Cohesive Constraints in A Beam Search Phrase-based Decoder

Nguyen Bach and Stephan Vogel
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{nbach, stephan.vogel}@cs.cmu.edu

Colin Cherry
Microsoft Research
One Microsoft Way
Redmond, WA, 98052, USA
collinc@microsoft.com

Abstract

Cohesive constraints allow the phrase-based decoder to employ arbitrary, non-syntactic phrases, and encourage it to translate those phrases in an order that respects the source dependency tree structure. We present extensions of the cohesive constraints, such as exhaustive interruption count and rich interruption check. Furthermore, we present analyses related to the impact of cohesive constraints across language pairs with different reordering models and dependency parsers. Our experiments show that the cohesion-enhanced decoder performs statistically significant better than the standard phrase-based decoder on English→Spanish. Improvements between 0.4 and 1.8 BLEU point are also obtained on English→Iraqi, Arabic→English and Chinese→English systems.

1 Introduction

Word movement is a defining characteristic of the machine translation problem. The fact that word order can change during translation makes the problem fundamentally different from related tasks such as tagging and automatic-speech-recognition. In fact, if one allows unrestricted changes in word order during translation, that alone is sufficient to show it to be NP complete, by analogy to the Traveling Salesman Problem (Knight, 1999). Despite the importance of movement, the popular phrase-based translation paradigm (Koehn et al., 2003) devotes surprisingly little modeling capacity to the issue. A very simple reordering model is to base the cost for word movement only the distance in the source sentence between the previous and the current word or phrase during the translation process. More recently, data-driven models, which condition the probability of phrase-to-phrase transitions on the words involved, have been proposed to address this issue (Tillman, 2004; Koehn et al., 2005; Al-Onaizan and Papineni, 2006; Kuhn et al., 2006; Galley and Manning, 2008).

Alternatively, one can employ syntax in the modeling of movement. By viewing language in terms of its hierarchical structure, one can more easily expose regularities in the sorts of movement that occur during translation. A number of syntactic methods are driven by formal syntax alone (Wu, 1997; Chiang, 2005), while others employ linguistic syntax derived from a parse tree (Galley et al., 2004; Quirk et al., 2005). Each of these approaches requires a parser-like decoder, and represents a departure from phrase-based decoding.

The well-studied phrase-based architecture can also benefit from syntactic intuitions. Phrasal decoding can be augmented easily, either by syntactic pre-processing or through search-space constraints. Pre-processing approaches parse the source sentence and use the tree to apply rules which re-order the source into a more target-like structure before the translation begins. These rules can be learned (Xia and McCord, 2004) or designed by hand (Collins et al., 2005; Wang et al., 2007). The pre-processing approach benefits from its simplicity and modularity, but it suffers from providing at most a one-best guess at syntactic movement. Search-space constraints limit the phrasal decoder's translation search using syntactic intuitions. Zens et al.(2004) demonstrated how to incorporate formally syntactic binary-bracketing constraints into phrase-based decoding. Recently, it has been shown that syntactic cohesion, the notion that syntactic phrases in the source sentence tend to remain contiguous in the target (Fox, 2002), can be incorporated into phrasal decoding as well, by following the simple intuition that any source subtree that has begun translation, must be completed before translating another part of the tree (Cherry, 2008; Yamamoto et al., 2008).

In this paper, we explore this approach, cohesive phrasal decoding, focusing on empirical issues left unexplored by previous investigations. Cherry (2008) proposed the notion of a soft cohesion constraint, where detected violations are allowed during decoding, but incur

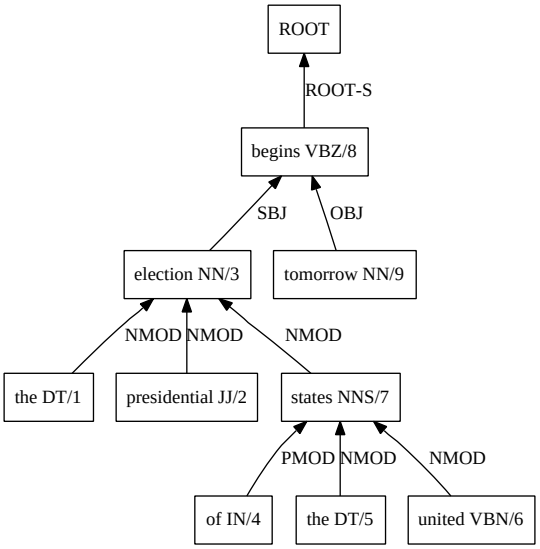


Figure 1: Example of an English source-side dependency tree structure for the sentence "the presidential election of the united states begins tomorrow".

a penalty. The flexibility of a soft penalty is appealing, given that cohesion does not perfectly characterize translation movement (Fox, 2002). But while cohesive decoding is well-defined for a hard constraint, soft constraints leave room for several design decisions. Should penalties persist as long as violations remain unresolved? Are some violations worse than others? Do cohesive constraints also improve systems that already benefit from large language models or lexical re-ordering models? We investigate these questions with a number of variant cohesive constraints. Furthermore, experimental results have so far been reported for English, French and Japanese only. We add to this body of work substantially, by experimenting with Spanish, Chinese, Iraqi and Arabic. Finally, we investigate the impact of the choice of parser and parse quality on cohesive decoding.

2 Cohesion Constraints

Phrase-based machine translation is driven by a phrasal translation model, which relates phrases (contiguous segments of words) in the source to phrases in the target. This translation model can be derived from a word-aligned bitext. Translation candidates are scored according to a linear model combining several informative feature functions. Crucially, the decoder incorporates translation model scores and n-gram language model scores. The component features are weighted to minimize a translation error criterion on a development set (Och, 2003). Decoding the source sentence takes the form of a beam search through the translation space, with intermediate states corresponding to partial translations. The decoding process advances by extending a state with the translation of a source phrase, until each source word has been translated exactly once. Re-ordering occurs when the source phrase to be translated does not immediately follow the previously translated phrase. This is penalized with a discriminatively-trained distortion penalty. In order to calculate the current translation score, each state can be represented by a triple:

- A coverage vector C indicates which source words have already been translated.
- A span $\bar{f}$ indicates the last source phrase translated to create this state.
- A target word sequence stores context needed by the target language model.

As cohesion concerns only movement in the source sentence, we can completely ignore the language model context in our description of the different cohesion constraints, i.e. we will show the decoder state only as a $(\bar{f}, C)$ tuple.

To enforce cohesion during the state expansion process, cohesive phrasal decoding has been proposed in (Cherry, 2008; Yamamoto et al., 2008). The cohesion-enhanced decoder enforces the following constraint: once the decoder begins translating any part of a source subtree, it must cover all the words under that subtree before it can translate anything outside of it. This notion can be applied to any projective tree structure, but we follow Cherry (2008) and use dependency trees, which have been shown to demonstrate greater cross-lingual cohesion than other structures (Fox, 2002). We use a tree data structure to store the dependency tree. Each node in the tree contains surface word form, word position, parent position, dependency type and POS tag. An example of the dependency tree data structure is shown in Figure 1. We use T to stand for our dependency tree, and $T(n)$ to stand for the subtree rooted at node n. Each subtree $T(n)$

Algorithm 1 Interruption Check (Coh1) (Cherry, 2008)

Input: Source tree T, previous phrase $\bar{f}_h$, current phrase $\bar{f}_{h+1}$, coverage vector C_h

$Interruption \leftarrow False$
$C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
$F \leftarrow$ the left and right-most tokens of $\bar{f}_h$
for each of $f \in F$ **do**
  Climb the dependency tree from f until you reach the highest node n such that $\bar{f}_{h+1} \notin T(n)$.
  if n exists and $T(n)$ is not covered in C_{h+1} **then**
    $Interruption \leftarrow True$
  end if
end for
Return $Interruption$

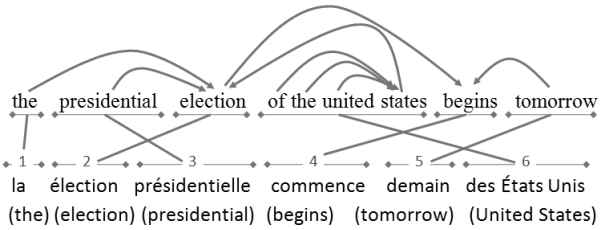


Figure 2: A candidate translation where Algorithm 1 does not fire

Algorithm 2 Exhaustive Interruption Check (Coh2)

Input: Source tree T, previous phrase f_h, current phrase f_{h+1}, coverage vector C_h

1: $Interruption \leftarrow False$
2: $C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
3: $F \leftarrow \{f | C_h(f) = 1\}$
4: **for** each of $f \in F$ **do**
5: Climb the dependency tree from f until you reach the highest node n such that $\bar{f}_{h+1} \notin T(n)$.
6: **if** n exists and $T(n)$ is not covered in C_{h+1} **then**
7: $Interruption \leftarrow True$
8: **end if**
9: **end for**
10: Return $Interruption$

covers a span of contiguous source words; for subspan $\bar{f}$ covered by $T(n)$, we say $\bar{f} \in T(n)$.

Cohesion is checked as we extend a state $(\bar{f}_h, C_h)$ with the translation of $\bar{f}_{h+1}$, creating a new state $(\bar{f}_{h+1}, C_{h+1})$. Algorithm 1 presents the cohesion check described by Cherry (2008). Line 3 selects focal points, based on the last translated phrase. Line 5 climbs from each focal point to find the largest subtree that needs to be completed before the translation process can move elsewhere in the tree. Line 6 checks each such subtree for completion. Since there are a constant number of focal points (always 2) and the tree climb and completion checks are both linear in the size of the source, the entire check can be shown to take linear time.

The selection of only two focal points is motivated by a "**violation free**" assumption. If one assumes that the translation represented by $(\bar{f}_h, C_h)$ contains no cohesion violations, then checking only the end-points of $\bar{f}_h$ is sufficient to maintain cohesion. However, once a soft cohesion constraint has been implemented, this assumption no longer holds.

2.1 Exhaustive Interruption Check (Coh2)

Because of the "violation free" assumption, Algorithm 1 implements the design decision to only suffer a violation penalty once, when cohesion is initially broken. However, this is not necessarily the best approach, as the decoder does not receive any further incentive to return to the partially translated subtree and complete it. For example, Figure 2 illustrates a translation candidate of the English sentence "the presidential election of the united states begins tomorrow" into French. We consider $\bar{f}_4$ = "begins", $\bar{f}_5$ = "tomorrow". The decoder already translated "the presidential election" making the coverage vector C_5 = "1 1 1 0 0 0 0 1 1". Algorithm 1 tells the decoder that no violation has been made by translating "tomorrow" while the decoder should be informed that there exists an outstanding violation. Algorithm 1 found the violation when the decoder previously jumped from "presidential" to "begins", and will not find another violation when it jumps from "begins" to "tomorrow".

Algorithm 3 Interruption Count (Coh3)

Input: Source tree T, previous phrase $\bar{f}_h$, current phrase $\bar{f}_{h+1}$, coverage vector C_h

1: $ICount \leftarrow 0$
2: $C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
3: $F \leftarrow$ the left and right-most tokens of $\bar{f}_h$
4: **for** each of $f \in F$ **do**
5: Climb the dependency tree from f until you reach the highest node n such that $\bar{f}_{h+1} \notin T(n)$.
6: **if** n exists **then**
7: **for** each of $e \in T(n)$ and $C_{h+1}(e) = 0$ **do**
8: $ICount = ICount + 1$
9: **end for**
10: **end if**
11: **end for**
12: Return $ICount$

Algorithm 2 is a modification of Algorithm 1, changing only line 3. The resulting system checks all previously covered tokens, instead of only the left and right-most tokens of $\bar{f}_h$, and therefore makes no violation-free assumption. For the example above, Algorithm 2 will inform the decoder that translating "tomorrow" also incurs a violation. Because $|F|$ is no longer constant, the time complexity of Coh2 is worse than Coh1. However, we can speed up the interruption check algorithm by hashing cohesion checks, so we only need to run Algorithm 2 once per $(\bar{f}_{h+1}, C_{h+1})$.

2.2 Interruption Count (Coh3) and Exhaustive Interruption Count (Coh4)

Algorithm 1 and 2 described above interpret an interruption as a binary event. As it is possible to leave several words untranslated with a single jump, some interruptions may be worse than others. To implement this observation, an interruption count is used to assign a penalty to cohesion violations, based on the number of words left uncovered in the interrupted subtree. For the example in Section 2.1, Algorithm 4 will return 4 for $ICount$ ("of";

Algorithm 4 Exhaustive Interruption Count (Coh4)

Input: Source tree T, previous phrase f_h, current phrase f_{h+1}, coverage vector C_h

1: $ICount \leftarrow 0$
2: $C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
3: $F \leftarrow \{f | C_h(f) = 1\}$
4: **for** each of $f \in F$ **do**
5: Climb the dependency tree from f until you reach the highest node n such that $\bar{f}_{h+1} \notin T(n)$.
6: **if** n exists **then**
7: **for** each of $e \in T(n)$ and $C_{h+1}(e) = 0$ **do**
8: $ICount = ICount + 1$
9: **end for**
10: **end if**
11: **end for**
12: Return $ICount$

Algorithm 5 Rich Interruption Constraints (Coh5)

Input: Source tree T, previous phrase $\bar{f}_h$, current phrase $\bar{f}_{h+1}$, coverage vector C_h

1: $Interruption \leftarrow False$
2: $ICount \leftarrow 0$
3: $VerbCount \leftarrow 0$
4: $NounCount \leftarrow 0$
5: $C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
6: $F \leftarrow$ the left and right-most tokens of $\bar{f}_h$
7: **for** each of $f \in F$ **do**
8: Climb the dependency tree from f until you reach the highest node n such that $\bar{f}_{h+1} \notin T(n)$.
9: **if** n exists **then**
10: **for** each of $e \in T(n)$ and $C_{h+1}(e) = 0$ **do**
11: $Interruption \leftarrow True$
12: $ICount = ICount + 1$
13: **if** POS of e is "VB" **then**
14: $VerbCount \leftarrow VerbCount + 1$
15: **else if** POS of e is "NN" **then**
16: $NounCount \leftarrow NounCount + 1$
17: **end if**
18: **end for**
19: **end if**
20: **end for**
21: Return $Interruption$, $ICount$, $VerbCount$, $NounCount$

"the"; "united"; "states"). The modification of Algorithm 1 and 2 lead to Interruption Count (Coh3) and Exhaustive Interruption Count (Coh4) algorithms, respectively. The changes only happen in lines 1, 6 and 7. We use an additional bit vector to make sure that if a node has been reached once, it is not counted again during the same interruption check.

2.3 Rich Interruption Constraints (Coh5)

The cohesion constraints in Sections 2.1 and 2.2 do not leverage node information in the dependency tree structures. We propose the rich interruption constraints (Coh5) algorithm to combine four constraints which are Interruption, Interruption Count, Verb Count and Noun Count. The first two constraints are identical to what was described above. Verb and Noun count constraints are enforcing the following rule: a cohesion violation will be penalized more in terms of the number of verb and noun words that have not been covered. For example, we want to translate the English sentence "the presidential election of the united states begins tomorrow" to French with the dependency structure as in Figure 1. We consider $\bar{f}_h$ = "the united states", $\bar{f}_{h+1}$ = "begins". The coverage bit vector C_{h+1} is "0 0 0 0 1 1 1 1 0". Algorithm 5 will return true for $Interruption$, 4 for $ICount$ ("the"; "presidential"; "election"; "of"), 0 for $VerbCount$ and 1 for $NounCount$ ("election").

3 Experiments

We built baseline systems using GIZA++ (Och and Ney, 2003), Moses' phrase extraction with the grow-diag-final-end heuristic (Koehn et al., 2007), a standard phrase-based decoder (Vogel, 2003), the SRI LM toolkit (Stolcke, 2002), the suffix-array language model (Zhang and Vogel, 2005), a distance-based word reordering model with a window of 3, and the maximum number of target phrases restricted to 10. Results are reported using lowercase BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). All model weights were trained on development sets via minimum-error rate training (MERT) (Venupopal and Vogel, 2005) with 200 unique n-best lists and optimizing toward BLEU. To shorten the training time, a multi-threaded GIZA++ version was used to utilize multi-processor servers (Gao and Vogel, 2008). We used the MALT parser (Nivre et al., 2006)[1] to obtain source English dependency trees and the Stanford parser for Arabic and Chinese (Marneffe et al., 2006). In order to decide whether the translation output of one MT engine is significantly better than another one, we used the bootstrap method (Zhang et al., 2004) with 1000 samples ($p < 0.05$). We perform experiments on English→Iraqi, English→Spanish, Arabic→English and Chinese→English. Detailed corpus statistics are shown in Table 1. Table 2 shows results in lowercase BLEU and TER; bold type is used to indicate highest scores. An italic text indicates the score is statistically significant better than the baseline.

The first step in validating the proposed approach was

[1]We would like to thank Johan Hall and Joakim Nirve for helpful suggestions on training and using the English dependency model

	English→Iraqi		English→Spanish		Arabic→English		Chinese→English	
	English	Iraqi	English	Spanish	Arabic	English	Chinese	English
sentence pairs	654,556		1,310,127		5,359,543		10,964,230	
unique sent. pairs	510,314		1,287,016		5,111,961		9,041,423	
avg. sentence length	8.4	5.9	27.4	28.6	25.7	29.7	24.9	28.1
# words	5.5 M	3.8 M	35.8 M	37.4 M	138 M	159 M	272.5 M	308.2 M
vocabulary	34 K	109 K	117 K	173 K	690 K	364K	1.4 M	845 K

Table 1: Corpus statistics of English→Iraqi, English→Spanish, Arabic→English and Chinese→English systems

	English→Iraqi		English→Spanish		Arabic→English				Chinese→English			
	june08		nct07		mt08-nw		mt08-wb		dev07-nw		dev07-wb	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
Baseline	23.58	61.03	32.04	49.97	48.53	45.03	33.77	56.30	25.14	62.32	23.65	61.66
+Coh1	24.45	58.89	*32.72*	49.18	48.78	44.92	34.15	**56.01**	26.46	61.04	23.95	**61.05**
+Coh2	**24.73**	58.75	*32.81*	49.02	48.47	45.23	**34.20**	56.42	***26.92***	61.24	23.92	61.45
+Coh3	24.19	59.25	*32.87*	48.88	48.70	44.84	33.91	56.29	26.3	61.46	**24.19**	61.51
+Coh4	24.66	**58.68**	*33.20*	48.42	**48.85**	**44.73**	33.86	56.38	26.73	**60.94**	24.03	61.42
+Coh5	24.42	59.05	***33.27***	**48.09**	48.57	45.07	34.10	56.37	26.05	61.69	23.76	61.52

Table 2: Scores on held-out evaluation sets of baseline and cohesion-enhanced systems for English→Iraqi, English→Spanish, Arabic→English and Chinese→English language pairs

to check if it works for the other language pairs. Our English-Iraqi data come from the DARPA TransTac program. The target domain is force protection which includes checkpoints and house-hold searches, and extends to civil affairs, medical, and training dialogs.

We used TransTac T2T July 2007 (july07) as the development set and TransTac T2T June 2008 (june08) as the held-out evaluation set. Each test set has 4 reference translation. We applied the suffix-array LM up to 6-gram with Good-Turing smoothing. In Table 2, cohesive constraints produced improvements ranging between **0.5** and **1.2** BLEU point on the held-out evaluation set.

We have shown that the proposed cohesion-enhanced decoder outperformed the baseline English→Iraqi systems. The English→Iraqi system used a small training size and came from force protection domain. The English→Iraqi pair also differs according to the language family. English is an Indo-European language while Iraqi is a Semitic language of the Afro-Asiatic language family. The next step in validating the proposed approach was to test on a language pair comes from the same Indo-European language family with a medium training size, different domain and written style.

We used the Europarl and News-Commentary parallel corpora for English→Spanish as provided in the ACL-WMT 2008[2] shared task evaluation. Detailed corpus statistics are given in Table 1. We built the baseline system using the parallel corpus restricting sentence length to 100 words for word alignment and a 4-gram SRI LM with modified Kneyser-Ney smoothing. We used nc-devtest2007(ncd07) as the development set and nc-test2007 (nct07) as the held-out evaluation set. Each test set has 1 translation reference. Table 2 shows that we obtained improvements ranging between **0.7** and **1.2** BLEU. All cohesive constraints perform **statistically significant** better than the baseline on the held-out evaluation set.

The previous results indicate that cohesive constraints contribute to the improvements of translation systems from English to other languages. However, many of today's high-profile translation tasks are concerned with translation into English. We experiment with the GALE data to test this other direction, and to examine cohesion's effect on competition-grade systems, which include other powerful movement features, such as large language models.

To validate these questions we present experimental results for the large-scale Arabic→English and Chinese→English systems. Unlike previous experiments, the source languages are Arabic and Chinese. Our Arabic-English and Chinese-English data come from the DARPA GALE program[3] and belong to newswire and broadcast news domain. Detailed corpus statistics are shown in Table 1. A 5-gram SRI LM was trained from the English Gigaword Corpus V3, which contains several newspapers for the years between 1994 and 2006. We also included the English side of the bilingual training data, resulting in a total of 2.7 billion running words after tokenization. For Arabic→English system we used NIST

[2]http://www.statmt.org/wmt08

[3]This training data was used in GALE P3 Evaluation

MT-06 as the development set and NIST MT-08 NW (mt08-nw) and WB (mt08-wb) as held-out evaluation sets. For Chinese→English system we used NIST MT-05 as the development set and Dev07Blind NW (dev07-nw) and WB (dev07-wb)[4] as held-out evaluation sets. Each test set has 4 reference translation. Table 2 shows results in BLEU and TER. The best improvements in BLEU we obtained are **0.3** on MT-08 NW and **0.4** on MT-08 WB for Arabic-English. We obtained **1.8** BLEU on Dev07Blind NW and **0.5** on Dev07Blind WB for Chinese-English over the baseline. Coh2 performed **statistically significant** better than the baseline system on Dev07Blind NW.

4 Discussion and Analysis

Experimental results of cohesive constraints on different language pairs have been described in Section 3, in this section we vary the ordering capability of the baseline system, and perform other forms of error analysis.

4.1 Interactions with reordering models

We first investigate the interactions of cohesive contraints with lexicalized reordering models on the performance of the translation system. The question we are trying to answer is whether the improvements of cohesive constraints are subsumed by a strong reordering model. Koehn et al. (2005) proposed the lexicalized reordering model which conditions reordering probabilities on the word of each phrase pair. The lexicalized reordering model has shown substantial improvements over the distance-based reordering model.

	dev07-nw		dev07-wb	
	BLEU	TER	BLEU	TER
Baseline	25.14	62.32	23.65	61.66
+Lex	26.07	61.56	23.68	61.71
+Lex+Coh1	26.52	62.00	24.47	61.69
+Lex+Coh2	**26.62**	**60.71**	24.95	**60.33**
+Lex+Coh3	26.53	61.62	**25.04**	61.06
+Lex+Coh4	26.53	60.86	24.79	60.69
+Lex+Coh5	26.35	60.74	24.88	60.44

Table 3: Performances of the GALE Chinese→English system with lexicalized reordering models in comparison with cohesion-enhanced systems

Table 3 shows the performance of Chinese→English system on the held-out evaluation set when we include lexicalized reordering models and cohesive constraints in the baseline system with a distance-based reordering model[5]. The system with lexicalized reordering model *+lex* gained over the baseline system by 0.9 BLEU point on dev07-nw set and performed similar on dev07-wb set. However, the performance of *+lex* is still weaker than most cohesive constraints in Table 2. Furthermore, when cohesive constraints are added on top of the lexicalized reordering model we observed a gain by **0.5** BLEU point on dev07-nw and a substantial gain by **1.4** BLEU on dev07-wb set. Coh2 model obtained best scores in most cases.

	dev07-nw		dev07-wb	
	BLEU	TER	BLEU	TER
Baseline	25.14	62.32	23.65	61.66
+Lex	26.07	61.56	23.68	61.71
+Lex+w5	26.21	61.06	24.87	60.84
+Lex+w5+Coh1	26.92	60.30	25.27	60.81
+Lex+w5+Coh2	**27.13**	**60.21**	25.12	60.95
+Lex+w5+Coh3	27.09	60.76	25.10	60.79
+Lex+w5+Coh4	26.79	60.50	**25.37**	**60.48**
+Lex+w5+Coh5	26.87	61.04	25.06	61.03

Table 4: Performances of the GALE Chinese→English system with lexicalized reordering models and reordering window 5 in comparison with cohesion-enhanced systems

After having empirical evidence for the improvements of cohesive constraints over systems with lexicalized reordering models, we insvetigate the impact of the reordering window. Table 4 demonstrates the translation performances of systems with different reordering limits and reordering models. The baseline system used distance-based reordering model with reordering window of 3. Meanwhile, *+lex* and *+lex+w5* used lexicalized reordering models with reordering window of 3 and 5, respectively. *+lex+w5* gained over the *+lex* system by 0.1 BLEU point on dev07-nw and 1.1 BLEU on dev07-wb. However, *+lex+w5* is still weaker than *+lex+Coh2* system in Table 3. We add cohesive constraints on top of *+lex+w5*. Cohesion-enhanced systems performed better than *+lex+w5* by **0.9** BLEU on dev07-nw and **0.5** BLEU point on dev07-wb.

4.2 The decoder behaviors

The cohesive constraints essentially act as filters on the generated hypotheses. As longer phrases can induce more cohesion violations, it is interesting to see how big an effect the different cohesive constraints have on the selection of phrases used in the final first best translation. The average length of phrases used in the translations is shown in Table 5. We see that indeed the cohesion constraints bias toward using shorter phrases.

We also analyzed how often a cohesion violation actually occurs under the different versions. Triple $(\bar{f}_h, \bar{f}_{h+1}, C_{h+1})$ can either trigger a cohesion violation or signal no violation independent of the actual translation generated. Therefore, we count the number of different triples and how many of them led to a cohesion vio-

[4]This test set is distributed by the GALE Rosetta team

[5]Note that we ran MERT separately for each system

	june-08	nc-test2007	mt08-NW	mt08-WB
Baseline	2.3	2.01	1.88	1.54
+Coh1	2.26	1.89	1.81	1.50
+Coh2	2.24	1.92	1.89	1.56
+Coh3	2.26	1.97	1.88	1.54
+Coh4	2.13	2.01	1.87	1.53
+Coh5	2.16	1.89	1.82	1.52

Table 5: The average length of phrases used in the translations

	june-08	nc-test2007	mt08-NW	mt08-WB
+Coh1	0.3896	0.4001	0.4786	0.4412
+Coh2	0.4305	0.4547	0.5198	0.4789
+Coh3	0.3887	0.3974	0.4777	0.4404
+Coh4	0.4304	0.4546	0.5198	0.4790
+Coh5	0.3916	0.4003	0.4852	0.4469

Table 6: Ratios between the number of times the interruption check fires and the total number of interruptions check in the different variants

lation. Results are summarized in Table 6. As expected, since Coh 2 and 4 perform exhaustive interruption checks they have higher ratio than the others. The ratios of Coh 1, 3 and 5 are close but not exactly the same because of hypothesis recombination and pruning during the decoding process. This is also true for the Coh 2 and 4.

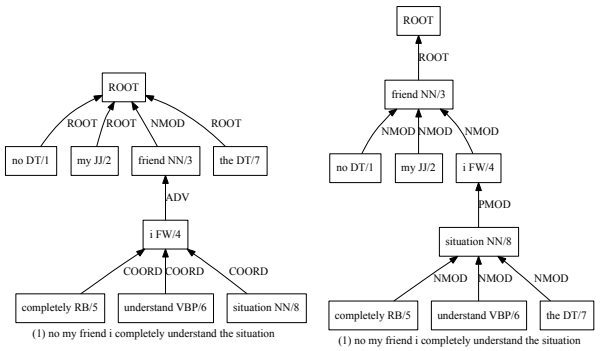
4.3 The role of dependency parser

We analyze the influence of the dependency parser on the performance of the translation system. We experimented with the MALT parser and the Stanford parser with default parameters on the English→Iraqi system described in Section 3. Performances on the unseen test set are shown in Table 7. Experimental results show that 1) either using MALT or Stanford parser the proposed approaches still outperform the baseline; 2) the MALT parser has a tendency to give better BLEU scores than the Stanford parser whereas the Stanford parser is faster than the MALT parser in our experimental setup.

	MALT Parser		Stanford Parser	
	BLEU	TER	BLEU	TER
Baseline	23.58	61.03	23.58	61.03
+ Coh1	24.45	58.89	24.17	58.79
+ Coh2	**24.73**	58.75	24.12	58.83
+ Coh3	24.19	59.25	24.37	58.81
+ Coh4	24.66	**58.68**	24.44	58.71
+ Coh5	24.42	59.05	23.99	59.55

Table 7: Comparison between using MALT parser and Stanford parser on English→Iraqi system

A general question of what quality of parser is required for cohesive constraints to work is important (Quirk and Corston-Oliver, 2006). To answer this question, we trained two MALT parser models, M1 and M2, on different sizes of Penn Treebank V3 data. The performances in term of unlabeled attachment score on the CoNLL-07 dependency test set are 19.41% and 86.21% for M1 and M2, respectively. Figure 3 illustrates difference dependency tree structures produced by M1 and M2 models. Table 8 shows the comparison of using M1 and M2 for English→Iraqi and English→Spanish systems. The results show that when applying these models to English→Iraqi, M1 performs better than M2 in most cases except Coh4. However, when the models are applied to English-Spanish then M2 is better than M1 in most cases except Coh2. The reason is that M1 and M2 models were only trained on Penn Treebank which belongs to newswire domain. M2's high performance on the newswire data has a positive effect on the Spanish test set, which is also drawn from a newswire domain. Meanwhile, the Iraqi defense text, which is quite different from newswire, seems to have no stable correlation with (newswire) parse quality, with M1 helping in some versions of the cohesion constraint, and M2 helping in others.



(a) M1 (b) M2

Figure 3: Dependency trees produced by M1 and M2.

	English→Iraqi		English→Spanish	
	M1	M2	M1	M2
Baseline	23.58	23.58	32.04	32.04
+ Coh1	24.16	23.86	31.92	32.29
+ Coh2	24.32	24.30	32.40	32.30
+ Coh3	24.23	24.06	31.89	32.60
+ Coh4	23.86	**24.54**	32.43	32.81
+ Coh5	24.26	24.22	32.53	**33.00**

Table 8: The impact of parser quality on the performance of English→Iraqi and English→Spanish systems in BLEU score

5 Conclusions and Future Work

In this paper, we explored cohesive phrasal decoding, focusing on variants of cohesive constraints. We pro-

posed four novel cohesive constraints namely exhaustive interruption check (Coh2), interruption count (Coh3), exhaustive interruption count (Coh4) and rich interruption constraints (Coh5). Our experimental results show that with cohesive constraints the system generates better translations in comparison with strong baselines. To ensure the robustness and effectiveness of the proposed approaches, we conducted experiments on 4 different language pairs, namely English→Iraqi, English→Spanish, Arabic→English and Chinese→English. These experiments also covered a wide range of training corpus sizes, ranging from 500K sentence pairs up to 10 million sentence pairs. Furthermore, the effectiveness of our proposed methods was shown when we applied them to systems using a 2.7 billion words 5-gram LM, different reordering models and dependency parsers. All five approaches give positive results. While the improvements are not statistically significant at the 95% level in most cases, there is nonetheless a consistent pattern indicating that the observed improvements are stable. The most reliable approach seems to be Coh2, a solution which does not make the violation free assumption.

In future work, we plan to apply cohesion constraints to learn reordering models. The cohesion constraints tell the decoder which cohesive movements are available, but the decoder has no opinion on how likely those moves are. A normal lexical reordering model is defined in terms of transitions between two phrases in sequence, $previous$ and $next$, which have a specific relationship to each other, such as $monotone$, $swap$, or $discontinuous$. Statistics on those relationships make up the lexical reordering model. The cohesion constraints, as described in this paper, can also be considered in terms of $previous$ and $next$. One can think of the check as checking the largest source subtree the decoder is leaving by transitioning from $previous$ source to $next$ source. Furthermore, linguistic analysis, such as root form, affixes, dependency types, and so on, can be used to define new cohesion constraints.

Acknowledgments

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL-COLING'06)*, pages 529–536, Sydney, Australia.

Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 72–80, Columbus, Ohio, June. Association for Computational Linguistics.

David Chiang. 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, June. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, USA, June.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*, pages 304–311, Philadelphia, PA, July 6-7.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, Hawaii, USA.

Michel Galley, Mark Hopkins, Kevin Knight, and Marcu Daniel. 2004. What's in a translation rule? In *the proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL-04)*, Boston, USA, May.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Proceedings of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*, Columbus, Ohio, USA.

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

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics of (HLT-NAACL'03)*, pages 48–54, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'05)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, demo sessions*, pages 177–180, Prague, Czech Republic, June.

Roland Kuhn, Denis Yuen, Michel Simard, Patrick Paul, George Foster, Eric Joanis, and Howard Johnson. 2006. Segment choice models: Feature-rich models for global distortion in statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL'06)*, pages 25–32, New York, NY.

Marie-Catherine Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC2006)*, Genoa, Italy.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC2006)*, Genoa, Italy.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 1:29, pages 19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Conf. of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, July.

Chris Quirk and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, Sydney, Australia.

Chris Quirk, Aruk Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, USA, June.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA'06)*, pages 223–231, August.

Andreas Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904, Denver.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL'04)*, pages 101–104. Short paper.

Ashish Venupopal and Stephan Vogel. 2005. Considerations in maximum mutual information and minimum classification error training for statistical machine translation. In *In the Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05)*, Budapest, Hungary.

Stephan Vogel. 2003. SMT decoder dissected: Word reordering. In *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'03)*, pages 561–566, Bejing, China, Oct.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'07)*, pages 737–745.

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

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics (COLING'04)*, pages 508–514, Geneva, Switzerland, August.

Hirofumi Yamamoto, Hideo Okuma, and Eiichiro Sumita. 2008. Imposing constraints from the source tree on ITG constraints for SMT. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 1–9, Columbus, Ohio, June. Association for Computational Linguistics.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of the 20th international conference on Computational Linguistics (COLING'04)*, pages 205–211, Geneva, Switzerland, August.

Ying Zhang and Stephan Vogel. 2005. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT'05)*, Budapest, Hungary, May. The European Association for Machine Translation.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 2051–2054.