose a soft version of our cohesion constraint. We perform our interruption check, but we do not invalidate any hypotheses. Instead, each hypothesis maintains a count of the number of extensions that have caused interruptions during its construction. This count becomes a feature in the decoder's log-linear model, the weight of which is trained with MERT. After the first interruption, the exact meaning of further interruptions becomes difficult to interpret; but the interruption count does provide a useful estimate of the extent to which the translation is faithful to the source tree structure.

Initially, we were not certain to what extent this feature would be used by the MERT module, as BLEU is not always sensitive to syntactic improvements. However, trained with our French-English tuning set, the interruption count received the largest absolute feature weight, indicating, at the very least, that the feature is worth scaling to impact decoder.

3.3 Implementation

We modify the Moses decoder (Koehn et al., 2007) to translate head-annotated sentences. The decoder stores the flat sentence in the original sentence data structure, and the head-encoded dependency tree in an attached tree data structure. The tree structure caches the source spans corresponding to each of its subtrees. We then implement both a hard check for interruptions to be used before hypotheses are placed on the stack,[4] and a soft check that is used to calculate an interruption count feature.

[3] This coverage vector is maintained by all phrasal decoders to track how much of the source sentence has been covered by the current partial translation, and to ensure that the same token is not translated twice.

[4] A hard cohesion constraint used in conjunction with a traditional distortion limit also requires a second linear-time check to ensure that all subtrees currently in progress can be finished under the constraints induced by the distortion limit.

Set	Cohesive	Uncohesive
Dev-Test	1170	330
Test	1563	437

Table 2: Number of sentences that receive cohesive translations from the baseline decoder. This property also defines our evaluation subsets.

4 Experiments

We have adapted the notion of syntactic cohesion so that it is applicable to phrase-based decoding. This results in a translation process that respects source-side syntactic boundaries when distorting phrases. In this section we will test the impact of such information on an English to French translation task.

4.1 Experimental Details

We test our cohesion-enhanced Moses decoder trained using 688K sentence pairs of Europarl French-English data, provided by the SMT 2006 Shared Task (Koehn and Monz, 2006). Word alignments are provided by GIZA++ (Och and Ney, 2003) with grow-diag-final combination, with infrastructure for alignment combination and phrase extraction provided by the shared task. We decode with Moses, using a stack size of 100, a beam threshold of 0.03 and a distortion limit of 4. Weights for the log-linear model are set using MERT, as implemented by Venugopal and Vogel (2005). Our tuning set is the first 500 sentences of the SMT06 development data. We hold out the remaining 1500 development sentences for development testing (dev-test), and the entirety of the provided 2000-sentence test set for blind testing (test). Since we require source dependency trees, all experiments test English to French translation. English dependency trees are provided by Minipar (Lin, 1994).

Our cohesion constraint directly targets sentences for which an unmodified phrasal decoder produces uncohesive output according to the definition in Section 2. Therefore, we present our results not only on each test set in its entirety, but also on the subsets defined by whether or not the baseline naturally produces a cohesive translation. The sizes of the resulting evaluation sets are given in Table 2.

Our development tests indicated that the soft and hard cohesion constraints performed somewhat similarly, with the soft constraint providing more stable, and generally better results. We confirmed these trends on our test set, but to conserve space, we provide detailed results for only the soft constraint.

4.2 Automatic Evaluation

We first present our soft cohesion constraint's effect on BLEU score (Papineni et al., 2002) for both our dev-test and test sets. We compare against an unmodified baseline decoder, as well as a decoder enhanced with a lexical reordering model (Tillman, 2004; Koehn et al., 2005). For each phrase pair in our translation table, the lexical reordering model tracks statistics on its reordering behavior as observed in our word-aligned training text. The lexical reordering model provides a good comparison point as a non-syntactic, and potentially orthogonal, improvement to phrase-based movement modeling. We use the implementation provided in Moses, with probabilities conditioned on bilingual phrases and predicting three orientation bins: straight, inverted and disjoint. Since adding features to the decoder's log-linear model is straight-forward, we also experiment with a combined system that uses both the cohesion constraint and a lexical reordering model.

The results of our experiments are shown in Table 3, and reveal some interesting phenomena. First of all, looking across columns, we can see that there is a definite divide in BLEU score between our two evaluation subsets. Sentences with cohesive baseline translations receive much higher BLEU scores than those with uncohesive baseline translations. This indicates that the cohesive subset is easier to translate with a phrase-based system. Our definition of cohesive phrasal output appears to provide a useful feature for estimating translation confidence.

Comparing the baseline with and without the soft cohesion constraint, we see that cohesion has only a modest effect on BLEU, when measured on all sentence pairs, with improvements ranging between 0.2 and 0.5 absolute points. Recall that the majority of baseline translations are naturally cohesive. The cohesion constraint's effect is much more pronounced on the more difficult uncohesive subsets, showing absolute improvements between 0.5 and 1.1 points.

Considering the lexical reordering model, we see that its effect is very similar to that of syntactic cohesion. Its BLEU scores are very similar, with lex-

	Dev-Test			Test		
System	All	Cohesive	Uncohesive	All	Cohesive	Uncohesive
base	32.04	33.80	27.46	32.35	33.78	28.73
lex	32.19	33.91	27.86	**32.71**	33.89	**29.66**
coh	32.22	33.82	**28.04**	**32.88**	**34.03**	**29.86**
lex+coh	**32.45**	34.12	**28.09**	**32.90**	34.04	**29.83**

Table 3: BLEU scores with an integrated soft cohesion constraint (coh) or a lexical reordering model (lex). Any system significantly better than base has been highlighted, as tested by bootstrap re-sampling with a 95% confidence interval.

ical reordering also affecting primarily the uncohesive subset. This similarity in behavior is interesting, as its data-driven, bilingual reordering probabilities are quite different from our cohesion flag, which is driven by monolingual syntax.

Examining the system that employs both movement models, we see that the combination (**lex+coh**) receives the highest score on the dev-test set. A large portion of the combined system's gain is on the cohesive subset, indicating that the cohesion constraint may be enabling better use of the lexical reordering model on otherwise cohesive translations. Unfortunately, these same gains are not born out on the test set, where the lexical reordering model appears unable to improve upon the already strong performance of the cohesion constraint.

4.3 Human Evaluation

We also present a human evaluation designed to determine whether bilingual speakers prefer cohesive decoder output. Our comparison systems are the baseline decoder (**base**) and our soft cohesion constraint (**coh**). We evaluate on our dev-test set,[5] as it has our smallest observed BLEU-score gap, and we wish to determine if it is actually improving. Our experimental set-up is modeled after the human evaluation presented in (Collins et al., 2005). We provide two human annotators[6] a set of 75 English source sentences, along with a reference translation and a pair of translation candidates, one from each system. The annotators are asked to indicate which of the two system translations they prefer, or if they consider them to be equal. To avoid bias, the competing systems were presented anonymously and in random order. Following (Collins et al., 2005), we provide the annotators with only short sentences: those with source sentences between 10 and 25 tokens long. Following (Callison-Burch et al., 2006), we conduct a targeted evaluation; we only draw our evaluation pairs from the uncohesive subset targeted by our constraint. All 75 sentences that meet these two criteria are included in the evaluation.

	Annotator #2			
Annotator #1	base	coh	equal	sum (#1)
base	**6**	7	1	14
coh	8	**35**	4	47
equal	7	4	**3**	14
sum (#2)	21	46	8	

Table 4: Confusion matrix from human evaluation.

The aggregate results of our human evaluation are shown in the bottom row and right-most column of Table 4. Each annotator prefers **coh** in over 60% of the test sentences, and each prefers **base** in less than 30% of the test sentences. This presents strong evidence that we are having a consistent, positive effect on formerly non-cohesive translations. A complete confusion matrix indicating agreement between the two annotators is also given in Table 4. There are a few more off-diagonal points than one might expect, but it is clear that the two annotators are in agreement with respect to **coh**'s improvements. A combination annotator, which selects **base** or **coh** only when both human annotators agree and equal otherwise, finds **base** is preferred in only 8% of cases, compared to 47% for **coh**.

[5]The cohesion constraint has no free parameters to optimize during development, so this does not create an advantage.

[6]Annotators were both native English speakers who speak French as a second language. Each has a strong comprehension of written French.

(1+)	creating structures that do not currently exist and reducing . . .
base	de créer des structures qui existent actuellement et ne pas réduire . . . *to create structures that* ***actually exist*** *and* ***do not reduce*** *. . .*
coh	de créer des structures qui n ' existent pas encore et réduire . . . *to create structures that* ***do not yet exist*** *and* ***reduce*** *. . .*
(2−)	. . . repealed the 1998 directive banning advertising
base	. . . abrogée l'interdiction de la directive de 1998 de publicité *. . . repealed* ***the ban*** *from the 1998 directive* ***on advertising***
coh	. . . abrogée la directive de 1998 l'interdiction de publicité *. . . repealed the 1998 directive* ***the ban on advertising***

Table 5: A comparison of baseline and cohesion-constrained English-to-French translations, with English glosses.

5 Discussion

Examining the French translations produced by our cohesion constrained phrasal decoder, we can draw some qualitative generalizations. The constraint is used primarily to prevent distortion: it provides an intelligent estimate as to when source order must be respected. The resulting translations tend to be more literal than unconstrained translations. So long as the vocabulary present in our phrase table and language model supports a literal translation, cohesion tends to produce an improvement. Consider the first translation example shown in Table 5. In the baseline translation, the language model encourages the system to move the negation away from "exist" and toward "reduce." The result is a tragic reversal of meaning in the translation. Our cohesion constraint removes this option, forcing the decoder to assemble the correct French construction for "does not yet exist." The second example shows a case where our resources do not support a literal translation. In this case, we do not have a strong translation mapping to produce a French modifier equivalent to the English "banning." Stuck with a noun form ("the ban"), the baseline is able to distort the sentence into something that is almost correct (the above gloss is quite generous). The cohesive system, even with a soft constraint, cannot reproduce the same movement, and returns a less grammatical translation.

We also examined cases where the decoder overrides the soft cohesion constraint and produces an uncohesive translation. We found this was done very rarely, and primarily to overcome parse errors. Only one correct syntactic construct repeatedly forced the decoder to override cohesion: Minipar's conjunction representation, which connects conjuncts in parent-child relationships, is at times too restrictive. A sibling representation, which would allow conjuncts to be permuted arbitrarily, may work better.

6 Conclusion

We have presented a definition of syntactic cohesion that is applicable to phrase-based SMT. We have used this definition to develop a linear-time algorithm to detect cohesion violations in partial decoder hypotheses. This algorithm was used to implement a soft cohesion constraint for the Moses decoder, based on a source-side dependency tree.

Our experiments have shown that roughly 1/5 of our baseline English-French translations contain cohesion violations, and these translations tend to receive lower BLEU scores. This suggests that cohesion could be a strong feature in estimating the confidence of phrase-based translations. Our soft constraint produced improvements ranging between 0.5 and 1.1 BLEU points on sentences for which the baseline produces uncohesive translations. A human evaluation showed that translations created using a soft cohesion constraint are preferred over uncohesive translations in the majority of cases.

Acknowledgments Special thanks to Dekang Lin, Shane Bergsma, and Jess Enright for their useful insights and discussions, and to the anonymous reviewers for their comments. The author was funded by Alberta Ingenuity and iCORE studentships.

References

Y. Al-Onaizan and K. Papineni. 2006. Distortion models for statistical machine translation. In *COLING-ACL*, pages 529–536, Sydney, Australia.

C. Callison-Burch, M. Osborne, and P. Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *EACL*, pages 249–256.

C. Cherry and D. Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *COLING-ACL*, Sydney, Australia, July. Poster.

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

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *ACL*, pages 531–540.

J. Eisner. 2003. Learning non-ismorphic tree mappings for machine translation. In *ACL*, Sapporo, Japan. Short paper.

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

J. Graehl and K. Knight. 2004. Training tree transducers. In *HLT-NAACL*, pages 105–112, Boston, USA, May.

K. Knight. 1999. Squibs and discussions: Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, December.

P. Koehn and C. Monz. 2006. Manual and automatic evaluation of machine translation. In *HLT-NACCL Workshop on Statistical Machine Translation*, pages 102–121.

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

P. Koehn, A. Axelrod, A. Birch Mayne, C. Callison-Burch, M. Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*. Demonstration.

R. Kuhn, D. Yuen, M. Simard, P. Paul, G. Foster, E. Joanis, and H. Johnson. 2006. Segment choice models: Feature-rich models for global distortion in statistical machine translation. In *HLT-NAACL*, pages 25–32, New York, NY.

D. Lin and C. Cherry. 2003. Word alignment with cohesion constraint. In *HLT-NAACL*, pages 49–51, Edmonton, Canada, May. Short paper.

D. Lin. 1994. Principar - an efficient, broad-coverage, principle-based parser. In *COLING*, pages 42–48, Kyoto, Japan.

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

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL 2004: Main Proceedings*, pages 161–168.

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *ACL*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

C. Quirk, A. Menezes, and C. Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *ACL*, pages 271–279, Ann Arbor, USA, June.

C. Tillman. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL*, pages 101–104. Short paper.

A. Venugopal and S. Vogel. 2005. Considerations in maximum mutual information and minimum classification error training for statistical machine translation. In *EAMT*.

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

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

F. Xia and M. McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of Coling 2004*, pages 508–514.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *ACL*, pages 523–530.

R. Zens, H. Ney, T. Watanabe, and E. Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *COLING*, pages 205–211, Geneva, Switzerland, August.