Improving Dependency Parsing with Interlinear Glossed Text and Syntactic Projection

Ryan Georgi[1] Fei Xia[1] William D Lewis[2]
(1) University of Washington, Seattle, WA 98195, USA
(2) Microsoft Research, Redmond, WA 98052, USA
{rgeorgi, fxia}@uw.edu, wilewis@microsoft.com

ABSTRACT
Producing annotated corpora for resource-poor languages can be prohibitively expensive, while obtaining parallel, unannotated corpora may be more easily achieved. We propose a method of augmenting a discriminative dependency parser using syntactic projection information. This modification will allow the parser to take advantage of unannotated parallel corpora where high-quality automatic annotation tools exist for one of the languages. We use corpora of interlinear glossed text—short bitexts commonly found in linguistic papers on resource-poor languages with an additional gloss line that supports word alignment—and demonstrate this technique on eight different languages, including resource-poor languages such as Welsh, Yaqui, and Hausa. We find that incorporating syntactic projection information in a discriminative parser generally outperforms deterministic syntactic projection. While this paper uses small IGT corpora for word alignment, our method can be adapted to larger parallel corpora by using statistical word alignment instead.

KEYWORDS: Dependency Parsing, Syntactic Projection, Interlinear Glossed Text.

1 Introduction

The development of large-scale treebanks has significantly improved the performance of statistical parsers (e.g. Klein and Manning, 2001; Collins and Koo, 2005; de Marneffe et al., 2006) Unfortunately, resources such as these typically only exist for a handful of languages, because building large treebanks is very labor-intensive and often cost-prohibitive. As thousands of other languages have no such resources, other techniques are required to attain similar performance.

One way in which natural language tools might be created for resource-poor languages is by using resources containing translations between resource-rich and resource-poor languages and use the alignment information to transfer information to the resource-poor ones. Syntactic projection takes advantage of existing tools used to annotate a resource-rich language in a corpus, and transfers the analysis to the resource-poor language by means of syntactic projection using word-to-word alignment (Yarowsky and Ngai, 2001; Hwa et al., 2002).

While little annotated data typically exists for resource-poor languages, some work has been done examining the utility of interlinear glossed text (IGT) (Xia and Lewis, 2007; Lewis and Xia, 2008), a format used for illustrative examples in linguistic papers. An IGT instance includes a language line which is a phrase or sentence in a foreign language, a gloss line that shows word-to-word translation, and a translation line which is normally in English. We chose IGT for this study because the gloss line can serve as a bridge for aligning the words in the language line and the translation line. The method proposed in this paper can be easily extended when IGT is replaced by bitexts of sufficient size to train a high-quality word aligner.

In this paper, we investigate the possibility of using small corpora of interlinear text and syntactic projection to bootstrap a dependency parser and improve the resulting parses over projection alone.

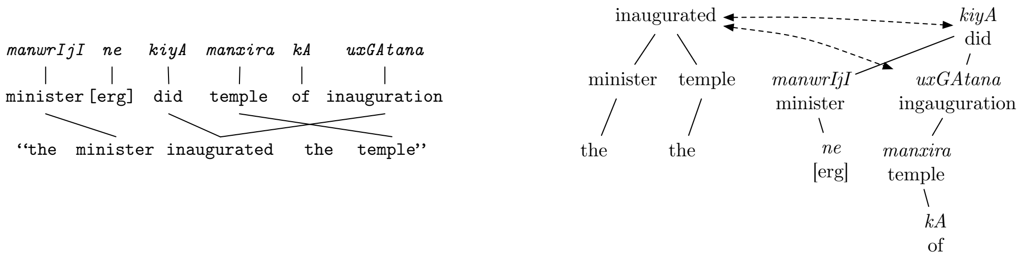
2 Background

Before presenting our system, we will first describe previous studies on syntactic projection. Next, we will describe the phenomenon of linguistic divergence, and explain why this can affect projection performance. Finally, we will describe interlinear glossed text (IGT) in more detail and highlight the ways in which it is well-suited to this task.

2.1 Syntactic Projection

Syntactic projection via word alignment has shown promise in adapting resources between languages. Merlo et al. (2002) demonstrated a technique of classifying verb types via projection, while Yarowsky and Ngai (2001) worked on projecting POS taggers and NP bracketers. Hwa et al. (2004) bootstrapped both phrase and dependency parsers. While these systems did not match the performance of supervised systems, they do succeed in demonstrating that even a small amount of information can significantly boost the performance of a baseline system.

Syntactic projection, however, suffers from a major flaw—using word alignment to transfer analyses between languages assumes that the language pair represents the similar sentences using similar structures. Hwa et al. (2002) referred to this as the Direct Correspondence Assumption, or DCA. As useful as this assumption may be, Dorr (1994) discussed how languages may be divergent in their representations of similar sentences in semantic, syntactic, or lexical representations.


manwrIjI ne kiyA manxira kA uxGAtana
minister [erg] did temple of inauguration
"the minister inaugurated the temple"
inaugurated
kiyA
did
minister
temple
manwrIjI
minister
uxGAtana
ingauguration
the
the
ne
[erg]
manxira
temple
kA
of

(a) Original IGT representation and word alignment between the three lines.

(b) The dependency trees for Hindi and English

Figure 1: An example of the light verb construction in Hindi.

2.2 Linguistic Divergence

Dorr (1994) defined several types of translation divergence. These variations, or linguistic divergence, can be problematic for projection algorithms, as the assumptions that projection relies on may not hold. One example which may affect projections is the light-verb construction. Take as an example the Hindi sentence in Figure 1, where an English verb ("inaugurated") is represented in Hindi as a light verb ("did") plus a noun ("inaugurate"). As a result, the dependency structures in English and Hindi are not identical, as illustrated in Figure 1(b).

In order to address some of the noise inherent in projection results, Ganchev et al. (2009) took an approach of using "soft" constraints to improve projection results. Rather than commit to a single parse, Ganchev et al. used statistical methods to disambiguate between multiple parse options, and showed significant improvements over a purely deterministic approach.

In our previous study, (Georgi et al., 2012), we looked at measuring forms of alignment between dependency structures to quantify the amount of divergence between languages. In this paper we will look at how performance varies among a set of typologically different languages. We hope to investigate the correlation between performance in these languages and these quantitative measures in future work.

2.3 Interlinear Glossed Text (IGT)

IGT is a resource well-suited to the task of adapting dependency parsing to new languages. As seen in Figure 1a, an IGT instance contains a foreign-language line, an English translation, and a gloss line, which provides additional annotation about the foreign language line. Xia and Lewis (2007) demonstrated that interlinear glossed text can be used to obtain high quality word-to-word alignments with a relatively small amount of data. Xia and Lewis further show

	Hindi	German	Irish	Hausa	Korean	Malagasy	Welsh	Yaqui
# IGT Instances	147	105	46	77	103	87	53	68
# of Words (F)	963	747	252	424	518	489	312	350
# of Words (E)	945	774	278	520	731	646	329	544

Table 1: Data set sizes for all languages. For the number of words, the number of words in the foreign (F) language are given first, followed by the number of English (E) words.

that these heuristics can be used to augment statistical methods to produce higher-quality alignments over statistical methods alone, suggesting that a small corpus of IGT may be able to provide a bootstrap for alignment on much larger parallel corpora.

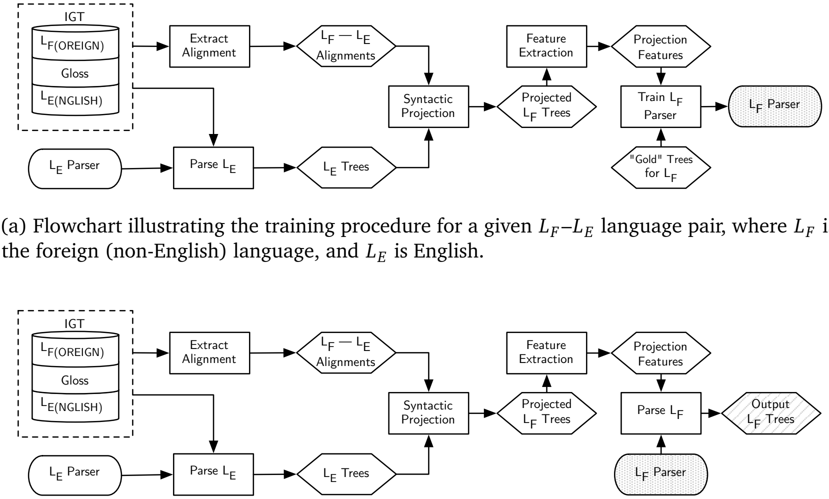
Where does one find IGT corpora? The Online Database of INterlinear text (ODIN) (Lewis and Xia, 2010) is an online resource of IGT instances for that contains approximately two hundred thousand instances for 1274 languages. Lewis and Xia (2008) use ODIN data for 97 languages to perform syntactic projection and determine basic word order. They found that the languages in this sample with 40 or more instances could be used to predict basic word order with 99% accuracy. With such broad language coverage, ODIN is an ideal resource for providing information for resource-poor languages for which little other data exists.

3 Methodology

Given a set of IGT instances, our system works as follows (see Figure 2):

1. Align words in the language line and translation line (the translation lines are all English in our experiments)
2. Parse the translation lines using an English parser
3. Project the dependency structure of the translation line to the language line
4. Extract features from the projected structure and use them to train a parser.

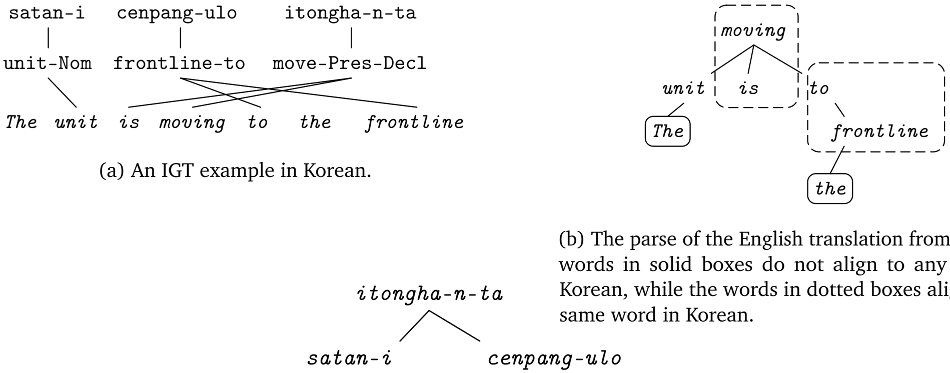
If the input is a set of parallel sentences instead of IGT, the process will be exactly the same except that word alignment in step (1) will be done by training a statistical word aligner, instead of using the gloss line in IGTs as a bridge between the language lines and the translation lines.



(a) Flowchart illustrating the training procedure for a given L_F–L_E language pair, where L_F is the foreign (non-English) language, and L_E is English.

(b) A flowchart showing an overview of the testing procedure for the L_F–L_E language pair using the L_F parser produced in the training phase, and augmented by the information projected from the parsed L_E portion of the bitext. A Future system will include an augmented statistical aligner to produce the L_F–L_E alignments.

Figure 2: Flowcharts illustrating the training and testing phases of the proposed system.



(a) An IGT example in Korean.

(b) The parse of the English translation from (a). The words in solid boxes do not align to any word in Korean, while the words in dotted boxes align to the same word in Korean.

(c) The projected tree for the Korean sentence, after the English words are replaced by Korean counterparts, words in the solid boxes are removed and words in the dotted boxes are collapsed into a single node.

Figure 3: The steps of the projection process, as illustrated using an IGT instance from Korean.

In this study, we will focus the last two steps. Therefore, for the first two steps, we use word alignment and English parse trees from the gold standard. The last two steps and the evaluation corpora are explained below.

3.1 Corpora

We used two different sources of language data for our experiment. The first was a set of guideline sentences for the Hindi treebank (Bhatt et al., 2009). These sentences were provided both in the IGT format, as well as with gold-standard dependency trees in both English and Hindi. The second set was the IGT data used in Xia and Lewis (2007), which includes IGTs for seven languages, plus manually annotated word alignment and dependency trees for both English and the foreign language. In all, eight languages were used for our experiments: Hindi, German, Irish, Hausa, Korean, Malagasy, Welsh, and Yaqui. The size of each data set is given in Table 1. We use the pre-existing, manually annotated parse trees in these two data sets as our gold standard for evaluation.

For the Hindi data, which did not include word alignment, we first automatically aligned the Hindi sentences and the English translation via the gloss line as in Xia and Lewis (2007), then manually corrected the alignment errors. While this manual correction step creates alignments that are not fully automated, the amount of effort required to make these modifications are minimal, and still greatly reduces the costs involved in creating parallel annotated corpora.

3.2 The Projection Algorithm

For the projection algorithm, we follow the dependency projection algorithm described in Xia and Lewis (2007), an example of which can be seen in Figure 3. Given the word alignment L_E and L_F and the parse tree T_E for L_E, the projection algorithm works as follows. For each English node e_i that aligns with foreign word f_i, we replace the node for e_i with the foreign word. If a single English node e_i aligns with multiple foreign words (f_i, f_j), we make multiple copies of e_i as siblings in the tree for each source word, then replace the English words with those from the source. If multiple English nodes align to a single foreign word, the node highest up in the tree is kept, and all others removed. Finally, remaining unaligned words in L_F are attached heuristically, following (Quirk et al., 2005).

3.3 The Parser

Due to linguistic divergence, projected trees are error-prone. Instead of making hard decisions based on projection, we use information from projected trees as a feature in a discriminative parser. This feature will be highly predictive, but not result in a strictly deterministic method like projection alone. We modified the MST Parser (McDonald et al., 2006) by adding features that check whether certain edges considered by the parser appear in the projected tree. We define two types of features: `BOOL`, and `TAG`.

The first feature type, `BOOL`, looks at the current parent→child edge being considered by the parser and returns `true` if the edge matches one in the projected tree. While this feature was a logical starting point, we also wondered if certain word classes of English projected better than others, and so the second feature type, `TAG`, creates a feature for each $\left(POS_{parent}, POS_{child}\right)$ pair such that the feature is true if the current parent→child edge being considered matches the one in the projected tree, and the POS tags of the parent and child are POS_parent and POS_{child}, respectively.

4 Experiments

As shown in Figure 2a, the L_F parser is trained with two kinds of input: (1) projected L_F trees from which `BOOL` and `TAG` features are extracted, and (2) "Gold" trees for L_F from which the standard features used by the MST parser are extracted.

We ran two sets of experiments. In the first, the "Gold" trees are the same as the projected L_F trees. This is to replicate the case when no gold standard is available for L_F. In the second set of experiments, the "Gold" trees are indeed the manually-corrected trees for the L_F sentences. The results are shown in Tables 2a and 2b, respectively.

In each set, there are several individual experiments. The "projection" row shows the results of evaluating the projected L_F trees directly without training a parser. The "MST baseline (B)" row shows the results when we train the MST parser without part-of-speech (POS) tag features and features from projected L_F trees. The third to seventh rows show the results when features from the projected L_F trees are added when training the MST parser. Finally, we want to see how well the system works if the projected L_F trees are perfect, so in the first two rows, we replace the projected L_F trees with the L_F trees from the gold standard. Because our data sets are small, we ran ten-fold cross validation. The highest results in each column for rows 3–9 are shown in bold.

5 Discussion

There are several observations to make from Table 2. First, comparing Tables 2a and 2b, it is clear that using the gold standard trees for L_F improves performance. Second, the projected trees are error-prone, as shown by the last row of the two tables, indicating linguistic divergence is very common. Third, in Table 2a, adding projection-derived features helps the MST baseline, but the results are not better than the "projection" row because the parser is trained on the projected trees only. In contrast, when the parser is trained on the correct parse trees with

	Hindi	German	Gaelic	Hausa	Korean	Malagasy	Welsh	Yaqui
B + Oracle	75.69	95.05	79.76	84.49	98.55	94.19	87.50	96.26
B + Oracle + POS	67.01	90.65	72.62	80.32	95.65	90.87	86.11	89.53
B + POS + Bool + Tag	66.09	87.07	73.41	78.70	**90.27**	89.21	86.11	84.04
B + POS + Tag	56.48	77.17	59.92	69.91	81.37	77.80	69.10	78.05
B + POS + Bool	66.67	87.62	74.21	79.40	89.44	88.59	84.38	85.04
B + POS	56.02	76.07	60.71	70.60	80.75	76.97	65.97	76.31
B + Bool	66.90	**87.90**	76.19	**80.56**	**90.27**	**90.25**	87.85	**86.53**
MST Baseline (B)	49.54	61.21	49.21	51.85	80.33	73.03	38.54	71.32
Projection	**67.82**	**87.90**	**78.57**	79.40	89.65	89.63	**89.58**	84.79

(a) Parse accuracy results for experiments where the parser was trained on automatically-produced projected trees for each language.

	Hindi	German	Gaelic	Hausa	Korean	Malagasy	Welsh	Yaqui
B + Oracle	98.03	98.07	95.63	99.31	99.17	97.93	98.26	96.51
B + Oracle + POS	97.34	97.94	89.29	94.44	98.34	97.72	94.44	97.26
B + POS + Bool + Tag	**79.05**	90.23	70.24	**88.66**	87.78	89.63	88.89	**86.53**
B + POS + Tag	76.74	83.49	65.87	82.64	82.40	79.46	72.92	83.79
B + POS + Bool	79.51	**91.47**	69.84	88.43	87.37	**90.25**	89.24	86.28
B + POS	77.20	82.12	63.10	83.80	80.75	81.54	75.00	81.80
B + Bool	77.31	87.76	70.24	87.73	**92.34**	89.42	**91.32**	85.54
MST Baseline (B)	65.16	62.72	55.16	72.22	80.75	73.03	51.39	66.08
Projection	67.82	87.90	**78.57**	79.40	89.65	89.63	89.58	84.79

(b) Parse accuracy results for experiments where the parser was trained on manually-corrected trees for each language.

Table 2: Parse accuracy results for all experiments. In each table, projection and baseline parser ("B") results are shown at the bottom. Oracle results, where the gold-standard trees (rather than projected trees) were used for the BOOL/TAG features, rather than projection are at top. "POS" represents the experiments where part-of-speech tags were projected from the English side, and "Bool" and "Tag" are features as explained in §3.3. The best result for the non-oracle runs is shown in bold.

additional features derived from projection (see Table 2b), the results are much better than both the MST baseline and projection, and often outperforms the heuristics-based projection algorithm as well. This indicates that, although projected trees are error-prone, using features from them indeed improves parsing performance.

6 Conclusion and Future Work

Building large-scale treebanks is very labor-intensive and often cost-prohibitive. As thousands of languages do not have such resources, syntactic projection has been proposed to transfer syntactic structure from resource-rich languages to resource-poor languages and the projected structure is used to bootstrap NLP tools. However, the projected structure is error-prone due to linguistic divergence.

We propose to augmenting a discriminative dependency parser by adding features extracted from projected structures, and have evaluated our system on eight typologically diverse languages, most of which are extremely resource-poor. The experiments show that the augmented parser outperforms both the original parser without the projection-derived features and the parse trees produced by the projection algorithm only. While the corpora used here are very small, they nonetheless are working examples and show that despite the linguistic divergence, parsing performance can be improved by using features extracted from the dependency trees produced by syntactic projection. Using the IGT data found in the ODIN database and elsewhere on the web, it is conceivable that our method can be applied to bootstrap dependency parsers for hundreds of languages at a very low cost.

For future work, there are several avenues we would like to investigate further. First, the results in this study have all been obtained from very small sets of data. Extending one or more of these languages out to a larger data set may give us a better understanding of the effects of incorporating IGT information. Second, we plan to extend our system to take advantage of a large amount of bitext if it is available. For that, we will train statistical word aligners and test how word alignment errors affect system performance. Finally, while the features used in this study look solely at the result of the projected trees, we would like to add features that look at different divergence types as discussed in Georgi et al. (2012).

Acknowledgments

This work is supported by the National Science Foundation Grant BCS-0748919.

References

Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D. M., and Xia, F. (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.

Collins, M. and Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Fifth International Conference on Language Resources and Evaluation*.

Dorr, B. J. (1994). Machine translation divergences: a formal description and proposed solution. *Computational Linguistics*, 20:597–633.

Ganchev, K., Gillenwater, J., and Taskar, B. (2009). Dependency grammar induction via bitext projection constraints. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 369–377.

Georgi, R., Xia, F., and Lewis, W. (2012). Measuring the divergence of dependency structures cross-linguistically to improve syntactic projection algorithms. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2004). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 1(1):1–15.

Hwa, R., Resnik, P., Weinberg, A., and Kolak, O. (2002). Evaluating translational correspondence using annotation projection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Klein, D. and Manning, C. (2001). An O(n^3) agenda-based chart parser for arbitrary probabilistic context-free grammars. Technical report.

Lewis, W. and Xia, F. (2008). Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing*.

Lewis, W. D. and Xia, F. (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.

McDonald, R., Lerman, K., and Pereira, F. (2006). Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 216–220. Association for Computational Linguistics.

Merlo, P., Stevenson, S., Tsang, V., and Allaria, G. (2002). A multilingual paradigm for automatic verb classification. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 207–214.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the ACL 2005*. Microsoft Research.

Xia, F. and Lewis, W. D. (2007). Multilingual Structural Projection across Interlinear Text. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

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