# Transferring Syntactic Relations from English to Hindi Using Alignments on Local Word Groups

**Aswarth Dara, Prashanth Mannem, Hemanth Sagar Bayyarapu and Avinesh PVS**
Language Technologies Research Center
International Institute of Information Technology
Hyderabad, AP, India - 500032
{abhilash.d,prashanth,hemanth.sagar,avinesh}@research.iiit.ac.in

## Abstract

Various works have used word alignments in parallel corpora to transfer information like POS tags, syntactic trees and word senses from source to target sentences. In this paper, we work on the problem of projecting syntactic relations from English to morphologically rich Hindi parallel text. We show the effectiveness of Local Word Groups (LWGs) in simplifying alignments as well as in transferring syntactic dependencies by building an alignment model with LWGs as base units and training a dependency parser on the relations projected using these LWGs. The LWG alignment model using GIZA++ scores decreases the Alignment Error Rate by 1.16 points when compared to the best GIZA++ model trained on lemmas. We also show that a dependency parser trained on the syntactic relations projected using LWGs obtained statistical significant improvements over the relations projected using lemmas by a margin of 3.49%.

## 1 Introduction

Data driven dependency parsers rely on the availability of large amounts of annotated data and building them is a time consuming, labour intensive and expensive task. Recent efforts in treebank creation have looked at the concept of *annotation projection* from a resource-rich language to a resource-poor one in a parallel corpus (Yarowsky et al., 2001). The central idea is to transfer the relevant information from source to target text given the analysis on the source side and word-alignments in a parallel corpora. A projected dependency treebank is created by projecting source parse from an automatic parser on to a target sentence using word alignments between the source and target sentences (Hwa et al., 2005; Jiang and Liu, 2009; Spreyer and Kuhn, 2009; Spreyer et al., 2010).

Difficulties in transferring syntactic relations from the source to target sentences arise mainly due to (i) errors in the source parse, (ii) errors in word alignments and (iii) differences in the dependency annotation schemes of the two languages (Ganchev et al., 2009). The third one can be handled by systematically identifying the differences in the annotation schemes and applying relevant transformations (Hwa et al., 2005).

In this work, we minimize the scope of errors in (i) and (ii) and the effect they have on projection task by proposing a novel technique for projecting syntactic dependencies using alignments between local word groups instead of word forms. The aim is to make the projection task simpler by transferring the relations in source LWGs to the corresponding target LWGs first and then get the alignments and projections on the head words of these LWGs. We show how LWGs can effectively be used to handle the inflectional variations in Hindi and thereby improve the word alignment accuracy between English and Hindi. We also show that the projection of syntactic relations becomes easier and more effective when dealing with LWGs rather than word-forms. We present the experiments and report the improvements in results obtained by using LWGs in alignment accuracy and parsing accuracy for English-Hindi language pair. The approach described in the paper is not specific to the English-Hindi language pair and can be extended to other language pairs involving at least one morphologically rich language.

The rest of the paper is organized as follows: Section 2 gives the related work and section 3 introduces the concept of LWGs. Section 4 presents the alignment models used in this work and section 5 describes the dependency projection algorithm. Section 6 lists the experiments conducted and reports the results. Section 7 ends the paper with the concluding remarks and gives the scope of future work in this direction.

## 2 Related Work

Earlier works in projecting dependencies using word alignments have mainly concentrated on aligning word forms, projecting dependencies based on the aligned words and then improving the projections using post-projection transformations.

(Hwa et al., 2005) project the syntactic dependencies from one language to another using the notion of *direct correspondence*. If the syntactic tree is not fully connected, they used post-projection transformations to make it complete and reported significant improvements due to it. (Ganchev et al., 2009) used open source posterior alignment toolkit PostCAT (Graca et al., 2009) for word alignments to project the syntactic dependencies in a way similar to (Hwa et al., 2005). They use some soft-constriants for word alignments to prevent false transfer of information and also experiment with the projections by varying post-projection transformation rules.

For dependency projection, (Spreyer and Kuhn, 2009) also start with the notion of *direct correspondence* with additional constraints involving (i) only the bi-directional alignment links (alignment links marked from both source to target language and target to source language) and (ii) the completeness of the projected tree. This ensures the reduction of errors in the data. It is further extended to consider the uni-directional alignments based on the partial analyses built from the confident bi-directional links. They report no significant increase in accuracy with the above extension because of the increase in noise. To reduce this drawback and make use of non-fully connected trees, they stick to the use of bi-directional alignment links in conjunction with a fixed number of fragmented analyses in a sentence.

(Jiang and Liu, 2009) refer to an alignment matrix and a dynamic programming algorithm to search for a completed projected tree. To reduce the impact of word alignment errors, (Jiang and Liu, 2010) consider compact representation of multiple GIZA++ (Och and Ney, 2000) alignment results. For each pair of words in a target sentence, they calculate the score of an edge depending on the alignment scores and source parses information. During parsing, they treat each syntactic tree as a set of dependency edges without relying on the complete tree.

The above approaches either pick most confident alignments links for dependency projection or pick the alignments as it is and then reduce the noise in the data using post-projection transformation rules. In this work, we focus on reducing both alignment and dependency projection complexity using LWGs. The relations within LWGs are deterministically marked irrespective of the alignments and by using only the head word of each LWG during alignment we alleviate data sparseness arising due to the the presence of inflectional variations in Hindi.

## 3 Local Word Groups

We define *local word groups* as minimal contiguous sequence of words in a sentence with a fixed word order and deterministic (trivially predictable) syntactic structure among them. For example, `will be given` is a verb group with the syntactic structure `will`→`be`→`given` which can be determined deterministically given the POS tags. Each LWG has a head word which is the syntactic head of the group of words (`will` in the example). LWG is similar to chunk except that LWGs are always minimal and refer to the fixed order property of the components. The advantage of using LWGs over chunks is the accuracy of identifying LWGs is higher than the accuracy of identifying chunk boundaries in source and target language. Also, the dependency relations within a chunk are non-deterministic in few cases (if we consider noun chunks). Moreover they can be computed with minimum computational effort.

The advantages LWGs provide during word alignment and transfer of syntactic information from source language to target language are:

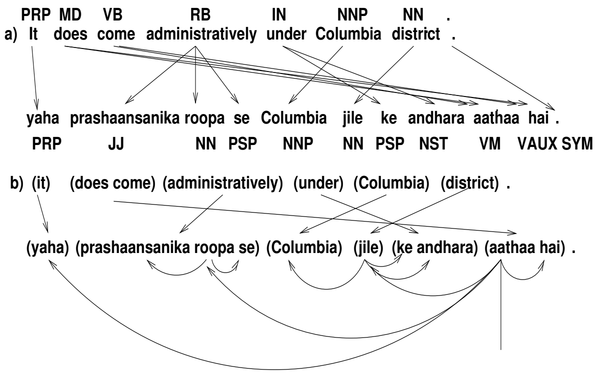
1. It transform certain kinds of many-to-one, one-to-many and many-to-many alignments between word forms as one-to-one alignments between the LWGs thereby reducing data sparsity during alignment.

2. Since the internal structure of a LWG is fixed, we can confidently mark the syntactic relations within a LWG leaving out scope for errors arising out of word alignment and source dependency parses.

3. Number of syntactic relations to be projected reduced from number of words in a sentence

to number of LWGs thereby reducing the scope of errors in projection.

In Figure 1(a), *administratively* is aligned to *prashaansanika roopa se*. Once these alignments are obtained, in order to get the dependency relations `roopa`→`prashaansanika` and `roopa`→`se` correct from the projection of dependencies, the source dependency parse should be correct and the word alignment to all the three target words should be correct. The word alignment complexity further increases in the case of aligning *does come* to *aathaa hai* correctly considering the presence of inflectional variations in Hindi. Unless all the four alignments are correct, the dependency projection will not be correct. The complexity is further increased due to the non-availability of large bilingual corpus. In such cases, LWGs play an important role in reducing the complexity as well as in reducing the scope of errors to certain extent. We consider three kinds



PRP MD VB RB IN NNP NN .
a) It does come administratively under Columbia district .
yaha prashaansanika roopa se Columbia jile ke andhara aathaa hai .
PRP JJ NN PSP NNP NN PSP NST VM VAUX SYM
b) (it) (does come) (administratively) (under) (Columbia) (district) .
(yaha) (prashaansanika roopa se) (Columbia) (jile) (ke andhara) (aathaa hai) .

Figure 1: Alignments without and with LWGs

of LWGs verb groups (VG), preposition (postposition in case of Hindi) groups (PG) and adverb groups (ADVG). Noun groups are not considered since the internal syntactic structure is non-trivial to predict. For instance, in the example in figure 1(a), finding the syntactic structure for the words in the noun group `Columbia`←`district` is non-deterministic (non-trivial) given the POS tags. VM and VAUX are the two POS tags used to denote main verb and auxiliary verb in Hindi. Any occurrence of the pattern `VM VAUX*` is considered a verb LWG eg. (`kara rahaa hai` [*doing*], `karna chahtaa hu` [*want to do*]). Any continuous sequence of auxiliary verbs and a main verb is taken as a LWG in English for eg. *are going*, *did not go*. Postpositions in Hindi are also grouped into LWG. This is done to make alignments easy between prepositions in English and multi-word postpositions in Hindi such as `ke saatha` [*with*], `ke baahara` [*outside*] etc. Adverbs in English often end up as `roopa se` [*+ly*] morpheme in Hindi. Combining the adjective/noun with `roopa se` into a LWG helps the alignment model. Similarly few other cases where forming LWGs help in making alignments easy were identified and are marked as LWGs.

In Figure 1, (`aathaa hai` [*does come*]) is a verb group, `ke andhara` [*under*], are post-positions and (`prashaansanika roopa se` [*administratively*]) is an adverb group. The syntactic relations `aathaa`→`hai`, `roopa`→`prashaansanika` and `roopa`→`se` are marked directly irrespective of the word alignments.

Table 1 lists the corpus stats of word forms and LWGs. Apart from the advantages of LWGs listed above in this section, the reduction in the sentence length also helps improve the alignment accuracy.

| Metric | English | Hindi |
|---|---|---|
| No. of Words | 224K | 251K |
| Avg. sentence length with words | 19.29 | 21.61 |
| No. of LWGs | 191K | 226K |
| Avg sentence length with LWGs | 16.45 | 19.49 |

Table 1: Statistics of the entire corpus used with respect to words and LWGs.

## 4 Word Alignment

### 4.1 Scoring Function

We use a slight variant of competitive linking algorithm (CL) described in (Melamed, 2000) as scoring function and is discussed below in section 4.2. We also generate mappings between source and target language word classes from a small bilingual corpus containing hand-aligned gold alignments and manually annotated word class information. For each source word class, the mappings are stored by decreasing order of target word class co-occurrences (Wcmap).

### 4.2 LWG model

Instead of computing scores between source and target word forms in a two-dimensional array as in (Melamed, 2000), we compute scores between source LWGs and target LWGs. For each source LWG, the candidate LWGs are restricted to the ones whose content word's POS tag are in the mapping list of source LWG content word's POS. The alignment pair with the maximum score is chosen, the corresponding row and column are deleted for further processing. The process is repeated until it finishes. The differences here when compared to competitive linking algorithm (Melamed, 2000) are that we use LWGs instead of words and restricting the possible candidate LWGs by taking word class information into account. We refer to this model as CL-LWG in the rest of the paper.

Once the one-to-one alignments between source and target LWGs are computed, the word to word alignments have to be recovered from them in a postprocessing step. This is done by assigning alignments between all the words in a target LWG to all the words in the source LWG and vice versa. This postprocessing step makes the final set contain contain one-to-many, many-to-one and many-to-many alignments.

### 4.3 Case of Left Out *Units*

In the model described above, there is a possibility of source or target *units* being left out without any alignments. This is because the model only does one-to-one mapping and the number of *units* in the source and target sentence could be different. The left out *units* need not be a case of NULL alignments (such as determiners). The alignments for the left out *units* are assigned by using a greedy best first alignment strategy.

### 4.4 Weighted Model

The CL-LWG alignment model can be further extended to use alignment scores from GIZA++ model apart from the scoring function described in Section 4.1. A weighted score of the score from the scoring function and GIZA++ can be used to decode the model. Let s1 be the score of the scoring function described above and s2 be the score obtained from GIZA++ then the weighted score S is defined as

$$S = \lambda * s1 + (1 - \lambda) * s2; \qquad (1)$$

whereas $\lambda$ is varied from 0 to 1 and the best value of $\lambda$ is chosen by tuning it on the development dataset.

## 5 Dependency Projection

Once the alignments are obtained, projection of dependencies from source to target sentences is undertaken. The input is a set of alignments between source and target language and the source dependency parse with the projected dependency parse of the target sentence as the output. Parses for English have been obtained using first order MST parser (McDonald et al., 2005) trained on 2-21 sections of Penn Treebank (Marcus et al., 1993).

### 5.1 Annotation Scheme Differences

When projecting relations from source to target text, the annotation scheme of source treebank is carried over to the target side. This projected target dependency treebank may not be consistent with the annotation scheme of the target language. The annotation scheme differences need to be handled to make the projected parses consistent for dependency parse evaluation. This also ensures that the projected treebank could be used for bootstrapping parsers (Steedman et al., 2003; Reichart and Rappoport, 2007).

Some of the major differences in the English and Hindi annotation schemes are

1. Head of a Verb Phrase (VP) is the main verb in Hindi whereas the head of a VP in an English sentence is the auxiliary verb.

2. Head of a Prepositional Phrase (PP) in Hindi is noun whereas it is the preposition in English.

3. In Hindi, conjunct is the head in case of both noun and verb co-ordination whereas in English it is not.

There are two possible ways of addressing this issue. The first one is to apply tranformations before dependency projection (pre-projection) i.e., to the source parse and the second one involves applying tranformations to the target parse once the dependency projection is made (post-projection). This assures that the projected treebank is consistent with the annotation scheme of the target language. It does not make any difference while choosing whether to apply pre-projection or post-projection. In our case, we choose to apply pre-projection i.e., to the source dependency parse just for our convenience. Appropriate pre-projection transformations shown in Figure 2 have been applied to the source parse to reflect the target language annotation scheme described in (Begum et al., 2008).

Figure 2(a) presents the tranformations applied in case of finite verbs (*going*) i.e., switch the dependencies between the current parent (*has*) and finite verb (*going*) then make the rest of them as dependents to the finite verb. In case of PP chunk, switch the dependencies of noun (*complaint*) and preposition (*on*) i.e., make parent of noun as parent of preposition and make the preposition (*on*) as dependent to the noun (*complaint*) as shown in Figure 2(b). In case of noun co-ordination shown in Figure 2(c), switch the dependencies between the rightmost conjunct (*officials*) and co-ordination word (*and*). Then the remaining words (*investors*, *managers* and ,) are made as dependents to the co-ordinated word (*and*). Same co-ordination transformations shown in Figure 2(d) are applied in case of verb co-ordination except that the switching of dependencies take place between the left most finite verb (*makes*) and co-ordination word (*and*) whereas it is rightmost conjunct in case of noun co-ordination (*officials*).

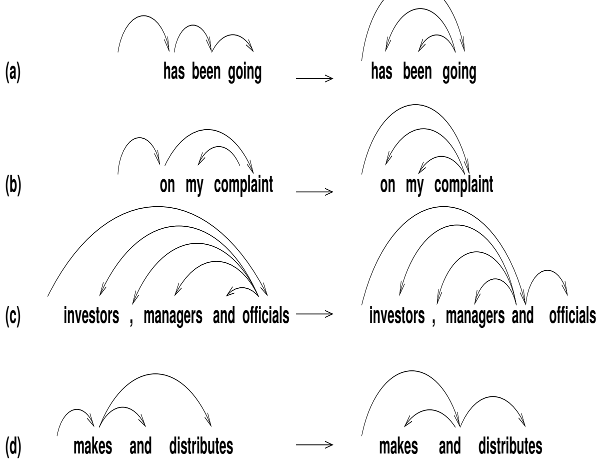


Figure 2: Transformations applied to the source parse.

### 5.2 Projection Algorithm

Let S($s_1$,$s_2$,..,$s_m$), T($t_1$,$t_2$,..,$t_n$) be the source and target language sentences of length m and n. For each dependency relation ($s_i$, $s_j$) where $s_i$ is the head of $s_j$, the projection is done depending on the type of alignment $s_i$ and $s_j$ have with the words in the target sentence. The one-many, many-one and many-many alignments are transformed into one-one alignments by making the syntactic head word of the "many" words in the alignment as "one" representative word and the rest as the dependents of the head word.

1. If $s_i$ is aligned to $t_k$ (*one-one* alignment) and $s_j$ is aligned to $t_l$ (*one-one* alignment) then mark the dependency relation ($t_k$, $t_l$).

2. If $s_i$ is aligned to $t_a$,...,t$k$ (*one-many* alignment)and $s_j$ is aligned to $t_l$ (*one-one* alignment) then pick the head word $t_h$ from $t_a$,...,t$k$ target words and make the remaining as dependents to the head word and mark the dependency relation as ($t_h$, $t_l$).

3. if $s_a$,...,s$i$ are aligned to $t_k$ (*many-one* alignment) and $s_j$ is aligned to $t_l$ then pick the head word $s_h$ from $s_a$,...,s$i$ source words and repeat step 1.

4. The *many-many* alignments are also converted to *one-one* by first performing the *one-many* transformation in step 2 and then the transformation for *many-one* in step 3.

If there are any unaligned words in the target sentence then they are left as unconnected words in the projected target dependency tree. The head word from a list of words in the above transformations is obtained using word class information.

## 6 Experiments

Our experiments have been carried out on English-Hindi parallel text from EMILLE corpus. EMILLE is a 63 million word electronic corpus of South Asian languages, especially those spoken as minority languages in UK (Xiao et al., 2004). The dataset has 3341 training sentences and 90 manually aligned test sentences (with 1121 English words and 1323 Hindi words). It is a small corpus compared to corpora available for other language pairs.

We evaluate the word alignments as well as the projected target relations to measure the effectiveness of LWGs in alignment as well as in projection process. For evaluating the quality of projected target relations, a parser is trained on the dependency edges obtained using projections on alignments between LWGs and those obtained using projections on just word/lemma alignments.

### 6.1 Word Alignment Evaluation

The alignments are compared with those obtained by GIZA++ (Och and Ney, 2000) and PostCat (Graca et al., 2009). We train two GIZA++ models for comparison with our model. The first model is trained with the word forms from the above dataset and second one with the lemmas (roots) of the words. For each of these models, the best among the grow-diag, grow-diag-final and grow-diag-final-and (gdfa) modes have been taken for comparison. Two models are trained using Post-CAT, same as with GIZA++, one with the word forms and the other with the lemmas of the words. For each of these models, the best among the baseline, agreement and substochastic modes have been taken for comparison. The alignment models are evaluated using the standard alignment evaluation metrics precision, recall, F score and Alignment Error Rate (AER) as described in (Hua et al., 2005).

| Id | Models | P | R | AER |
|---|---|---|---|---|
| *WF-PC* | PostCAT | 59.49 | 46.73 | 47.65 |
| *WF-GZ* | GIZA++ | 47.30 | **52.55** | 50.2 |
| *WF-LWG* | CL-LWG | **60.94** | 47.1 | **46.88** |

Table 2: Evaluation Results with Word Forms as *Units*. WF denotes word forms, PC is PostCAT, GZ is GIZA++ and LWG is our model.

Table 2 presents the Precision (P), Recall (R) and AER (1-F) for all the models. All the models in the table use the word forms given in the parallel sentences and do not consider the lemmas. Our LWG model (*WF-LWG*) performs better than the GIZA++ model (*WF-GZ*) and PostCAT model (*WF-PC*) by a margin of 3.32 and 1.23 points respectively . This shows that the LWGs are more effective than word forms when it comes to English-Hindi parallel data.

Since Hindi is a morphologically rich language and the bilingual corpus we use is small, data sparsity becomes a major problem for word alignment. This can be alleviated to some extent by learning from lemmas instead of the word forms themselves.

| Id | Models | P | R | AER |
|---|---|---|---|---|
| *LM-PC* | PostCAT | 51.07 | 42.25 | 53.75 |
| *LM-GZ* | GIZA++ | **62.14** | **49.78** | **44.72** |
| *LM-LWG* | CL-LWG | 60.15 | 48.58 | 46.24 |

Table 3: Evaluation Results with Root Words as *Units*. LM denotes lemmas (roots) of the words.

Table 3 lists the performance of the PostCAT, GIZA++ and LWG model when trained with lemmas. In case of LWGs, the lemmas of the head words are used. GIZA++ trained on lemmas (*LM-GZ*) has a significant reduction in AER when compared to the one trained on word forms (*WF-GZ*). This is expected due to the above mentioned reason of Hindi being morphologically rich and lemmas reducing data sparseness. The lemmatized LWG model (*LM-LWG*) under performs when compared to GIZA++ trained on lemmas (*LM-GZ*) even though in the case of training on word forms, *WF-LWG* out performed *WF-GZ*. This is because LWGs with word forms reduce the problem of data sparseness when compared to just word forms being used in GIZA++ model *WF-GZ* but the gains in lemmatized model is not that significant to beat the gains achieved in lemmatized GIZA++ model. Moreover, the scoring function of GIZA++ is more sophisticated than the one we use.

To get the combined effects of using LWGs and GIZA++'s scoring function, two models using weighted scores of scores from LWG model and GIZA++ model for word forms (*WF-LWG-WGT*) and lemmas (*LM-LWG-WGT*) were built.

| Id | Models | P | R | AER |
|---|---|---|---|---|
| *WF-LWG-WGT* | WM | 61.86 | 47.94 | 45.98 |
| *LM-LWG-WGT* | WM | **62.85** | 51.21 | **43.56** |

Table 4: Evaluation Results for Weighted Models (WMs)

Table 4 lists the evaluation scores for the weighted models. The combined models beat the respective best word form and lemma models (i.e., *WF-LWG* and *LM-GZ*) by 0.9 and 1.16 respectively. The best alignment accuracy was achieved by the combined model trained on lemmas.

Experiments were conducted by combining words in each LWG using underscores and also by using only head word of each LWG as input to GIZA++. Using a post-processing step, the word alignments were recovered from the resulting alignments. The alignment accuracies decreased significantly using this approach.

### 6.2 Evaluation of projected parses

Projections from English dependency trees using alignments produced by various models described above have been obtained and trained separately. We used a modified version of bidirectional parser (Shen and Joshi, 2008) described in (Mannem and Dara, 2011) to train the projected dependencies. The parsers evaluation was done on manually an-

| Alignment | Word forms (WF) | | | Lemmas (LM) | | |
|---|---|---|---|---|---|---|
| | *EMILLE* | *EILMT* | *BOTH* | *EMILLE* | *EILMT* | *BOTH* |
| PC | 72.76 | 72.31 | 74.26 | 72.15 | 72.23 | 72.48 |
| GZ | 74.93 | 77.71 | 77.17 | 75.87 | 77.33 | 77.08 |
| LWG-WGT | **78.06** | **79.50** | **79.79** | **78.76** | **80.52** | **80.57** |

Table 5: Parsing accuracy corresponding to the parsers trained on projections from alignment models when evaluated with gold POS tags. The six respective parsers are referred by prefixing WF- and LM- to the three alignment models (PC, GZ and LWG-WGT).

| Alignment | Word forms (WF) | | | Lemmas (LM) | | |
|---|---|---|---|---|---|---|
| | *EMILLE* | *EILMT* | *BOTH* | *EMILLE* | *EILMT* | *BOTH* |
| PC | 61.06 | 61.60 | 62.50 | 61.98 | 61.90 | 61.51 |
| GZ | 63.40 | 65.85 | 65.71 | 64.51 | 65.50 | 65.50 |
| LWG-WGT | **66.86** | **67.88** | **68.37** | **67.50** | **68.22** | **68.75** |

Table 6: Parsing accuracy corresponding to the parsers trained on projections from alignment models when evaluated with automatic POS tags. The six respective parsers are referred by prefixing WF- and LM- to the three alignment models (PC, GZ and LWG-WGT).

notated test data with gold POS tags released as part of ICON-2010 NLP Tools Contest (ICON, 2010). We also presented the results on the manually annotated test data with the automatic POS tags.

We use an additional English-Hindi parallel corpus containing 8169 sentences developed as part of a consortium project (Venkatapathy, 2008) [1]. This dataset wasn't used for evaluating word alignments due to the lack of manually annotated alignments. This sentence aligned corpus was developed to aid building translation systems for the tourism domain and doesn't have any human annotated word alignments. The corpus is a collection of articles about various tourist and pilgrimage places. It has a high occurrence of proper nouns as a result of this. The corpus is also noisy with typographical errors, mismatched sentences and unfaithful translations.

The parser is trained on the projected data extracted using alignments from all the three different alignment models using word forms and LWGs. Table 5 gives the Unlabeled Attached Score (UAS) of the parser on the test data. The parsers corresponding to *WF-LWG-WGT* and *LM-LWG-WGT* alignments obtained siginificant improvements of 2.62 points and 3.49 points in parsing accuracy over the ones using GIZA++ alignments on words and lemmas when both the datasets are used for training. The results obtained are statistically significant (McNemar's and $p < 0.05$). The parser trained on the projected data obtained using *LM-LWG-WGT* alignments achieved an improvement of over 0.78 points over the one trained using *WF-LWG-WGT* alignments and it is expected.

Table 6 gives the parsing accuracies on test data with automatic POS tags. The parsing accuracy corresponding to *LM-LWG-WGT* alignment model has dropped from 80.57 to 68.75 due to the difference in usage of POS tags. For all the datasets, the parser trained on the projected data obtained using word alignments from the weighted models (*WF-LWG-WGT*, *LM-LWG-WGT*) performed the best and the results are statistically significant (McNemar's and $p < 0.05$) over the models using alignments from GIZA++ and PostCAT in terms of both word forms and lemmas. (Ambati et al., 2010) achieved an accuracy of 85.5% UAS by training on Hyderabad Dependency Treebank (Bhatt et al., 2009) and evaluating with automatic POS tags. This would be an upperbound for approaches like ours which do not use any annotated data to build the parser.

The Hindi dependency treebank has chunks marked for every sentence and the dependencies are marked between chunk head words. Following the tradition in Hindi dependency parsing, we

[1]The original training and test sets in this corpus contained 11,300 and 500 sentences respectively. But, both the datasets had a large number of sentence repetitions. The sizes reported in this paper are after removing all duplicate sentences. http://ltrc.iiit.ac.in/icon2008/nlptools.php

evaluate and compare the two parsing models on their inter-chunk and intra-chunk dependencies. A total of 6454 dependency relations are present in the test data out of which 3401 are intra-chunk dependencies and the rest 3053 are inter-chunk dependencies. The models which achieved best accuracy (*LM-LWG-WGT*, *WF-GZ* and *WF-PC*) are taken into comparison when both the datasets are used for training.

Table 7 gives the accuracies w.r.t few POS tags achieved by various parsing models tested with gold POS tags in the test data. These POS tags are the most frequent tags of chunk head words and the accuracies denote the inter-chunk accuracies of the models. The analysis on intra-chunk dependencies hasn't been shown since there isn't much difference in the accuracies between the two parsing models. This is because, though *WF-PC* and *WF-GZ* don't use any LWGs, the intra-chunk relations are corrected and are made similar to those produced using *LM-LWG-WGT* by using transformation rules during projection as described in section 3. As seen in Table 7 for inter-chunk dependencies, *LM-LWG-WGT* alignment based parsing model performs better than the model using *WF-GZ* and *WF-PC* alignments for most POS tags (except CC tag) when compared to *WF-GZ*. This is because of better alignments achieved by *LM-LWG-WGT* using LWGs over *WF-GZ*, *WF-PC* and thereby better dependency projections to learn from.

| **POS** | **Total** | **WF-PC** | **WF-GZ** | **LM-LWG-WGT** |
|---|---|---|---|---|
| NN | 1257 | 57.28 | 58.15 | **66.35** |
| VM | 710 | 33.10 | 44.65 | **55.91** |
| NNP | 444 | 38.29 | 45.27 | **50.00** |
| CC | 213 | 39.90 | **52.58** | 46.95 |
| PRP | 209 | 35.41 | 47.85 | **51.67** |
| JJ | 129 | 72.10 | 74.42 | **88.37** |
| RB | 33 | 24.24 | 45.45 | **51.51** |

Table 7: Parsing Accuracies on inter-chunk dependencies when both the datasets are combined. Total represent the total number of inter-chunk dependencies with each POS tag occurred in the test data. Third, fourth and fifth columns represent the percentage of correct inter-chunk dependencies for each POS tag obtained using the respective alignment models.

## 7 Conclusions and Future Work

In this work, we presented a novel approach for syntactic transfer of relations from source to target sentences by using alignments between LWGs instead of word forms. We also showed the effectiveness of LWGs while handling one-to-many, many-to-one and many-to-many alignments during both word alignment and dependency projection. The LWG weighted alignment model decreased the AER by 1.16 points over GIZA++ trained on lemmas and a bidirectional dependency parser trained on the syntactic relations projected using LWGs outperforms the relations projected using lemmas by a statistically significant margin of 3.49 points. We also presented evaluation of parsers in terms of inter chunk and intra chunk dependencies. Extending this work to other resource-poor Indian language pairs will be the starting point of our future work.

## References

Bharat Ram Ambati, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010. Two methods to incorporate local morphosyntactic features in hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, SPMRL '10, pages 22–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Begum, S. Husain, A. Dhwaj, D. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for indian languages. In *In Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNLP)*, Hyderabad, India.

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 *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 186–189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *ACL '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 369–377, Morristown, NJ, USA. Association for Computational Linguistics.

J Graca, K Ganchev, and B Taskar. 2009. Postcat - posterior constrained alignment toolkit. In *In The Third Machine Translation Marathon.*

Wu Hua, Wang Haifeng, and Liu Zhanyi. 2005. Alignment model adaptation for domain-specific word alignment. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 467–474, Morristown, NJ, USA. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11:311–325, September.

ICON. 2010. Nlp tools contest 2010.

Wenbin Jiang and Qun Liu. 2009. Automatic adaptation of annotation standards for dependency parsing: using projected treebank as source corpus. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, pages 25–28, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wenbin Jiang and Qun Liu. 2010. Dependency parsing and projection based on word-pair classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 12–20, Morristown, NJ, USA. Association for Computational Linguistics.

Prashanth Mannem and Aswarth Dara. 2011. Partial parsing from bitext projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1597–1606, Portland, Oregon, USA, June. Association for Computational Linguistics.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. 19(2):313–330.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

I. Dan Melamed. 2000. Models of translational equivalence among words. *Comput. Linguist.*, 26:221–249, June.

Franz Josef Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *COLING '00: Proceedings of the 18th conference on Computational linguistics*, pages 1086–1090, Morristown, NJ, USA. Association for Computational Linguistics.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL'07*, pages 616–623.

Libin Shen and Aravind Joshi. 2008. LTAG dependency parsing with bidirectional incremental construction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 495–504, Honolulu, Hawaii, October. Association for Computational Linguistics.

Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 12–20, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kathrin Spreyer, Lilja Øvrelid, and Jonas Kuhn. 2010. Training parsers on partial trees: A cross-language comparison. In ELRA, editor, *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10).*

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 331–338, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sriram Venkatapathy. 2008. Nlp tools contest 2008: Summary.

Zhonghua Xiao, Tony Mcenery, Paul Baker, and Andrew Hardie. 2004. Developing asian language corpora: standards and practice. In *In Proceedings of the Fourth Workshop on Asian Language Resources.*

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, HLT '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.