Measuring the Divergence of Dependency Structures Cross-Linguistically to Improve Syntactic Projection Algorithms

Ryan Georgi[†], **Fei Xia**[†], **William D. Lewis**[*]

University of Washington[†], Microsoft Research[*]
rgeorgi@uw.edu, fxia@uw.edu, wilewis@microsoft.com

Abstract

Syntactic parses can provide valuable information for many NLP tasks, such as machine translation, semantic analysis, etc. However, most of the world's languages do not have large amounts of syntactically annotated corpora available for building parsers. Syntactic projection techniques attempt to address this issue by using parallel corpora between resource-poor and resource-rich languages, bootstrapping the resource-poor language with the syntactic analysis of the resource-rich language. In this paper, we investigate the possibility of using small, parallel, annotated corpora to automatically detect divergent structural patterns between two languages. These patterns can then be used to improve structural projection algorithms, allowing for better performing NLP tools for resource-poor languages, in particular those that may not have large amounts of annotated data necessary for traditional, fully-supervised methods. While this detection process is not exhaustive, we demonstrate that important instances of divergence are picked up with minimal prior knowledge of a given language pair.

Keywords: Multilinguality, Translation Divergence, Syntactic Projection

1 Introduction

The past two decades have seen significant progress in all fields of NLP. Unfortunately, most studies have focused on a select handful of "resource-rich" languages, with the vast majority of the world's languages un-studied or understudied. Work on resource-rich languages has benefitted from the availability of annotated data resources, such as treebanks or engineered grammars. The use of such annotated resources has resulted in many state-of-the-art systems, but the underlying work, specifically the annotation of corpora used to train tools, has required significant work to be done on a per-language basis. Because of the costs inherent in doing work, the vast majority of the world's resource-poor languages still lack high-performance NLP tools. One manner of addressing this lack of annotated data is to bootstrap annotation on an resource-poor language using annotation from a resource-rich one. This is done by "projecting" syntactic information such as part-of-speech tags or syntactic structures by means of word alignment and parallel corpora.

In this paper, we propose a method for analyzing a language pair and determining the degree and types of divergence between two languages. This systematic identification of divergence types could then lead to better informed syntactic projections, and subsequently can improve the tools built upon such data.

2 Previous Work

The potential benefits of syntactic projection have been demonstrated by Yarowsky and Ngai (2001) and Hwa et al. (2004), who show that taggers and parsers trained on data created by projection algorithms, while exhibiting performance below that of fully-supervised methods, still establish a valuable starting point for resource building and adaptation. Lewis and Xia (2008) used projected phrase structures to determine the basic word order for 97 languages using a database of Interlinear Glossed Text (IGT), where an IGT instance includes a source sentence, a gloss line, and a translation (usually in English) of the source sentence, as illustrated in Figure 1. They automatically aligned the three lines in IGT, used the alignment to project syntactic structures from the translation line to the source line, and then inferred the word order in the source language. For languages with just 10–39 IGT instances, the accuracy of predicting basic word order was 79%, and 99% for languages with more than 40 instances.

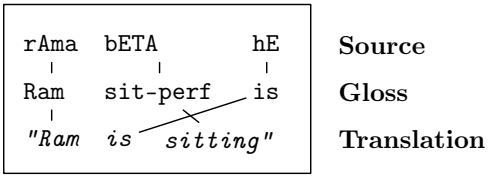


Figure 1: An instance of Interlinear Glossed Text (IGT) for Hindi, and the alignment information it supplies between Hindi and English.

These studies illustrate the promise of projection for bootstrapping new tools in resource-poor languages, but are limited by a reliance on the assumption that syntactic structures of the two sentences in a given sentence pair are similar. Hwa et al. (2002) labeled this assumption the *Direct Correspondence Assumption*, or DCA, and Dorr (1994) goes into depth about a number of cases that contradict this assumption. In parallel dependency treebanks, which will be used and examined in this paper, while DCA often holds, it is well-known that exceptions exist. The question is: how often exceptions occur and whether there are patterns for the exceptions? That is the focus of this paper.

3 Methodology

Dorr (1994) identifies a number of ways in which languages may diverge, specifically syntactic and semantic differences in mappings between languages. Our goal in this work is to create a methodology by which some common types of divergences can be detected automatically from bitext in order to improve the performance of existing structural projection methods.

Our approach has several steps. First, we propose a metric to measure the degree of *match* between source and target trees (§3.1). Second, we define three operations on trees in order to capture three common types of divergence (§3.2). Third, we apply the operations on a tree pair and show how the operations could affect the degree of tree *match* (§3.3). After explaining the relation of our operations to Dorr's divergence types (§3.4), we discuss how knowing these divergence types can be useful in improving structural projection algorithms (§3.5).

3.1 Comparing Dependency Trees

One of the key aspects of our method was devising a metric to compare dependency trees cross-linguistically, as most existing tree similarity measures, such as F-score or dependency accuracy, are intended to score trees representing the same sentence (e.g., a tree produced by a parser and the gold standard for the sentence). We, on the other hand, would like to compare two trees for a sentence pair. A *sentence pair* is a pair of sentences which are translations of each other, and the dependency trees for the two sentences in a sentence pair form a *tree pair*. Given that the trees in a tree pair are for sentences in two different languages, we must make use of word alignment as a means to find correspondence between the nodes in the two trees.

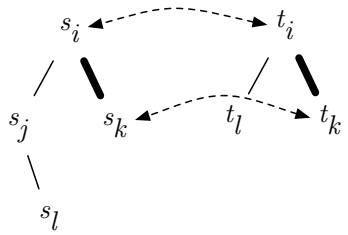


Figure 2: Definition of a *matched* edge in a tree pair

We measure similarity of two trees by counting the percentage of *matched* edges in the two trees according to a word alignment. As shown in Figure 2, an edge $\langle s_i, s_k \rangle$ in the source dependency tree is said to *match* an edge $\langle t_i, t_k \rangle$ in the target tree if s_i is aligned to t_i and s_k is aligned to t_k. Given a tree pair (S, T), we define the matches from S to T, $match(S \rightarrow T)$, as the percent of edges in S that match some edge in T. Similarly, $match(T \rightarrow S)$ is defined to be the percentage of edges in T that match some edges in S. Because the numbers of edges in S and T are often different, the $match$ function is not symmetric.

Given a parallel treebank, (L_S, L_T), we define $match(L_S \rightarrow L_T)$ as the percentage of edges in L_S that match some edge in the corresponding target trees in L_T.

3.2 Defining Tree Operations

When an edge $\langle s_i, s_k \rangle$ in a tree does not match any edge in the target tree, there are three very common cases:

C1. s_i or s_k is a spontaneous word. Given a sentence pair, a word is *spontaneous* if it does not align to any other word in the other sentence.

C2. s_i and s_k are both aligned with the same node t_i in the other tree (see Fig 3(a)).

C3. s_i and s_k are aligned to two nodes in the other tree, but the direction of dependency is reversed on the other tree (see Fig 3(b)).

To understand the effect of these cases on $match(L_S \rightarrow L_T)$, we define three operations on a tree — *remove*, *merge*, and *swap*.

O1 – Remove a Node

The *remove* operation removes a node from a tree. If the node is a leaf node, the operation simply deletes the node. If the node is an internal node, the operation removes the edge between the node and its parent j, and makes the node's children new children of j, as shown in Figure 4(a) and Definition 1 in Table 1.

O2 – Merge a Node with its Parent

The *merge* operation is used to collapse a child node l with its parent j. While conceptually the merged node should be called *l+j*, for the sake of simplicity we still call it *j*. As a result, merging a node *l* with its parent *j* is the same as removing *l*, as in Figure 4(b) and Definition 2 in Table 1. Nevertheless, the two operations have different purposes; they are used to handle Case C1 and C2, respectively (see Table 2).

O3 – Swap a Node with its Parent

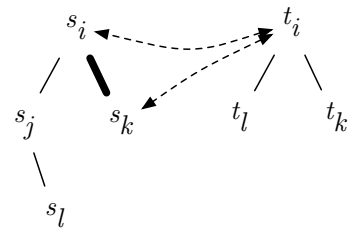
The *swap* operation reverses the dependency relation between a node l and its parent j; that is, after the swap operation, j becomes a child of l (see Figure 4(c) and Definition 3 in Table 1).

This operation can be used to handle certain divergence types such as *demotional* and *promotional* divergence, discussed in further detail in §3.4.

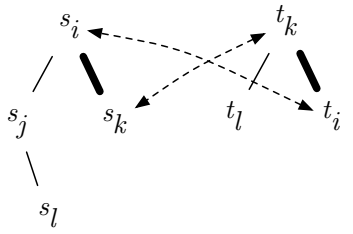
3.3 Calculating Tree Matches After Applying Operations

The operations O1–O3 are proposed to handle common divergence cases in C1–C3. To measure how common C1–C3 is in a language pair, we design an algorithm that transforms a tree pair based on a word alignment.

The algorithm takes a tree pair (S, T) and a word alignment A as input and creates a modified tree pair (S', T') and an updated word alignment A' as output. It has several steps. First, spontaneous nodes (nodes that do not align to any node on the other tree) are removed from each tree. Next, if a node and its parent align to the same node on the other tree, they are merged and the word alignment is changed accordingly. Finally, the swap operation is applied to a node v_s and its parent p_s in one tree if they align to v_t and c_t respectively and c_t is a child of v_t in the other tree. The pseudocode of the algorithm is shown in Table 2.

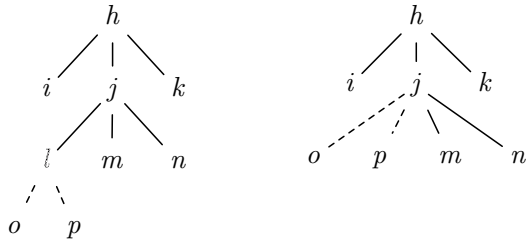


(a) A "merge" alignment

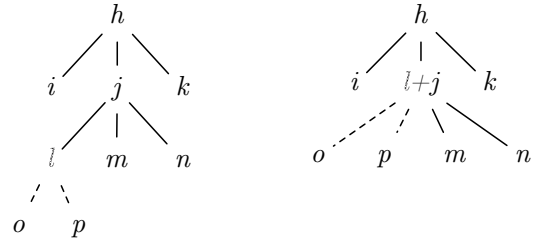


(b) A "swap" alignment

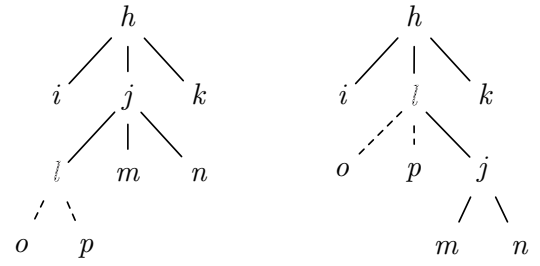
Figure 3: Examples of different alignment types that can be systematically detected.



(a) Before and after the node l has been removed (O1).



(b) Before and after the nodes l and j have been merged (O2). The merged node *l+j* is actually still labeled as *j* in Tables 1 and 2 for the sake of simplicity.



(c) Before and after the nodes l and j have been swapped (O3).

Figure 4: Trees showing the results of the operations defined in O1–O3. Dotted lines indicate edges between a modified node and its children.

Now given a parallel treebank (L_S, L_T) and word alignment between each sentence pair, we can measure the impact of C1–C3 by comparing $match(L_S \rightarrow L_T)$ scores before and after applying operations O1–O3. This process can also reveal some patterns of divergence (e.g., what types of nodes are often merged), and the patterns can later be used to enhance existing projection algorithms.

Definition 1: Remove a node l.
Let $G = (V, E)$ be the original graph $Remove(l, G) = (V', E')$ **where** $V' = V - \{l\}$ **and** $E' = E - \{(a, l) \mid a = parent(l, G)\}$ $-\{(l, b) \mid l = parent(b, G)\}$ $+\{(a, b) \mid a = parent(l, G), l = parent(b, G)\}$
Definition 2: Merge a node l with its parent j.
Let $G = (V, E)$ be the original graph $Merge(l, j, G) = (V', E')$ **where** $V' = V - \{l\}$ **and** $E' = E - \{(j, l)\}$ $-\{(l, b) \mid l = parent(b, G)\}$ $+\{(j, b) \mid l = parent(b, G)\}$
Definition 3: Swap a node l with its parent j.
Let $G = (V, E)$ be the original graph $Swap(l, j, G) = (V, E')$ **where** $E' = E - \{(j, l)\} + \{(l, j)\}$ $-\{(a, j) \mid a = parent(j, G)\}$ $+\{(a, l) \mid a = parent(j, G)\}$

Table 1: Definitions for the $Remove$, $Merge$, and $Swap$ operations. Here, $parent(i, G)$ returns the parent node of i in Graph G.

3.4 Relationship to Dorr (1994)

Dorr (1994) lists seven types of divergence for language pairs. While our analysis method is more coarse-grained than the Lexical Conceptual Structure (LCS) that Dorr proposes, it nonetheless is able to capture some of the same cases.

For instance, Figure 5 illustrates an example of what Dorr (1994) identified as "promotional" divergence, where *usually*, a dependent of the verb *goes* in English, is "promoted" to become the main verb, *suele* in Spanish. In this case, the direction of the dependency between *usually* and *goes* is reversed in Spanish, and thus the *swap* operation can be applied to the English tree and result in a tree that looks very much like the Spanish tree. A similar operation is performed for *demotional* divergence cases, such as aligning "I like eating" with the German translation *"Ich esse gern"* ("I eat likingly"). Here, the main verb in English ("like") is *demoted* to an adverbial modifier in German ("*gern*"). The

Algorithm: Alter tree pairs based on word alignment.

input: A tree pair (S, T)
Word alignment A between S and T
output: Altered tree pair (S', T')
Altered alignment A'
Let $G_S = (V_S, E_S)$ be the graph for S;
Let $G_T = (V_T, E_T)$ be the graph for T;
Let A be the set of alignment edges $\{(v_s, v_t)\}$;

// Step 1(a): Remove spontaneous nodes from S
foreach $v_s \in V_S$ **do**:
  if $\neg\exists x : (v_s, x) \in A$ **then**:
    $G_S = Remove(v_s, G_S)$;
// Step 1(b): Remove spontaneous nodes from T
foreach $v_t \in V_T$ **do**:
  if $\neg\exists x : (x, v_t) \in A$ **then**:
    $G_T = Remove(v_t, G_T)$;
// Step 2(a): Find nodes to merge in S and merge them
foreach $(v_s, v_t) \in A$ **do**:
  let $p_s = parent(v_s, G_S)$
  if $(p_s, v_t) \in A$ **then**:
    $G_S = Merge(v_s, p_s, G_S)$;
    $A = A - \{(v_s, v_t)\}$;
// Step 2(b): Find nodes to merge in T and merge them
foreach $(v_s, v_t) \in A$ **do**:
  let $p_t = parent(v_t, G_T)$
  if $(v_s, p_t) \in A$ **then**:
    $G_T = Merge(v_t, p_t, G_T)$;
    $A = A - \{(v_s, v_t)\}$;
// Step 3: Find nodes to swap in S and swap them
foreach $(v_s, v_t) \in A$ **do**:
  let $p_s = parent(v_s, G_S)$
  if $\exists c_t : v_t = parent(c_t, G_T)$ and $(p_s, c_t) \in A$ **then**:
    $Swap(v_s, p_s, G_S)$;
return (S, T, A)

Table 2: Algorithm for altering a tree pair (S, T) based on word alignment A.

swap operation is applicable to both types of divergence and treats them equivalently, and so it essentially can handle a superset of promotional and demotional divergence, namely,"head-swapping."

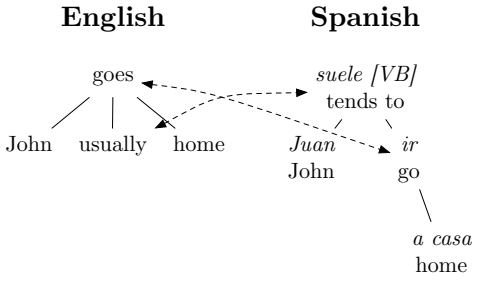


Figure 5: An example of *promotional* divergence from Dorr (1994). The reverse in parent-child relation is handled by the $Swap$ operation.

Another type of divergence that can be captured by our approach is Dorr's "structural" divergence type, as illustrated in Figure 6. The difference between the English and Spanish structures in this case is the form of the argument that the verb takes. In English, it is a noun phrase; in Spanish, it is a prepositional phrase. While the tree operations defined previously do not explicitly recognize this difference in syntactic labels, the divergence can be handled by the *remove* operation, where the spontaneous *"en"* in the Spanish side is removed.

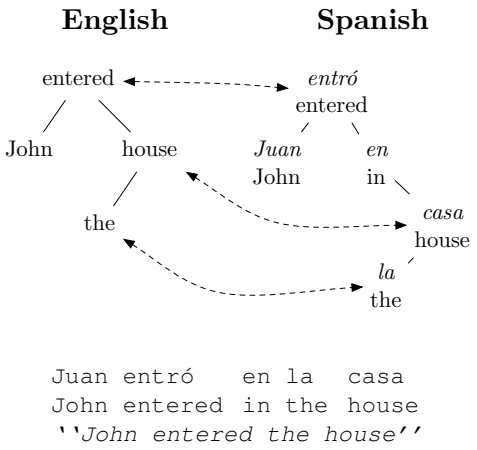


Figure 6: Example of structural divergence, which is handled by the *remove* operation.

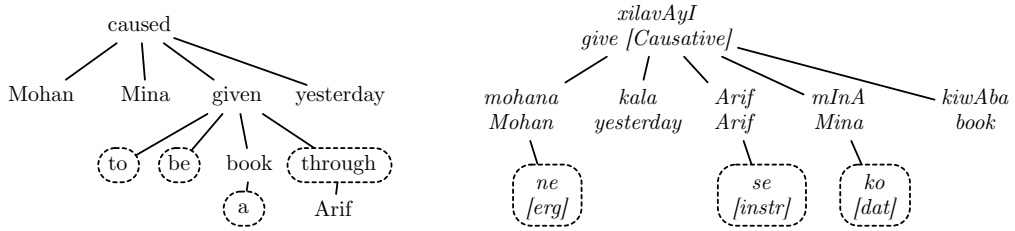
Next, Dorr's description of *conflational* divergence lines up well with the *merge* operation (*see Fig 4(b)*). Figure 7 illustrates an example for English and Hindi, where both sides have spontaneous words (e.g., *to* and *a* in English) and a causative verb in Hindi corresponds to multiple verbs in English. Figure 7(b) shows the original tree pairs, 7(c) demonstrates the altered tree pairs after removing spontaneous words from both sides. Figure 7(d) shows the tree pairs after the English verbs are merged into a single node. It is clear that the *remove* and *merge* operations make the Hindi and English trees much similar to each other.

In addition to the four divergence types mentioned above, additional operations could be added to handle other divergence types in Dorr (1994). For instance, if dependency types (e.g. patient, agent) are given in the dependency structure, we can define a new operation that changes the dependency type of an edge to account for *thematic* divergence, where thematic roles are switched as in "I like Mary" in English vs. *"María me gusta a mí"* (Mary pleases me) in Spanish. Similarly, an operation that changes the POS tag of a word can be added to to cover *categorial* divergence where words representing the same semantic content have different word categories in the two languages, such as in "I am hungry" in English versus *"Ich habe Hunger"* (I have hunger) in German.

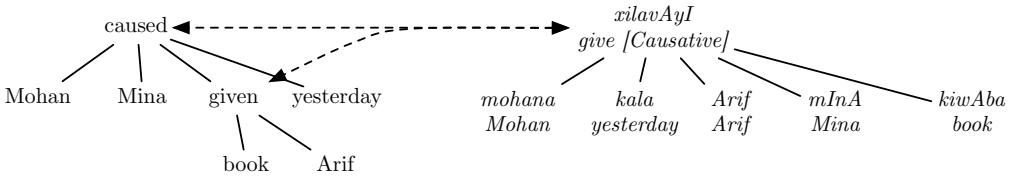
Compared to Dorr's divergence types, whose identification requires knowledge about the language pairs, our operations on the dependency structure relies on word alignment alone and can be applied automatically.

```
mohana ne    kala       Arif se     mInA ko    kiwAba     xilavAyI
Mohan  [erg] yesterday  Arif [inst] Mina [dat] book       give-caus
''Mohan caused Mina to be given a book through Arif yesterday.''
```

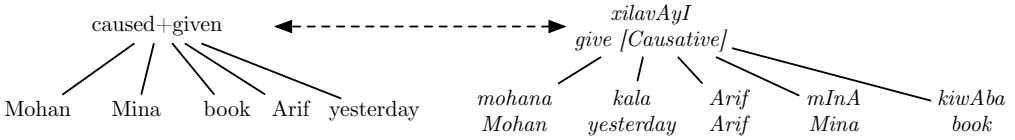
(a) Interlinear text of a sentence pair.



(b) Initial trees showing spontaneous words on both sides.



(c) Altered trees after removing spontaneous words from both sides, and showing conflational divergence between multiple English words and a single Hindi word.



(d) Altered trees after merging multiple words on the English side.

Figure 7: Case of conflational divergence from English due to complex Hindi morphology.

3.5 Extending Projection Algorithms

The projection algorithm as described in Xia and Lewis (2007) uses heuristics to handle spontaneous words and many-to-one word alignments, because the algorithm does not assume prior knowledge about either language. Furthermore, it does not handle head swapping because the identification of head swapping requires access to dependency structures on both sides whereas their projection algorithm assumes that it has access to the syntactic structure on only one side.

Our method couples the divergent cases C1–C3 with corresponding operations O1–O3. As the operations are applied, statistics are kept on the nodes that are affected, and thus common divergence patterns can be detected by analyzing this data. By generalizing the data found in this analysis, rules that can handle common divergence types could be applied to particular language pairs that exhibit such patterns in the small training corpus. Some preliminary patterns found in our work on four language pairs is discussed in §4.3.

4 Experiments

With the matching function and tree operations defined in the previous section, we looked at a total of four language pairs: English–Hindi, English–German, English–Swedish, and German–Swedish, using the corpora in Table 3.

4.1 Data

Corpus	Languages	#Sents	#Words
SMULTRON	English	111	1196
	German		1124
	Swedish		1050
Hindi Treebank	English	147	943
	Hindi		963

Table 3: Summary of corpora used in our experiment, where #sents and #words refers to the number of sentences and words in each language.

Our work utilizes two corpora, the SMULTRON treebank (Volk et al., 2010) and guideline sentences in IGT form from the Hindi treebank (Bhatt et al., 2009). The statistics of the corpora are shown in Table 3.

In the SMULTRON Treebank, the German and Swedish phrase trees are marked for head children, allowing for the automatic extraction of dependency trees. The English side of the phrase structures do not contain edge labels and are

Eng→Hin	Hin→Eng	Operations
47.5%	46.2%	Baseline
65.9%	63.3%	After $Remove$
69.2%	69.2%	After $Merge$
90.0%	90.0%	After $Swap$
Eng→Swe	Swe→Eng	Operation
37.5%	41.9%	Baseline
58.2%	68.6%	After $Remove$
68.4%	68.6%	After $Merge$
78.0%	78.2%	After $Swap$
Eng→Ger	Ger→Eng	Operation
40.7%	43.4%	Baseline
59.5%	67.2%	After $Remove$
66.6%	67.2%	After $Merge$
77.0%	77.8%	After $Swap$
Ger→Swe	Swe→Ger	Operation
43.3%	45.1%	Baseline
69.1%	77.4%	After $Remove$
77.3%	77.4%	After $Merge$
81.4%	81.4%	After $Swap$

Table 4: Percentages of matches after various operations are performed. Each step shows the values of $match(A \rightarrow B)$ and $match(B \rightarrow A)$ for the language pair after the step for a given operation in Table 2 has been performed.

instead converted to dependency trees using a head percolation table (Collins, 1999).

From the Hindi Treebank guidelines, we extracted example sentences in the form of IGT (i.e., Hindi sentences, English gloss, and English translation) and the Hindi dependency structures manually created by the guideline designers. We obtained dependency structures for the English translation by running the Stanford dependency parser (de Marneffe et al., 2006) and then we hand corrected the structures. Word alignment is initially derived from the IGT instances using heuristic alignment following (Xia and Lewis, 2007), and later hand-corrected.

4.2 Match Results

By performing the algorithm in Table 2, we can calculate the $match(S \rightarrow T)$ and $match(T \rightarrow S)$ for every tree pair in our parallel treebanks before and after each operation and see the effect each operation has. As the operations are applied, the percentage of matches between the trees should increase until all the divergence cases that can be handled by operations O1–O3 have been resolved. At this point, the final match percentage can be seen as an estimate of the performance of a simple projection algorithm, if C1-C3 can be identified and handled by O1-O3. The results are shown in Table 4.

The baseline represents the percentage of matches in the trees before any operations have been applied, and is consistently below 50%. After removing spontaneous words, the percentage of matches goes up significantly for all the language pairs. The *merge* and *swap* steps increase the percentage further. After all operations are performed, there are still 10–23% of edges in the trees unaccounted for. These remaining cases are discussed in §4.4.

In addition to providing this estimate about how common C1-C3 are in a language pair, this similarity-maximizing process can also be used to keep statistics on the kinds of elements that are affected, which will be discussed in §4.3.

4.2.1 Discussion of Match Results

The match results themselves are informative, but limited in their estimate of language similarity. Although the English, German, and Swedish data all come from the SMULTRON corpus, the Hindi data source is significantly different, and thus its higher match score should not be interpreted as indicating a higher correlation between Hindi and English than English and the other languages. With more comparable corpora, however, the similarities should be more evident.

Of more significance, perhaps, is the identification of certain issues in the design of the treebanks. The Hindi Treebank shows an extremely large jump after the *swap* operation has been performed. A main contributor of this jump is due to the treatment of adpositions. In Hindi, adpositions are expressed as postpositions following the noun phrase, and the postposition depends on the head of the noun phrase. This is the reciprocal of the English convention of placing nouns preceded by preposition as the dependents of the preposition. Whether the difference in the treatment of adposition reflects true structural divergence in the language pair or merely a design choice in the annotation guidelines is a topic for debate.

4.3 Discussion of Patterns

While the match statistics are interesting and provide some rough insights into the language pairs, the ultimate goal of this work is to discern more concrete patterns between the language pairs. During the match identification process, statistics are gathered on the affected nodes, and some of the more salient patterns can be seen in Table 6.

4.3.1 Hindi↔English Patterns

Row 1 in Table 6 shows a clear example of a pattern of conflational divergence illustrated in Figure 7, where multiple base verbs are merged with the inflected main verb being conflated with a single verb in Hindi. Hindi, on the other hand, exhibits its own conflational pattern in row 4, combining N+V in 20% of merge cases. This pattern is consistent with instances of noun + light verb patterns, e.g., *John stole a book* in English is expressed as *John theft book did* in Hindi. Row 3 shows that another 65% of merges

Algorithm: Match Scoring

input: A tree pair (S, T) and an alignment A between them
output: Score for $Match(S \rightarrow T)$
Let $G_S = (V_S, E_S)$ be the graph for S;
Let $G_T = (V_T, E_T)$ be the graph for T;
Let A be the set of alignment edges;
$matches = 0$;
foreach $(p_s, c_s) \in E_S$ **do**:
  if $\exists\, p_t, c_t : p_t = parent(c_t, G_T)$ **and** $(p_s, p_t) \in A$ **and** $(c_s, c_t) \in A$ **then:**
    $matches$++;
return $100 \times \frac{matches}{|E_S|}$;

Table 5: Algorithm for scoring matches between trees.

Merges				
Lang Pair	Row #	Child POS	Parent POS	% Merges
Eng→Hin	1	VB	VBD	38.4%
	2	"To"	V	17.7%
Hin→Eng	3	VAUX	V	65.0%
	4	N	V	20.0%
Eng→Ger	5	N	N	52.5%
	6	ADJ	N	15.8%
Swaps				
Eng→Hin	7	V	Gerund	28.0%
	8	V	V	20.0%
Eng→Ger	9	DT	N	18.8%
	10	N	N	15.6%
Removals				
Eng→Hin	11	DT		55.6%
	12	TO		18.1%
Hin→Eng	13	PSP		67.5%
	14	VAUX		8.9%
Eng→Ger	15	IN		30.4%
	16	DT		10.8%
Ger→Eng	17	ART		26.8%
	18	APPR		18.5%

Table 6: Breakdown of significant merge and swap statistics for various language pairs, where the language to the left of the arrow is the one being altered.

were cases where Hindi represented the tense of a verb as an auxiliary verb, whereas in such cases in English tense is expressed by inflections applied to the verb.

Spontaneous words patterned similarly: 67% of removals were case markers (PSP) in Hindi that were either absent in English or applied as inflections to the noun (Row 13), while 55% of the spontaneous removals in English were definite or indefinite determiners, which are not used in Hindi (Row 11).

4.3.2 English↔German Patterns

The English↔German pair comes from the SMULTRON data. Beyond the variations one would expect in examining a different language pair (e.g., English-German versus English-Hindi), SMULTRON's annotations are somewhat divergent from those used treebank, and the granularity of annotation also differs.

The most obvious pattern from this pair is the Noun–Noun and Adj–Noun merges in the English side to accommodate the large amount of compounding that occurs in German (Rows 5 and 6), particularly in the economic domain that much of the SMULTRON data was taken from. Taken together, these merges account for over two-thirds of all the merges in the English side.

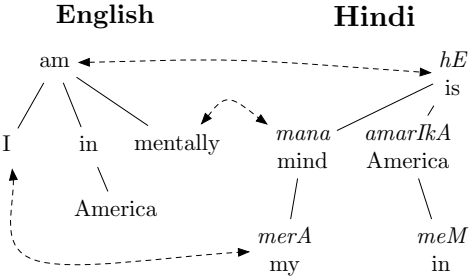
In terms of swaps, the picture is a little less clear, with English determiners taking the place of nouns (Row 9). Upon further inspection of the data, this appears to be primarily triggered by a particular case in German, the use of the preposition "*im*" — a word that actually is a contraction of "*in dem*", a preposition and a determiner. Due to the contraction, "*im*" is identified in the German treebank as a head, and the alignment of the English determiner with this ends up in a swap. The Noun–Noun case (Row 10) is a little more straightforward, swapping nouns in the English side that are headed differently in German due to divergent ways of expressing noun phrases.

Finally, removals in the English↔German data are also somewhat idiosyncratic. The removal of prepositions in English (Row 15) appears to be caused by and large by the preposition "of" in English, which is simply not necessary in the German translations, such as "*Zusammengefasste Ergebnisse*" versus "*Summary of results*." The spontaneous elements in the German side (Rows 17-18) are largely articles and circumposition particles which do not directly have corresponding English words.

All the patterns in Table 6 are collected automatically by extending the algorithm in Table 2 to record what kind of nodes have been removed, merged, or swapped. While some interpretation has been done here for clarity, it is easy to see how a projection algorithm could benefit from those patterns without requiring specific knowledge of the language pair.

4.4 Remaining Cases

Figure 8 shows a tree pair that would still have unmatched edges after the three operations have been applied.[1] The dependency edge *(in, America)* can be reversed by the *swap* operation to match the Hindi counterpart. The difficult part is the adverb *mentally* in English corresponds to the noun *mana (mind)* in Hindi. If the word alignment includes the three word pairs as indicated by the dotted lines, one potential way to handle this kind of divergence is to extend the definition of *merge* to allow edges to be merged on both sides simultaneously – in this case, merging *am* and *mentally* in the English side, and *hE (is)* and *mana (mind)* on the Hindi side.




```
merA mana amarIkA meM hE
my   mind America in  is
''I am mentally in America''
```

Figure 8: A tree pair that still has unmatched edges after applying the algorithm in Table 2. The dotted line indicates word alignment that would be needed to resolve the divergence with the *extended* merge operation.

5 Discussion

Two large issues that our methodology faces are data sparsity and translation quality of the sentence pairs in the data sets. The former is somewhat inevitable given the task—a reasonable amount of annotated data is not always likely to exist for languages with scarce electronic resources, and guaranteeing coverage is difficult. As with the Hindi data, however, using IGT as a resource has convenience in both covering wide varieties of phenomena in

[1] It is a topic of debate whether *mentally* in English should depend on *in* or *am*. If it depends on *in*, handling the divergence would be more difficult.

a language, and providing a gloss that assists in creating word-level alignments. Creating dependency annotation on a small set of data from a source like ODIN (Lewis, 2006) can get a lot of mileage with a small amount of investment.

Perhaps the more challenging issue is determining whether divergence in a language pair is caused by fundamental differences between the languages, or simply stylistic choices in translation. The latter of these scenarios appeared to be common in portions of the SMULTRON data, where translations appeared to be geared toward naturalness in the target language; in contrast, the translations in the Hindi guideline sentences were intended to be as literal as possible. Again, IGT provides a good possible solution, as such examples are often intended specifically for illustrative purposes.

6 Conclusion and Future Work

As the need for NLP tools to operate on resource-poor languages continues to increase, so does the need for electronic resources in these languages. In order to keep pace, semi-supervised methods to tune performance of NLP tools, such as those illustrated here, are ideal as they can scale a wide variety of languages with minimal human supervision.

We have demonstrated a generalizable approach to detecting patterns of structural divergence across language pairs using simple tree operations. While we cannot capture all cases, many of the regular patterns are captured. Our method shows first that there is significant room for improvement in current basic projection algorithms, and that there is promise in systematizing ways to find and deal with structural dissimilarities across languages.

In future work, we plan to expand the languages covered to include Chinese↔English, and possibly Czech↔English, taking advantage of the English–Chinese Translation Treebank (Bies et al., 2007) and the Prague Czech–English Dependency Treebank (Čmejrek et al., 2004). We also plan to examine broader language coverage by using the ODIN database (Lewis and Xia, 2010), which has IGT for a large number of languages. With such broad, multilingual coverage, we believe it is possible to examine the distribution of common divergence patterns, and design a better projection algorithm with automatic handling of common divergence patterns.

References

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia, 2009. A multi-representational and multi-layered treebank for Hindi/Urdu. In ***The Third Linguistic Annotation Workshop (The LAW III) in conjunction with ACL/IJCNLP 2009***. Association for Computational Linguistics.

Ann Bies, Martha Palmer, Justin Mott, and Colin Warner, 2007. ***English Chinese Translation Treebank v1.0***. Linguistic Data Consortium, Philadelphia. URL `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2007T02`.

Martin Čmejrek, Jan Cuřín, Jiří Havelka, Jan Hajič, and Vladislav Kuboň, 2004. Prague czech-english dependecy treebank: Syntactically annotated resources for machine translation. In ***4th International Conference on Language Resources and Evaluation***.

Michael Collins, 1999. ***Head-driven statistical models for natural language parsing***. Ph.D. thesis, University of Pennsylvania.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D Manning, 2006. Generating typed dependency parses from phrase structure parses. In ***Proceedings of LREC 2006***.

Bonnie Jean Dorr, 1994. Machine translation divergences: a formal description and proposed solution. ***Computational Linguistics***, 20:597–633.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak, 2004. Bootstrapping parsers via syntactic projection across parallel texts. ***Natural Language Engineering***, 1(1):1–15.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak, 2002. Evaluating translational correspondence using annotation projection. In ***Proceedings of ACL 2002***.

William D Lewis, 2006. ODIN: A model for adapting and enriching legacy infrastructure. In ***e-Science '06***, page 137.

William D Lewis and Fei Xia, 2008. Automatically identifying computationally relevant typological features. In ***Proceedings of IJCNLP 2008***.

William D Lewis and Fei Xia, 2010. Developing ODIN: A multilingual repository of annotated language data for hundreds of the world's languages. ***Journal of Literary and Linguistic Computing (LLC)***, 25(3):303–319.

Martin Volk, Anne Göhring, Torsten Marek, and Yvonne Samuelsson, 2010. SMULTRON (version 3.0) — The Stockholm MULtilingual parallel TReebank. URL `http://www.cl.uzh.ch/research/paralleltreebanks/smultron_en.html`, an English-French-German-Spanish-Swedish parallel treebank with sub-sentential alignments.

Fei Xia and William D Lewis, 2007. Multilingual structural projection across interlinear text. In ***Human Language Technologies: NAACL 2007***.

David Yarowsky and Grace Ngai, 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In ***Proceedings of NAACL***. Johns Hopkins University, Stroudsburg, PA.