Bilingual Parsing with Factored Estimation: Using English to Parse Korean

David A. Smith and **Noah A. Smith**
Department of Computer Science
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
{d,n}asmith@cs.jhu.edu

Abstract

We describe how simple, commonly understood statistical models, such as statistical dependency parsers, probabilistic context-free grammars, and word-to-word translation models, can be effectively combined into a unified bilingual parser that jointly searches for the best English parse, Korean parse, and word alignment, where these hidden structures all constrain each other. The model used for parsing is completely factored into the two parsers and the TM, allowing separate parameter estimation. We evaluate our bilingual parser on the Penn Korean Treebank and against several baseline systems and show improvements parsing Korean with very limited labeled data.

1 Introduction

Consider the problem of parsing a language L for which annotated resources like treebanks are scarce. Suppose we have a small amount of text data with syntactic annotations and a fairly large corpus of parallel text, for which the other language (e.g., English) is not resource-impoverished. How might we exploit English parsers to improve syntactic analysis tools for this language?

One idea (Yarowsky and Ngai, 2001; Hwa et al., 2002) is to *project* English analysis onto L data, "through" word-aligned parallel text. To do this, we might use an English parser to analyze the English side of the parallel text and a word-alignment algorithm to induce word correspondences. By positing a coupling of English syntax with L syntax, we can induce structure on the L side of the parallel text that is in some sense isomorphic to the English parse.

We might take the projection idea a step farther. A statistical English parser can tell us much more than the hypothesized best parse. It can be used to find *every* parse admitted by a grammar, and also scores of those parses. Similarly, translation models, which yield word alignments, can be used in principle to score competing alignments and offer alternatives to a single-best alignment. It might also be beneficial to include the predictions of an L parser, trained on any available annotated L data, however few.

This paper describes how simple, commonly understood statistical models—statistical dependency parsers, probabilistic context-free grammars (PCFGs), and word translation models (TMs)—can be effectively combined into a unified framework that jointly searches for the best English parse, L parse, and word alignment, where these hidden structures are all constrained to be consistent. This inference task is carried out by a bilingual parser. At present, the model used for parsing is completely factored into the two parsers and the TM, allowing separate parameter estimation.

First, we discuss bilingual parsing (§2) and show how it can solve the problem of joint English-parse, L-parse, and word-alignment inference. In §3 we describe parameter estimation for each of the factored models, including novel applications of log-linear models to English dependency parsing and Korean morphological analysis. §4 presents Korean parsing results with various monolingual and bilingual algorithms, including our bilingual parsing algorithm. We close by reviewing prior work in areas related to this paper (§5).

2 Bilingual parsing

The joint model used by our bilingual parser is an instance of a stochastic bilingual multitext grammar (2-MTG), formally defined by Melamed (2003). The 2-MTG formalism generates two strings such that each syntactic constituent—including individual words—in one side of the bitext corresponds either to a constituent in the other side or to $\emptyset$.

Melamed defines bilexicalized MTG (L_2MTG), which is a synchronous extension of bilexical grammars such as those described in Eisner and Satta (1999) and applies the latter's algorithmic speedups to L_2MTG-parsing.

Our formalism is not a precise fit to either unlexicalized MTG or L_2MTG since we posit lexical dependency structure only in *one* of the languages (English). The primary rationale for this is that we are dealing with only a small quantity of labeled data in language L and therefore do not expect to be able to accurately estimate its lexical affinities. Further, synchronous parsing is in practice computationally expensive, and eliminating lexicalization on one side reduces the run-time of the parser from $O(n^8)$ to $O(n^7)$. Our parsing algorithm is a simple transformation of Melamed's R2D parser that eliminates head information in all Korean parser items.

The model event space for our stochastic "half-bilexicalized" 2-MTG consists of rewrite rules of the following two forms, with English above and L below:

$$\begin{pmatrix} X[h_1] \\ A \end{pmatrix} \rightarrow \begin{matrix} h_1 \\ h_2 \end{matrix}, \quad \begin{pmatrix} X[h_1] \\ A \end{pmatrix} \rightarrow \begin{matrix} Y[h_1]Z[c_1] \\ BC \end{matrix}$$

where upper-case symbols are nonterminals and lower-case symbols are words (potentially $\emptyset$). One approach to assigning a probability to such a rule is to make an independence assumption, for example:

$$\Pr_{\text{bi}}\left(\begin{array}{c} X[h] \\ A \end{array} \rightarrow \begin{array}{c} Y[h_1]Z[c] \\ BC \end{array}\right) = \Pr_{\text{English}}(X[h] \rightarrow Y[h_1]Z[c_1]) \cdot \Pr_{L}(A \rightarrow BC)$$

There are two powerful reasons to model the bilingual grammar in this factored way. First, we know of no tree-aligned corpora from which bilingual rewrite probabilities could be estimated; this rules out the possibility of supervised training of the joint rules. Second, separating the probabilities allows separate estimation of the probabilities—resulting in two well-understood parameter estimation tasks which can be carried out independently.[1]

This factored modeling approach bears a strong resemblance to the factored monolingual parser of Klein and Manning (2002), which combined an English dependency model and an unlexicalized PCFG. The generative model used by Klein and Manning consisted of multiplying the two component models; the model was therefore deficient.

We go a step farther, replacing the deficient generative model with a log-linear model. The underlying parsing algorithm remains the same, but the weights are no longer constrained to sum to one. (Hereafter, we assume weights are additive real values; a log-probability is an example of a weight.) The weights may be estimated using discriminative training (as we do for the English model, §3.1) or as if they were log-probabilities, using smoothed maximum likelihood estimation (as we do for the Korean model, §3.3). Because we use this model only for inference, it is not necessary to compute a partition function for the combined log-linear model.

In addition to the two monolingual syntax models, we add a word-to-word translation model to the mix. In this paper we use a translation model to induce only a single best word matching, but in principle the translation model could be used to weight all possible word-word links, and the parser would solve the joint alignment/parsing problem.[2]

As a testbed for our experiments, the Penn Korean Treebank (KTB; Han et al., 2002) provides 5,083 Korean constituency trees along with English translations and their trees. The KTB also analyzes Korean words into their component morphemes and morpheme tags, which allowed us to train a morphological disambiguation model.

To make the most of this small corpus, we performed all our evaluations using five-fold cross-validation. Due to the computational expense of bilingual parsing, we produced a sub-corpus of the KTB limiting English sentence length to 10 words, or 27% of the full data. We then randomized the order of sentences and divided the data into five equal test sets of 280 sentences each (≈1,700 Korean words, ≈2,100 English words). Complementing each test set, the remaining data were used for training sets of increasing size to simulate various levels of data scarcity.

[1] Of course, it might be the case that some information *is* known about the relationship between the two languages. In that case, our log-linear framework would allow the incorporation of additional bilingual production features.

[2] Although polynomial, we found this to be too computationally demanding to do with our optimal parser in practice, but with pruning and/or A* heuristics it is likely to be feasible.

3 Parameter estimation

We now describe parameter estimation for the four component models that combine to make our full system (Table 1).

3.1 English syntax model

Our English syntax model is based on weighted bilexical dependencies. The model predicts the generation of a child (POS tag, word) pair, dependent upon its parent (tag, word) and the tag of the parent's most recent child on the same side (left or right). These events correspond quite closely to the parser described by Eisner's (1996) model C, but instead of the rules receiving conditional probabilities, we use a log-linear model and allow arbitrary weights. The model does not predict POS tags; it assumes they are given, even in test.

Note that the dynamic program used for inference of bilexical parses is indifferent to the origin of the rule weights; they could be log-probabilities or arbitrary numbers, as in our model. The parsing algorithm need not change to accommodate the new parameterization.

In this model, the probability of a (sentence, tree) pair (E, T) is given by:

$$\Pr(E, T) = \frac{\exp\left(\mathbf{f}(E, T) \cdot \theta\right)}{\sum_{E', T'} \exp(\mathbf{f}(E', T') \cdot \theta)} \qquad (1)$$

where θ are the model parameters and $\mathbf{f}$ is a vector function such that f_i is equal to the number of times a feature (e.g., a production rule) fires in (E, T).

Parameter estimation consists of selecting weights θ to maximize the conditional probability of the correct parses given observed sentences:[3]

$$\prod_i \Pr(T_i | S_i) = \prod_i \frac{\exp\left(\mathbf{f}(E_i, T_i) \cdot \theta\right)}{\sum_{T'} \exp(\mathbf{f}(E_i, T') \cdot \theta)} \qquad (2)$$

Another important advantage of moving to log-linear models is the simple handling of data sparseness. The feature templates used by our model are shown in Table 2. The first feature corresponds to the fully-described child-generation event; others are similar but less informative. These "overlapping" features offer a kind of backoff, so that each child-generation event's weight receives a contribution from several granularities of description.

Feature selection is done by simple thresholding: if a feature is observed 5 times or more in the training set, its weight is estimated; otherwise its weight is locked at

[3] This can be done using iterative scaling or gradient-based numerical optimization methods, as we did.

Model	Formalism	Estimation	Role
English syntax (§3.1)	bilexical dependency grammar	discriminative estimation	combines with Korean syntax for bilingual parsing
Korean morphology (§3.2)	two-sequence trigram model	discriminative estimation over a lattice	best analysis used as input to TM training and to parsing
Korean syntax (§3.3)	PCFG	smoothed MLE	combines with English syntax for bilingual parsing
Translation model (§3.4)	IBM models 1–4, both directions	unsupervised estimation (approximation to EM)	best analysis used as input to bilingual parsing

Table 1: A summary of the factored models described in this paper and their interactions.

$\langle T_P, W_P, T_A, T_C, W_C, D\rangle$	$\langle T_P, T_A, T_C, D\rangle$
$\langle T_P, T_A, T_C, W_C, D\rangle$	$\langle T_P, W_P, T_A, T_C, D\rangle$
$\langle T_P, W_P, \text{stop}, T_C, D\rangle$	$\langle T_P, \text{stop}, T_C, D\rangle$

Table 2: Feature templates used by the English dependency parser. T_X is a tag and W_X is a word. P indicates the parent, A the previous child, and C the next-generated child. D is the direction (left or right). The last two templates correspond to stopping.

0. If a feature is never seen in training data, we give it the same weight as the minimum-valued feature from the training set ($\theta_{\min}$). To handle out-of-vocabulary (OOV) words, we treat any word seen for the first time in the final 300 sentences of the training corpus as OOV. The model is smoothed using a Gaussian prior with unit variance on every weight.

Because the left and right children of a parent are independent of each other, our model can be described as a weighted split head automaton grammar (Eisner and Satta, 1999). This allowed us to use Eisner and Satta's $O(n^3)$ parsing algorithm to speed up training.[4] This speedup could not, however, be applied to the bilingual parsing algorithm since a split parsing algorithm will preclude inference of certain configurations of word alignments that *are* allowed by a non-split parser (Melamed, 2003).

We trained the parser on sentences of 15 words or fewer in the WSJ Treebank sections 01–21.[5] 99.49% dependency attachment accuracy was achieved on the training set, and 76.68% and 75.00% were achieved on sections 22 and 23, respectively. Performance on the English side of our KTB test set was 71.82% (averaged across 5 folds, $\sigma = 1.75$).

This type of discriminative training has been applied to log-linear variants of hidden Markov models (Lafferty et al., 2001) and to lexical-functional grammar (Johnson et al., 1999; Riezler et al., 2002). To our knowledge, it has not been explored for context-free models (including bilexical dependency models like ours). A review of discriminative approaches to parsing can be found in Chiang (2003).

[4]Our split HAG's head automaton states correspond to the POS tags of the dependent words; this makes the head automaton deterministic and offers an additional speedup.

[5]The parser does not model POS-tags; we assume they are known. Head words in the WSJ corpus were obtained using R. Hwa's `const2dep` tool.

3.2 Korean morphological analysis

A Korean word typically consists of a head morpheme followed by a series of closed-class dependent morphemes such as case markers, copula, topicalizers, and conjunctions. Since most of the semantic content resides in the leading head morpheme, we eliminate for word alignment all trailing morphemes, which reduces the KTB's vocabulary size from 10,052 to 3,104.

Existing morphological processing tools for many languages are often unweighted finite-state transducers that encode the possible analyses for a surface form word. One such tool, `klex`, is available for Korean (Han, 2004).

Unfortunately, while the unweighted FST describes the set of valid analyses, it gives no way to choose among them. We treat this as a noisy channel: Korean morpheme-tag pairs are generated in sequence by some process, then passed through a channel that turns them into Korean words (with loss of information). The channel is given by the FST, but without any weights. To select the best output, we model the source process.

We model the sequence of morphemes and their tags as a log-linear trigram model. Overlapping trigram, bigram, and unigram features provide backoff information to deal with data sparseness (Table 3). For each training sentence, we used the FST-encoded morphological dictionary to construct a lattice of possible analyses. The lattice has a "sausage" form with all paths joining between each word.

We train the feature weights to maximize the weight of the correct path relative to all paths in the lattice. In contrast, Lafferty et al. (2001) train to maximize the the probability of the tags given the words. Over training sentences, maximize:

$$\prod_i \Pr(T_i, M_i|\text{lattice}) = \prod_i \frac{\exp(\mathbf{f}(T_i, M_i)\cdot\theta)}{\sum_{(T',M')\in\text{lattice}}\exp(\mathbf{f}(T', M')\cdot\theta)} \quad (3)$$

where T_i is the correct tagging for sentence i, M_i is the correct morpheme sequence.

There are a few complications. First, the coverage of the FST is of course not universal; in fact, it cannot analyze 4.66% of word types (2.18% of tokens) in the KTB.

$\langle T_{i-2}, M_{i-2}, T_{i-1}, M_{i-1}, T_i, M_i \rangle$	$\langle T_i \rangle$
$\langle T_{i-2}, M_{i-2}, T_{i-1}, M_{i-1}, T_i \rangle$	$\langle T_i, M_i \rangle$
$\langle T_{i-2}, T_{i-1}, M_{i-1}, T_i, M_i \rangle$	$\langle T_{i-1}, T_i \rangle$
$\langle T_{i-2}, T_{i-1}, M_{i-1}, T_i, \rangle$	$\langle T_{i-1}, T_i, M_i \rangle$
$\langle T_{i-2}, T_{i-1}, T_i, M_i \rangle$	$\langle T_{i-1}, M_{i-1}, T_i, \rangle$
$\langle T_{i-1}, M_{i-1}, T_i, M_i \rangle$	$\langle T_{i-2}, T_{i-1}, T_i \rangle$

Table 3: Feature templates used by the Korean morphology model. T_x is a tag, M_x is a morpheme.

We tag such words as atomic common nouns (the most common tag). Second, many of the analyses in the KTB are not admitted by the FST: 21.06% of correct analyses (by token) are not admitted by the FST; 6.85% do not have an FST analysis matching in the first tag and morpheme, 3.63% do not have an FST analysis matching the full tag sequence, and 1.22% do not have an analysis matching the first tag. These do not include the 2.18% of tokens with no analysis at all. When this happened in training, we added the correct analysis to the lattice.

To perform inference on new data, we construct a lattice from the FST (adding in any analyses of the word seen in training) and use a dynamic program (essentially the Viterbi algorithm) to find the best path through the lattice. Unseen features are given the weight $\theta_{\min}$. Table 4 shows performance on ambiguous tokens in training and test data (averaged over five folds).

3.3 Korean syntax model

Because we are using small training sets, parameter estimates for a *lexicalized* Korean probabilistic grammar are likely to be highly unreliable due to sparse data. Therefore we use an unlexicalized PCFG. Because the POS tags are given by the morphological analyzer, the PCFG need not predict words (i.e., head morphemes), only POS tags.

Rule probabilities were estimated with MLE. Since only the sentence nonterminal *S* was smoothed (using add-0.1), the grammar could parse any sequence of tags but was relatively sparse, which kept bilingual run-time down. [6]

When we combine the PCFG with the other models to do joint bilingual parsing, we simply use the logs of the PCFG probabilities as if they were log-linear weights. A PCFG treated this way is a perfectly valid log-linear model; the exponentials of its weights just happen to satisfy certain sum-to-one constraints.

In the spirit of joint optimization, we might have also combined the Korean morphology and syntax models into one inference task. We did not do this, largely out of concerns over computational expense (see the discussion of translation models in §3.4). This parser, independent of the bilingual parser, is evaluated in §4.

[6] We also found that this type of smoothing and smoothing *all* nonterminals gave indistinguishable results on monolingual parsing. Alternatively, we could have trained the PCFG discriminatively (treating the PCFG rules as log-linear features), but because our training sets are small we do not expect such training to be very different from training the PCFG as a generative model with probabilities.

3.4 Translation model

In our bilingual parser, the English and Korean parses are mediated through word-to-word translational correspondence links. Unlike the syntax models, the translation models were trained without the benefit of labeled data. We used the GIZA++ implementation of the IBM statistical translation models (Brown et al., 1993; Och and Ney, 2003).

To obtain reliable word translation estimates, we trained on a bilingual corpus in addition to the KTB training set. The Foreign Broadcast Information Service dataset contains about 99,000 sentences of Korean and 72,000 of English translation. For our training, we extracted a relatively small parallel corpus of about 19,000 high-confidence sentence pairs.

As noted above, Korean's productive agglutinative morphology leads to sparse estimates of word frequencies. We therefore trained our translation models after replacing each Korean word with its first morpheme stripped of its closed-class dependent morphemes, as described in §3.2.

The size of the translation tables made optimal bilingual parsing prohibitive by exploding the number of possible analyses. We therefore resorted to using GIZA++'s hypothesized alignments. Since the IBM models only hypothesize one-to-many alignments from target to source, we trained using each side of the bitext as source and target in turn. We could then produce two kinds of alignment graphs by taking either the **intersection** or the **union** of the links in the two GIZA++ alignment graphs. All words not in the resulting alignment graph are set to align to $\emptyset$.

Our bilingual parser deals only in one-to-one alignments (mappings); the intersection graph yields a mapping. The union graph yields a set of links which may permit different one-to-one mappings. Using the union graph therefore allows for flexibility in the word alignments inferred by the bilingual parser, but this comes at computational expense (because more analyses are permitted).

Even with over 20,000 sentence pairs of training data, the hypothesized alignments are relatively sparse. For the intersection alignments, an average of 23% of non-punctuation Korean words and 17% of non-punctuation English words have a link to the other language. For the union alignments, this improves to 88% for Korean and 22% for English.

A starker measure of alignment sparsity is the accuracy of English dependency links projected onto Korean. Following Hwa et al. (2002), we looked at dependency links in the true English parses from the KTB where both the dependent and the head were linked to words on the Korean side using the intersection alignment. Note that Hwa et al. used not only the true English trees, but also hand-produced alignments. If we hypothesize that, if English words i and j are in a parent-child relationship, then so are their linked Korean words, then we infer an incomplete dependency graph for the Korean sentences whose precision is around 49%–53% but whose recall is

	Training sentences	All tags	All morphemes	First tag	First morpheme
Training set accuracy on ambiguous tokens	32	91.14 (1.41)	94.25 (2.59)	91.14 (1.41)	95.74 (2.49)
	64	89.76 (0.34)	93.39 (1.12)	89.76 (0.34)	95.23 (1.43)
	128	88.19 (0.91)	92.48 (1.25)	88.38 (1.08)	94.43 (1.02)
	512	83.69 (0.94)	89.59 (0.27)	85.03 (1.08)	91.95 (0.21)
	1024	82.55 (0.68)	89.28 (0.30)	84.22 (0.77)	91.67 (0.19)
Test set accuracy on ambiguous tokens	32	59.34 (2.52)	53.13 (2.09)	72.81 (1.96)	84.99 (3.11)
	64	59.34 (2.41)	54.76 (1.64)	72.68 (1.79)	85.54 (2.03)
	128	60.85 (2.15)	57.20 (2.01)	74.44 (1.17)	86.29 (1.14)
	512	63.99 (2.02)	63.24 (1.28)	75.14 (0.86)	85.82 (1.01)
	1024	65.26 (1.85)	66.03 (1.72)	75.22 (1.25)	85.62 (1.08)

Table 4: Korean morphological analysis accuracy on ambiguous tokens in the training and test sets: means (and standard deviations) are shown over five-fold cross-validation. Over 65% of word tokens are ambiguous. The accuracy of the first tag in each word affects the PCFG and the accuracy of the first morpheme affects the translation model (under our aggressive morphological lemmatization).

an abysmal 2.5%–3.6%. [7]

4 Evaluation

Having trained each part of the model, we bring them together in a unified dynamic program to perform inference on the bilingual text as described in §2. In order to experiment easily with different algorithms, we implemented all the morphological disambiguation and parsing models in this paper in Dyna, a new language for weighted dynamic programming (Eisner et al., 2004). For parameter estimation, we used the complementary DynaMITE tool. Just as CKY parsing starts with words in its chart, the dynamic program chart for the bilingual parser is seeded with the links given in the hypothesized word alignment.

All our current results are optimal under the model, but as we scale up to more complex data, we might introduce A* heuristics or, at the possible expense of optimality, a beam search or pruning techniques. Our agenda discipline is uniform-cost search, which guarantees that the first full parse discovered will be optimal—if none of the weights are positive. In our case we are maximizing sums of negative weights, as if working with log probabilities.[8]

When evaluating our parsing output against the test data from the KTB, we do not claim credit for the single outermost bracketing or for unary productions. Since unary productions do not translate well from language to language (Hwa et al., 2002), we collapse them to their lower nodes.

4.1 Baseline systems

We compare our bilingual parser to several baseline systems. The first is the Korean PCFG trained on the small KTB training sets, as described in §3.3. We also consider Wu's (1997) stochastic inversion transduction grammar (SITG) as well as strictly left- and right-branching trees. We report the results of five-fold cross-validation with the mean and standard deviation (in parentheses).

Since it is unlexicalized, the PCFG parses sequences of tags as output by the morphological analysis model. By contrast, we can build translation tables for the SITG directly from surface words—and thus not use any labeled training data at all—or from the sequence of head morphemes. Experiments showed, however, that the SITG using words consistently outperformed the SITG using morphemes. We also implemented Wu's tree-transformation algorithm to turn full binary-branching SITG output into flatter trees. Finally, we can provide extra information to the SITG by giving it a set of English bracketings that it must respect when constructing the joint tree. To get an upper bound on performance, we used the true parses from the English side of the KTB.

Only the PCFG, of course, can be evaluated on labeled bracketing (Table 6). Although labeled precision and recall on test data generally increase with more training data, the slightly lower performance at the highest training set size may indicate overtraining of this simple model. Unlabeled precision and recall show continued improvement with more Korean training data.

Even with help from the true English trees, the unsupervised SITGs underperform PCFGs trained on as few as 32 sentences, with the exception of unlabeled recall in one experiment. It seems that even some small amount of knowledge of the language helps parsing. Crossing brackets for the flattened SITG parses are understandably lower.

4.2 Bilingual parsing

The output of our bilingual parser contains three types of constituents: English-only (aligned to $\emptyset$), Korean-only (aligned to $\emptyset$), and bilingual. The Korean parse induced by the Korean-only and bilingual constituents is filtered so constituents with intermediate labels (generated by the binarization process) are eliminated.

A second filter we consider is to keep only the (re-

[7] We approximated head-words in the Korean gold-standard trees by assuming all structures to be head-final, with the exception of punctuation. That is, the head-words of sister constituents will elect the right-most, non-punctuation word among them as the head.

[8] In fact the English syntax model is not constrained to have non-positive weights, but we decrement every parameter by $\theta_{\max}$. For a given sentence, this will reduce every possible parse's weight by a constant value, since the same number of features fire in every parse; thus, the classification properties of the parser are not affected.

maining) bilingual constituents corresponding to an English head word's maximal span. This filter will eliminate constituents whose English correspondent is a head word with *some* (but not all) of its dependents. Such partial English constituents are by-products of the parsing and do not correspond to the modeled syntax.

With good word alignments, the English parser can help disambiguate Korean phrase boundaries and overcome erroneous morphological analyses (Table 5). Results without and with the second filter are shown in Table 7. Because larger training datasets lead to larger PCFGs (with more rules), the grammar constant increases. Our bilingual parser implementation is on the cusp of practicality (in terms of memory requirements); when the grammar constant increased, we were unable to parse longer sentences. Therefore the results given for bilingual parsing are on reduced test sets, where a length filter was applied: sentences with $|E| + |F| > \tau$ were removed, for varying values of τ.

4.3 Discussion

While neither bilingual parser consistently beats the PCFG on its own, they offer slight, complementary improvements on small training datasets of 32 and 64 sentences (Table 7). The bilingual parser without the English head span filter gives a small recall improvement on average at similar precision. Neither of these differences is significant when measured with a paired-sample t-test.

In contrast, the parser *with* the English head span filter sacrifices significantly on recall for a small but significant gain in precision at the 0.01 level. Crossing brackets at all levels are significantly lower with the English head span filter. We can describe this effect as a filtering of Korean constituents by the English model and word alignments. Constituents that are not strongly evident on the English side are simply removed. On small training datasets, this effect is positive: although good constituents are lost so that recall is poor compared to the PCFG, precision and crossing brackets are improved.

As one would expect, as the amount of training data increases, the advantage of using a bilingual parser vanishes—there is no benefit from falling back on the English parser and word alignments to help disambiguate the Korean structure. Since we have not pruned our search space in these experiments, we can be confident that all variations are due to the influence of the translation and English syntax models.

Our approach has this principal advantage: the various morphology, parsing, and alignment components can be improved or replaced easily without needing to retrain the other modules. The low dependency projection results (§3.4), in conjunction with our modest overall gains, indicate that the alignment/translation model should receive the most attention. In all the bilingual experiments, there is a small positive correlation (0.3), for sentences at each length, between the proportion of Korean words aligned to English and measures of parsing accuracy. Improved English parsers—such as Collins' models—have also been implemented in Dyna, the dynamic programming framework used here (Eisner et al., 2004).

5 Prior work

Combining separately trained systems and then searching for an (ideally) optimal solution is standard practice in statistical continuous speech recognition (Jelinek, 1998) and statistical machine translation (Brown et al., 1990). Composition is even more of a staple in finite-state frameworks (Knight and Graehl, 1998). Finally, factored models involving parses have been used to guide search. Charniak et al. (2003) combine separately trained parse production probabilities with translation probabilities to prune a parse forest hypothesized by the translation model. As discussed in §2, Klein and Manning (2002) guide their parser's search using a combination of separate unlexicalized PCFG and lexical dependency models.

The extent to which assumptions about similarity of syntax across languages are empirically valid has received attention in a few pilot studies. Fox (2002) has considered English and French, and Hwa et al. (2002) investigate Chinese and English. Xia et al. (2000) compare the rule templates of lexicalized tree adjoining grammars extracted from treebanks in English, Chinese, and Korean. In the context of machine translation, Dorr (1994) investigated divergences between two languages' structures.

Some proposals have sidestepped the empirical issue entirely. Wu (1997) and Alshawi et al. (2000) used unsupervised learning on parallel text to induce syntactic analysis that was useful for their respective applications in phrasal translation extraction and speech translation, though not necessarily similar to what a human annotator would select. Note a point of divergence of the SITG from our bilingual parsing system: SITG only allows words, but not higher structures, to match null in the other language and thus requires that the trees in parallel sentences be isomorphic. Yamada and Knight (2001) introduced tree-to-string alignment on Japanese data, and Gildea (2003) performed tree-to-tree alignment on the Korean Treebank, allowing for non-isomorphic structures; he applied this to word-to-word alignment. Finally, inspired by these intuitive notions of translational correspondence, Cherry and Lin (2003) include dependency features in a word alignment model to improve non-syntactic baseline systems.

In more formal work, Melamed (2003) proposes multitext grammars and algorithms for parsing them. Shieber and Schabes (1990) describe a synchronous tree adjoining grammar. While both of these formalisms require bilingual grammar rules, Eisner (2003) describes an algorithm for learning tree substitution grammars from unaligned trees.

Working on the Penn Korean Treebank, Sarkar and Han (2002) made a single training/test split and used 91% of the sentences to train a morphological disambiguator and lexicalized tree adjoining grammar (LTAG) based parsing system.

For a monolingual approach to training a parser with scarce resources, see (Steedman et al., 2003), who apply co-training and corrected co-training to bootstrapping an English parser starting with 1000 parsed training sen-

Truth	[TOP [NP ngyen.tay/NNC kong.pyeng/NNC cwung.tay/NNC]			[VP [NP ku/DAN to/NNC] ken.sel/NNC]	./SFN]
PCFG	[TOP ngyen.tay/VV [S [NP kong.pyeng/NNC cwung.tay/NNC]			[VP [NP ku/NPN to/NNX] ken.sel/NNC]	./SFN]]
Bilingual	[TOP [NP ngyen.tay/VV kong.pyeng/NNC$_1$ cwung.tay/NNC]			[VP [NP ku/NPN to/NNX] ken.sel/NNC]	./SFN$_2$]
Translation	The regimental$_1$ engineer company			constructed that road	.$_2$

Truth	[TOP [NP ku/DAN	sa.lam/NNC]	[NP ceng.chi/NNC	kwun.kwan/NNC]	?/SFN]
PCFG	[TOP [VP [NP ku/DAN	sa.lam/NNC	ceng.chi/NNC]	kwun.kwan/NNC]	?/SFN]
Bilingual	[TOP [NP ku/DAN$_1$	sa.lam/NNC]	[NP ceng.chi/NNC$_2$	kwun.kwan/NNC$_3$]	?/SFN$_4$]
Translation	He$_1$ is		a political$_2$	officer$_3$	?$_4$

Table 5: The gold-standard parse, PCFG parse, bilingual parse, and English translation for two selected test sentences. GIZA-aligned words are coindexed with subscripts. The bilingual parser recovers from erroneous morphological tagging in the first sentence and finds the proper NP bracketing in the second.

Method	Training Sentences	Unlabeled Precision	Unlabeled Recall	Labeled Precision	Labeled Recall	Crossing Brackets
PCFG training	32	57.03 (5.45)	78.45 (5.71)	51.13 (6.14)	70.26 (6.40)	0.71 (0.22)
	64	54.96 (4.98)	76.91 (6.71)	46.94 (4.38)	65.69 (5.99)	0.72 (0.25)
	128	52.60 (3.15)	73.20 (4.97)	43.46 (3.34)	60.48 (5.14)	0.82 (0.18)
	512	50.82 (1.46)	70.98 (2.00)	39.47 (2.49)	55.12 (3.42)	0.87 (0.06)
	1024	50.25 (0.82)	70.31 (1.32)	37.93 (1.45)	53.07 (2.16)	0.89 (0.04)
PCFG test	32	43.63 (4.40)	45.96 (5.38)	31.67 (3.47)	33.36 (4.19)	1.27 (0.16)
	64	45.90 (2.30)	46.68 (2.92)	34.29 (2.35)	34.91 (3.22)	1.18 (0.12)
	128	48.07 (4.14)	48.47 (4.45)	36.39 (3.37)	36.68 (3.50)	1.15 (0.14)
	512	50.88 (2.97)	51.89 (2.92)	**38.10** (3.22)	**38.82** (2.68)	1.10 (0.10)
	1024	**51.15** (2.17)	52.65 (1.74)	37.47 (1.89)	38.58 (1.64)	1.12 (0.08)
SITG	–	30.65 (1.97)	45.22 (3.43)	–	–	1.93 (0.17)
Flat SITG	–	41.78 (1.98)	33.59 (3.36)	–	–	0.94 (0.08)
SITG w/Eng. constit.	–	36.28 (0.70)	**52.68** (1.03)	–	–	1.60 (0.07)
Flat SITG w/Eng. constit.	–	42.55 (1.32)	30.64 (1.37)	–	–	**0.77** (0.06)
L-branching	–	25.62 (1.07)	35.83 (1.39)	–	–	2.04 (0.04)
R-branching	–	27.59 (1.03)	38.60 (1.75)	–	–	2.06 (0.11)

Table 6: Baseline parsing performance on Korean: the table shows means (and standard deviations) for five-fold cross-validation. The SITG system is evaluated on test data, but is trained without labeled data; the SITG with English trees uses true treebank English parses to constrain the search and thus represents an upper bound. The table shows means and standard deviations for five-fold cross-validation. The best test results in each column are in bold.

Method	Max. $\|E\| + \|F\|$ Test Sen. Length	Training Sentences	Unlabeled Precision	Unlabeled Recall	Labeled Precision	Labeled Recall	Crossing Brackets
PCFG	20	32	44.19 (4.41)	46.51 (5.32)	32.10 (3.47)	33.78 (4.14)	1.23 (0.16)
	20	64	46.39 (2.45)	47.03 (3.01)	34.69 (2.40)	35.20 (3.22)	1.15 (0.11)
	18	128	49.86 (4.83)	49.63 (4.74)	37.78 (3.74)	37.60 (3.61)	1.03 (0.13)
	17	512	53.89 (3.60)	54.60 (3.73)	40.61 (3.84)	41.10 (3.19)	0.87 (0.11)
	15	1024	57.87 (3.75)	59.39 (3.35)	43.92 (3.52)	45.07 (3.26)	0.61 (0.09)
Bilingual parsing	20	32	44.17 (3.97)	47.10 (4.81)	31.67 (3.65)	33.78 (4.29)	1.22 (0.14)
	20	64	46.30 (2.46)	47.73 (2.83)	34.14 (2.60)	35.23 (3.35)	1.15 (0.12)
	18	128	48.75 (3.64)	49.51 (4.08)	36.95 (2.65)	37.52 (2.92)	1.04 (0.10)
	17	512	52.77 (3.92)	54.21 (4.42)	39.73 (3.68)	40.78 (3.56)	0.88 (0.12)
	15	1024	56.70 (4.79)	58.85 (4.10)	43.09 (4.24)	44.71 (3.69)	0.60 (0.12)
Bilingual parsing, English head span filter	20	32	**45.65** (5.81)	28.83 (4.35)	**32.92** (4.60)	20.82 (3.53)	**0.72** (0.11)
	20	64	**47.15** (2.88)	28.73 (1.79)	34.65 (2.36)	21.14 (1.73)	**0.68** (0.08)
	18	128	49.65 (4.52)	28.74 (2.30)	**38.62** (3.69)	22.35 (1.76)	**0.59** (0.09)
	17	512	52.03 (4.21)	29.47 (2.71)	39.80 (2.92)	22.51 (1.32)	**0.50** (0.08)
	15	1024	54.78 (5.20)	29.74 (1.91)	42.01 (5.05)	22.78 (1.84)	**0.34** (0.09)

Table 7: Bilingual parsing performance on Korean: the table shows means (and standard deviations) for five-fold cross-validation. Bold-faced numbers in the bilingual parsers indicate **significant** improvements on the PCFG baseline using the paired-sample t-test at the 0.01 level.

tences. Although this technique has interesting properties, our combined optimization should be more stable since it does not involve iterative example selection.

6 Conclusion

We have presented a novel technique for merging simple, separately trained models for Korean parsing, English dependency parsing, and word translation, and optimizing the joint result using dynamic programming. We showed small but significant improvements for Korean parsers trained on small amounts of labeled data.

7 Acknowledgements

We would like to thank Elliott Drábek, Jason Eisner, Eric Goldlust, Philip Resnik, Charles Schafer, David Yarowsky, and the reviewers for their comments and assistance and Chung-hye Han, Na-Rae Han, and Anoop Sarkar for their help with the Korean resources. This work was supported under a National Science Foundation Graduate Research Fellowship and a Fannie and John Hertz Foundation Fellowship.

References

H. Alshawi, S. Bangalore, and S. Douglas. 2000. Learning dependency translation models as collections of finite-state head transducers. *Computational Linguistics*, 26(1):45–60.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

P. E. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

E. Charniak, K. Knight, and K. Yamada. 2003. Syntax-based language models for machine translation. In *Proc. MT Summit IX*.

C. Cherry and D. Lin. 2003. A probability model to improve word alignment. In *Proc. ACL*.

D. Chiang. 2003. Mildly context-sensitive grammars for estimating maximum entropy models. In *Proc. Formal Grammar*.

B. J. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

J. Eisner and G. Satta. 1999. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proc. ACL*.

J. Eisner, E. Goldlust, and N. A. Smith. 2004. Dyna: A declarative language for implementing dynamic programs. In *ACL Companion Vol.*

J. Eisner. 1996. An empirical comparison of probability models for dependency grammar. Technical Report IRCS-96-11, U. Penn.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *ACL Companion Vol.*

H. J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proc. EMNLP*.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proc. ACL*.

C.-H. Han, N.-R. Han, E.-S. Ko, H. Yi, and M. Palmer. 2002. Penn Korean Treebank: Development and evaluation. In *Proc. Pacific Asian Conf. Language and Comp.*

N.-R. Han. 2004. Klex: Finite-state lexical transducer for Korean. http://wave.ldc.upenn.edu/Catalog/-CatalogEntry.jsp?catalogId=LDC2004L01.

R. Hwa, P. Resnik, A. Weinberg, and O. Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proc. ACL*.

F. Jelinek. 1998. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA.

M. Johnson, S. Geman, S. Canon, Z. Chi, and S. Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proc. ACL*.

D. Klein and C. D. Manning. 2002. Fast exact natural language parsing with a factored model. In *NIPS*.

K. Knight and J. Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4).

J. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*.

I. D. Melamed. 2003. Multitext grammars and synchronous parsers. In *Proc. HLT-NAACL*.

F.-J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

S. Riezler, T. King, R. Kaplan, R. Crouch, J. Maxwell, and M. Johnson. 2002. Parsing the WSJ using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proc. ACL*.

A. Sarkar and C.-H. Han. 2002. Statistical morphological tagging and parsing of Korean with an LTAG grammar. In *Proc. TAG+6*, pages 48–56.

S. M. Shieber and Y. Schabes. 1990. Synchronous tree-adjoining grammars. In *Proc. ACL*, pages 253–258.

M. Steedman, R. Hwa, S. Clark, M. Osborne, A. Sarkar, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Example selection for bootstrapping statistical parsers. In *Proc. HLT-NAACL*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.

F. Xia, C.-H. Han, M. Palmer, and A. Joshi. 2000. Comparing lexicalized treebank grammars extracted from Chinese, Korean, and English corpora. In *Proc. 2nd Chinese Language Processing Workshop*.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proc. ACL*.

D. Yarowsky and G. Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proc. NAACL*.