Utilizing Extra-sentential Context for Parsing

Jackie Chi Kit Cheung and Gerald Penn
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3G4, Canada
{jcheung,gpenn}@cs.toronto.edu

Abstract

Syntactic consistency is the preference to reuse a syntactic construction shortly after its appearance in a discourse. We present an analysis of the WSJ portion of the Penn Treebank, and show that syntactic consistency is pervasive across productions with various left-hand side nonterminals. Then, we implement a reranking constituent parser that makes use of extra-sentential context in its feature set. Using a linear-chain conditional random field, we improve parsing accuracy over the generative baseline parser on the Penn Treebank WSJ corpus, rivalling a similar model that does not make use of context. We show that the context-aware and the context-ignorant rerankers perform well on different subsets of the evaluation data, suggesting a combined approach would provide further improvement. We also compare parses made by models, and suggest that context can be useful for parsing by capturing structural dependencies between sentences as opposed to lexically governed dependencies.

1 Introduction

Recent corpus linguistics work has produced evidence of syntactic consistency, the preference to reuse a syntactic construction shortly after its appearance in a discourse (Gries, 2005; Dubey et al., 2005; Reitter, 2008). In addition, experimental studies have confirmed the existence of syntactic priming, the psycholinguistic phenomenon of syntactic consistency[1]. Both types of studies, however, have limited the constructions that are examined to particular syntactic constructions and alternations. For instance, Bock (1986) and Gries (2005) examine specific constructions such as the passive voice, dative alternation and particle placement in phrasal verbs, and Dubey et al. (2005) deal with the internal structure of noun phrases. In this work, we extend these results and present an analysis of the distribution of all syntactic productions in the Penn Treebank WSJ corpus. We provide evidence that syntactic consistency is a widespread phenomenon across productions of various types of *LHS* nonterminals, including all of the commonly occurring ones.

Despite this growing evidence that the probability of syntactic constructions is not independent of the extra-sentential context, current high-performance statistical parsers (e.g. (Petrov and Klein, 2007; McClosky et al., 2006; Finkel et al., 2008)) rely solely on intra-sentential features, considering the particular grammatical constructions and lexical items within the sentence being parsed. We address this by implementing a reranking parser which takes advantage of features based on the context surrounding the sentence. The reranker outperforms the generative baseline parser, and rivals a similar model that does not make use of context. We show that the context-aware and the context-ignorant models perform well on different subsets of the evaluation data, suggesting a feature set that combines the two models would provide further improvement. Analysis of the rerankings made provides cases where contextual information has clearly improved parsing per-

[1] Whether or not corpus-based studies of consistency have any bearing on syntactic priming as a reality in the human mind is a subject of debate. See (Pickering and Branigan, 1999) and (Gries, 2005) for opposing viewpoints.

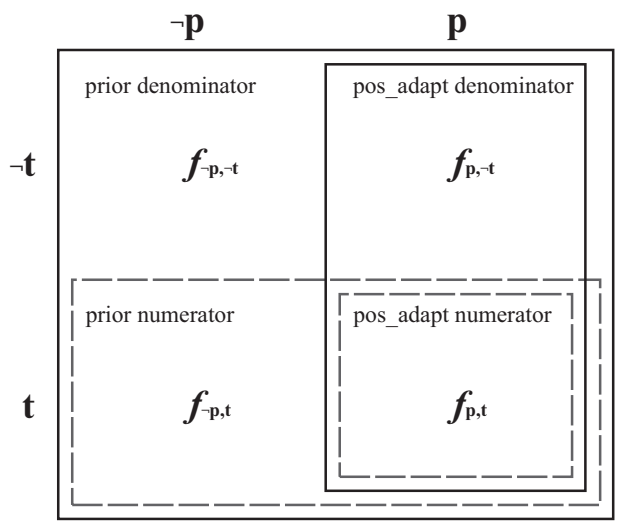

Figure 1: Visual representation of calculation of prior and positive adaptation probabilities. **t** represents the presence of a construction in the target set. **p** represents the presence of the construction in the prime set.

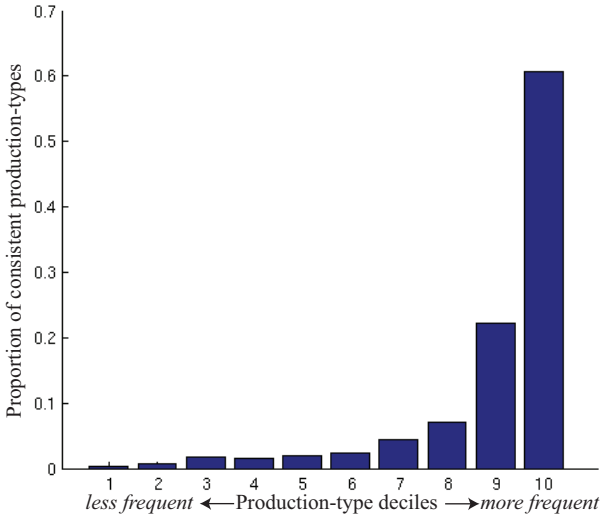

Figure 2: Production-types (singletons removed) categorized into deciles by frequency and the proportion of the production-types in that bin that is consistent to a significant degree.

formance, indicating the potential of extra-sentential contextual information to aid parsing, especially for structural dependencies between sentences, such as parallelism effects.

2 Syntactic Consistency in the Penn Treebank WSJ

Syntactic consistency has been examined by Dubey et al. (2005) for several English corpora, including the WSJ, Brown, and Switchboard corpora. They have provided evidence that syntactic consistency exists not only within coordinate structures, but also in a variety of other contexts, such as within sentences, between sentences, within documents, and between speaker turns in the Switchboard corpus. However, their analysis rests on a selected number of constructions concerning the internal structure of noun phrases. We extend their result here to arbitrary syntactic productions.

There have also been studies into syntactic consistency that consider all syntactic productions in dialogue corpora (Reitter, 2008; Buch and Pietsch, 2010). These studies find an inverse correlation between the probability of the appearance of a syntactic structure and the distance since its last occurrence, which indicates syntactic consistency. These studies, however, do not provide consistency results on subsets of production-types, such as by production LHS as our study does, so the implications that can be drawn from them for improving parsing are less apparent.

We adopt the measure used by Dubey et al. (2005) to quantify syntactic consistency, *adaptation probability*. This measure originates in work on lexical priming (Church, 2000), and quantifies the probability of a target word or construction w appearing in a "primed" context. Specifically, four frequencies are calculated, based on whether the target construction appears in the previous context (the prime set), and whether the construction appears after this context (the target set):

$$\begin{aligned} f_{p,\neg t}(w) &= \text{\# of times } w \text{ in prime set only} \\ f_{\neg p,t}(w) &= \text{\# of times } w \text{ in target set only} \\ f_{\neg p,\neg t}(w) &= \text{\# of times } w \text{ in neither set} \\ f_{p,t}(w) &= \text{\# of times } w \text{ in both sets} \end{aligned}$$

We also define N to be the sum of the four fre-

LHS	$prior$	pos_adapt	ratio	$+ > prior$ sig.	insig.	$+ < prior$ sig.
ADJP	0.03	0.05	1.96	26	251	0
ADVP	0.21	0.24	1.15	26	122	0
NP	0.17	0.22	1.27	281	2284	0
PP	0.56	0.58	1.04	32	125	0
PRN	0.01	0.03	4.60	12	82	0
PRT	0.06	0.08	1.40	3	3	0
QP	0.03	0.18	5.41	24	147	0
S	0.30	0.34	1.13	42	689	1
SBAR	0.15	0.20	1.31	13	68	0
SINV	0.01	0.01	1.00	3	77	0
VP	0.08	0.12	1.56	148	1459	0
WHADVP	0.04	0.08	1.84	2	8	0
WHNP	0.07	0.10	1.39	3	47	0
WHPP	0.01	0.02	2.65	1	1	0

Table 1: Weighted average by production frequency among non-singleton production-types of prior and positive adaptation probabilities, and the ratio between them. The columns on the right show the number of production-types for which the positive adaptation probability is significantly greater than, not different from, or less than the prior probability. We exclude *LHS*s with a weighted average prior of less than 0.005, due to the small sample size.

quencies. Then, we define the *prior* and the *positive adaptation* probability of a construction as follows (See also Figure 1):

$$prior(w) = \frac{f_{p,t}(w) + f_{\neg p,t}(w)}{N}$$

$$pos\_adapt(w) = \frac{f_{p,t}(w)}{f_{p,t}(w) + f_{p,\neg t}(w)}$$

A positive adaptation probability that is greater than the prior probability would be interpreted as evidence for syntactic consistency for that construction. We conduct χ^2 tests for statistical significance testing. We analyze the Penn Treebank WSJ corpus according this schema for all productions that occur in sections 2 to 22. These are the standard training and development sets for training parsers. We did not analyze section 23 in order not to use its characteristics in designing our reranking parser so that we can use this section as our evaluation test set. Our analysis focuses on the consistency of rules between sentences, so we take the previous sentence within the same article as the prime set, and the current sentence as the target set in calculating the probabilities given above. The raw data from which we produced our analysis are available at `http://www.cs.toronto.edu/~jcheung/wsj_parallelism_data.txt`.

We first present results for consistency in all the production-types[2], grouped by the *LHS* of the production. Table 1 shows the weighted average prior and positive adaptation probabilities for productions by *LHS*, where the weighting is done by the number of occurrence of that production. Production-types that only occur once are removed. It also shows the number of production-types in which the positive adaptation probability is statistically significantly greater than, not significantly different from, and significantly lower than the prior probability.

Quite remarkably, very few production-types are significantly less likely to reoccur compared to the prior probability. Also note the wide variety of *LHS*s for which there is a large number of production-types that are consistent to a significant degree. While a large number of production-types appears not to be significantly more likely to occur in a primed context, this is due to the large number of production-types which only appear a few times. Frequently occurring production-types mostly exhibit syntactic consistency.

We show this in Figure 2, in which we put non-singleton production-types into ten bins by fre-

[2]That is, all occurrences of a production with a particular *LHS* and *RHS*.

Ten most frequent production-types						
production	$f_{\neg p,t}$	$f_{p,t}$	$f_{p,\neg t}$	$prior$	pos_adapt	ratio
PP → IN NP	5624	26224	5793	0.80	0.82	1.02
NP → NP PP	9033	12451	9388	0.54	0.57	1.05
NP → DT NN	9198	10585	9172	0.50	0.54	1.07
S → NP VP	8745	9897	9033	0.47	0.52	1.11
S → NP VP .	8576	8501	8888	0.43	0.49	1.13
S → VP	8717	7867	9042	0.42	0.47	1.11
NP → PRP	7208	5309	7285	0.32	0.42	1.33
ADVP → RB	7986	3949	7905	0.30	0.33	1.10
NP → NN	7630	3390	7568	0.28	0.31	1.11
VP → TO VP	7039	3552	7250	0.27	0.33	1.23
Ten most consistent among 10% most frequent production-types						
production	$f_{\neg p,t}$	$f_{p,t}$	$f_{p,\neg t}$	$prior$	pos_adapt	ratio
QP → # CD CD	51	18	45	0.00	0.29	163.85
NP → JJ NNPS	52	7	53	0.00	0.12	78.25
NP → NP , ADVP	109	24	99	0.00	0.20	58.05
NP → DT JJ CD NN	63	6	67	0.00	0.08	47.14
PP → IN NP NP	83	10	87	0.00	0.10	43.86
QP → IN $ CD	51	3	49	0.00	0.06	42.28
NP → NP : NP .	237	128	216	0.01	0.37	40.34
INTJ → UH	59	4	60	0.00	0.06	39.26
ADVP → IN NP	108	11	83	0.00	0.12	38.91
NP → CD CD	133	21	128	0.00	0.14	36.21

Table 2: Some instances of consistency effects of productions. All productions' pos_adapt probability is significantly greater than its $prior$ probability at $p < 10^{-6}$.

quency and calculated the proportion of production-types in that bin for which the positive adaptation probability is significantly greater than the prior. It is clear that the most frequently occurring production-types are also the ones most likely to exhibit evidence of syntactic consistency.

Table 2 shows the breakdown of the prior and positive adaptation calculation components for the ten most frequent production-types and the ten most consistent (by the ratio pos_adapt / $prior$) productions among the top decile of production-types. Note that all of these production-types are consistent to a statistically significant degree. Interestingly, many of the most consistent production-types have NP as the *LHS*, but overall, productions with many different *LHS* parents exhibit consistency.

3 A Context-Aware Reranker

Having established evidence for widespread syntactic consistency in the WSJ corpus, we now investigate incorporating extra-sentential context into a statistical parser. The first decision to make is whether to incorporate the context into a generative or a discriminative parsing model.

Employing a generative model would allow us to train the parser in one step, and one such parser which incorporates the previous context has been implemented by Dubey et al. (2006). They implement a PCFG, learning the production probabilities by a variant of standard PCFG-MLE probability estimation that conditions on whether a rule has recently occurred in the context or not:

$$P(RHS|LHS, Prime) = \frac{c(LHS \rightarrow RHS, Prime)}{c(LHS, Prime)}$$

LHS and *RHS* represent the left-hand side and

right-hand side of a production, respectively. *Prime* is a binary variable which is `True` if and only if the current production has occurred in the prime set (the previous sentence). *c* represents the frequency count.

The drawback of such a system is that it doubles the state space of the model, and hence likely increases the amount of data needed to train the parser to a comparable level of performance as a more compact model, or would require elaborate smoothing. Dubey et al. (2006) find that this system performs worse than the baseline PCFG-MLE model, dropping F1 from 73.3% to 71.6%[3].

We instead opt to incorporate the extra-sentential context into a discriminative reranking parser, which naturally allows additional features to be incorporated into the statistical model. Many discriminative models of constituent parsing have been proposed in recent literature. They can be divided into two broad categories–those that rerank the N-best outputs of a generative parser, and those that make all parsing decisions using the discriminative model. We choose to implement an N-best reranking parser so that we can utilize state-of-the-art generative parsers to ensure a good selection of candidate parses to feed into our reranking module. Also, fully discriminative models tend to suffer from efficiency problems, though recent models have started to overcome this problem (Finkel et al., 2008).

Our approach is similar to N-best reranking parsers such as Charniak and Johnson (2005) and Collins and Koo (2005), which implement a variety of features to capture within-sentence lexical and structural dependencies. It is also similar to work which focuses on coordinate noun phrase parsing (e.g. (Hogan, 2007; Kübler et al., 2009)) in that we also attempt to exploit syntactic parallelism, but in a between-sentence setting rather than in a within-sentence setting that only considers coordination.

As evidence of the potential of an N-best reranking approach with respect to extra-sentential context, we considered the 50-best parses in the development set produced by the generative parser, and categorized each into one of nine bins depending on whether this candidate parse exhibits more, less, or the same amount of rule overlap with the previous correct parse than the generative baseline, and whether the candidate parse has a better, worse, or the same F1 measure than the generative baseline (Table 3). We find that a larger percentage of candidate parses which share more productions with the previous parse are better than the generative baseline parse than for the other categories, and this difference is statistically significant (χ^2 test).

	Overlap		
	less	*equal*	*more*
worse F1	32519 (81.8%)	7224 (69.3%)	17280 (75.4%)
equal F1	1023 (2.6%)	1674 (16.1%)	540 (2.4%)
better F1	6224 **(15.7%)**	1527 **(14.6%)**	5106 **(22.3%)**

Table 3: Correlation between rule overlap and F1 compared to the generative baseline for the 50-best parses in the development set.

3.1 Conditional Random Fields

For our statistical reranker, we implement a linear-chain conditional random field (CRF). CRFs are a very flexible class of graphical models which have been used for various sequence and relational labelling tasks (Lafferty et al., 2001). They have been used for tree labelling, in XML tree labelling (Jousse et al., 2006) and semantic role labelling tasks (Cohn and Blunsom, 2005). They have also been used for shallow parsing (Sha and Pereira, 2003), and full constituent parsing (Finkel et al., 2008; Tsuruoka et al., 2009). We exploit the flexibility of CRFs by incorporating features that depend on extra-sentential context.

In a linear-chain CRF, the conditional probability of a sequence of labels $\mathbf{y} = y_{\{t=1...T\}}$ given a sequence of observed output $\mathbf{x} = x_{\{t=1...T\}}$ and weight vector $\boldsymbol{\theta} = \theta_{\{k=1...K\}}$ is given as follows:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} exp(\sum_{t=1}^{T} \sum_{k} \theta_k f_k(y_{t-1}, y_t, \mathbf{x}, t))$$

[3] A similar model which conditions on whether productions have previously occurred *within the same sentence*, however, improves F1 to 73.6%.

where Z is the partition function. The feature functions $f_k(y_{t-1}, y_t, \mathbf{x}, t)$ can depend on two neighbouring parses, the sentences in the sequence, and the position of the sentence in the sequence. Since our feature functions do not depend on the words or the time-step within the sequence, however, we will write $f_k(y_{t-1}, y_t)$ from now on.

We treat each document in the corpus as one CRF sequence, and each sentence as one time-step in the sequence. The label sequence then is the sequence of parses, and the outputs are the sentences in the document. Since there is a large number of parses possible for each sentence and correspondingly many possible states for each label variable, we restrict the possible label state-space by extracting the N-best parses from a generative parser, and rerank over the sequences of candidate parses thus provided. We use the generative parser of Petrov and Klein (2007), a state-splitting parser that uses an EM algorithm to find splits in the nonterminal symbols to maximize training data likelihood. We use the 20-best parses, with an oracle F1 of 94.96% on section 23.

To learn the weight vector, we employ a stochastic gradient ascent method on the conditional log likelihood, which has been shown to perform well for parsing tasks (Finkel et al., 2008). In standard gradient ascent, the conditional log likelihood with a L2 regularization term for a Gaussian prior for a training corpus of N sequences is

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{t,k} \theta_k f_k(y_{t-1}^{(i)}, y_t^{(i)}) - \sum_{i=1}^{N} log\ Z^{(i)} - \sum_k \frac{\theta_k^2}{2\sigma^2}$$

And the partial derivatives with respect to the weights are

$$\frac{\partial L}{\partial \theta_k} = \sum_{i=1}^{N} \sum_t f_k(y_{t-1}^{(i)}, y_t^{(i)}) - \sum_{i=1}^{N} \sum_t \sum_{y,y'} f_k(y, y') P(y, y' | \mathbf{x}^{(i)}) - \sum_k \frac{\theta_k}{\sigma^2}$$

The first term is the feature counts in the training data, and the second term is the feature expectations according to the current weight vector. The third term corresponds to the penalty to non-zero weight values imposed by regularization. The probabilities in the second term can be efficiently calculated by the CRF-version of the forward-backward algorithm.

In standard gradient ascent, we update the weight vector after iterating through the whole training corpus. Because this is computationally expensive, we instead use stochastic gradient ascent, which approximates the true gradient by the gradient calculated from a single sample from the training corpus. We thus do not have to sum over the training set in the above expressions. We also employ a learning rate multiplier on the gradient. Thus, the weight update for the ith encountered training sequence during training is

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha_i \nabla L_{stochastic}(\boldsymbol{\theta})$$
$$\alpha_i = \eta \frac{\tau \times N}{\tau \times N + i}$$

The learning rate function is modelled on the one used by Finkel et al. (2008). It is designed such that α_i is halved after τ passes through the training set.

We train the model by iterating through the training set in a randomly permuted order, updating the weight vector after each sequence. The parameters η, τ, and σ are tuned to the development set. The final settings we use are $\eta = 0.08$, $\tau = 5$, and $\sigma = 50$. We use sections 2–21 of the Penn Treebank WSJ for training, 22 for development, and 23 for testing. We conduct 20-fold cross validation to generate the N-best parses for the training set, as is standard for N-best rerankers.

To rerank, we do inference with the linear-chain CRF for the most likely sequence of parses using the Viterbi algorithm.

3.2 Feature Functions

We experiment with various feature functions that depend on the syntactic and lexical parallelism between y_{t-1} and y_t. We use the occurrence of a rule in y_t that occurred in y_{t-1} as a feature. Based on the results of the corpus analysis, the first representation

(1) (S (NP (DT NN)) (VP (VBD)))

(2) (S (NP (NNS)) (VP (VBD)))

`Phrasal` features:
Template: (parent, $child_L$, $child_R$, repeated)
(S, edge, NP, +), (S, NP, VP, +), (S, VP, edge, +), (NP, edge, NNS, −), (NP, NNS, edge, −), (VP, edge, VBD, +), (VP, VBD, edge, +)

`Lexical` features:
Template: (parent, POS_L, POS_R, repeated)
(S, edge, NNS, −), (S, NNS, VBD, −), (S, VBD, edge, +), (NP, edge, NNS, −), (NP, NNS, edge, −), (VP, edge, VBD, +), (VP, VBD, edge, +)

Figure 3: Example of features extracted from a parse sequence specified down to the POS level.

Method	F1 (%)
Model-averaged	90.47
Combined, jointly trained −Context	90.33
Combined, jointly trained	90.31
Model-averaged −Context	90.22
`lexical` −Context	90.21
`lexical`	90.20
`phrasal`	90.12
`phrasal` −Context	89.74
Generative	89.70

Table 4: Development set (section 22) results of various models that we trained. Italicized are the models we use for the test set.

we tried was to simply enumerate the (non-lexical) productions in y_t along with whether that production is found in y_{t-1}. However, we found that our most successful feature function is to consider overlaps in partial structures of productions.

Specifically, we decompose a tree into all of the nonlexical vertically and horizontally markovized subtrees. Each of the subtrees in y_t marked by whether that same subtree occurs in the previous tree is a feature. The simple production representation corresponds to a vertical markovization of 1 and a horizontal markovization of infinite. We found that a vertical markovization of 1 and a horizontal markovization of 2 produced the best results on our data. We will call this model the `phrasal` model.

This schema so far only considers local substructures of parse trees, without being informed by the lexical information found in the leaves of the tree. We try another schema which considers the POS tag sequences found in each subtree. A feature then is the node label of the root of the subtree with the POS tag sequence it dominates, again decomposed into sequences of length 2 by markovization. We will call this model the `lexical` model.

To extract features from this sequence, we consider the substructures in the second parse, and mark whether they are found in the first parse as well. We add edge markers to mark the beginning and end of constituents. See Figure 3 for an example of features extracted by the two models.

We will consider various ways of combining the two schemata above in the next section. In addition, we also add a feature corresponding to the scaled log probability of a parse tree derived from the generative parsing baseline. Scaling is necessary because of the large differences in the magnitude of the log probability for different sentences. The scaling formula that we found to work best is to scale the maximum log probability among the N-best candidate parses to be 1.0 and the minimum to be 0.0.

3.3 Results

We train the two models which make use of extrasentential context described in the previous section, and use the model to parse the development and test set. We also trained a model which combines both sets of features, but we found that we get better performance by training the two models separately, then averaging the models by computing the respective averages of their features' weights. Thus, we use the model-averaged version of the models that consider context in the test set experiments. The generative parser forms the first baseline method to which we compare our results. We also train a reranker which makes use of the same features as we described above, but without marking whether each substructure occurs in the previous sentence. This is thus a reranking method which does not make use of the previous context. Again, we tried model averaging, but this produces less accurate parses on the

	LP	LR	F1	Exact	$\overline{CB}$	0CB	LP	LR	F1	Exact	$\overline{CB}$	0CB
	development set – length ≤ 40						development set – all sentences					
Generative	90.33	90.20	90.27	39.92	0.68	71.99	89.64	89.75	89.70	37.76	0.82	68.65
$+$*Context*	**91.25**	90.71	**90.98**	**41.25**	**0.61**	**73.45**	**90.62**	90.33	**90.47**	**38.88**	**0.74**	**70.47**
$-$*Context*	90.85	**90.78**	90.82	40.62	0.62	73.00	90.28	**90.38**	90.22	38.24	**0.74**	70.00

Table 5: Parsing results on the development set (section 22) of the Penn Treebank WSJ (%, except for $\overline{CB}$). *Generative* is the generative baseline of Petrov and Klein (2007), $+$*Context* is the best performing reranking model using previous context (model-averaged `phrasal` and `lexical`), $-$*Context* is the best performing reranking model not using previous context (jointly trained `phrasal` and `lexical`).

	LP	LR	F1	Exact	$\overline{CB}$	0CB	LP	LR	F1	Exact	$\overline{CB}$	0CB
	test set – length ≤ 40						test set – all sentences					
Generative	90.04	89.84	89.94	38.31	0.80	68.33	89.60	89.35	89.47	36.05	0.94	65.81
$+$*Context*	90.63	90.11	90.37	**39.02**	0.73	69.40	90.17	89.64	89.91	**36.84**	0.87	67.09
$-$*Context*	**90.64**	**90.43**	**90.54**	38.62	**0.72**	**69.84**	**90.20**	**89.97**	**90.08**	36.47	**0.85**	**67.55**

Table 6: Parsing results on the test set (section 23) of the Penn Treebank WSJ (%, except for $\overline{CB}$)

development set, so we use the jointly trained model on the test set. We will refer to this model as the context-ignorant or $-$Context model, as opposed to the previous context-aware or $+$Context model. The results of these experiments on the development set are shown in Table 4.

PARSEVAL results[4] on the development and test set are presented in Tables 5 and 6. We see that the reranked models outperform the generative baseline model in terms of F1, and that the reranked model that uses extra-sentential context outperforms the version that does not use extra-sentential context in the development set, but not in the test set. Using Bikel's randomized parsing evaluation comparator[5], we find that both reranking models outperform the baseline generative model to statistical significance for recall and precision. The context-ignorant reranker outperforms the context-aware reranker on recall ($p < 0.01$), but not on precision ($p = 0.42$). However, the context-aware model has the highest exact match scores in both the development and the test set.

The F1 result suggests two possibilities–either the context-aware model captures the same information as the context-ignorant model, but less effectively, or the two models capture different information about the parses. Two pieces of evidence point to the latter possibility. First, if the context-aware model were truly inferior, then we would expect it to outperform the context-ignorant model on almost no sentences. Otherwise, we would expect them to do well on different sentences. Table 7 shows that the context-aware model outperforms the context-ignorant model on nearly as many trees in the test section as the reverse. Second, if we hypothetically had an oracle that could determine whether the context-ignorant or the context-aware model would be more accurate on a sentence and if the two models were complementary to each other, we would expect to achieve a gain in F1 over the generative baseline which is roughly the sum of the gain achieved by each model separately. This is indeed the case, as we are able to achieve F1s of 91.23% and 90.89% on sections 22 and 23 respectively, roughly twice the improvement that the individual models obtain.

Sec.	$-$*Context better*	*same*	$+$*Context better*
22	157	1357	186
23	258	1904	254

Table 7: Context-aware vs. context-ignorant reranking results, by sentential F1.

To put our results in perspective, we now compare the magnitude of the improvement in F1 our context-

[4]This evaluation ignores punctuation and corresponds to the `new.prm` parameter setting on evalb.

[5]`http://www.cis.upenn.edu/~dbikel/software.html`

System	*Baseline*	*Best*	*Imp. (rel.)*
Dubey et al. (2006)	73.3	73.6	0.3 (1.1%)
Hogan (2007)	89.4	89.6	0.2 (1.9%)
This work	**89.5**	**89.9**	**0.4 (3.8%)**

Table 8: A comparison of parsers specialized to exploit intra- or extra-sentential syntactic parallelism on section 23 in terms of the generative baseline they compare themselves against, the best F1 their non-baseline models achieve, and the absolute and relative improvements.

aware model achieves over the generative baseline to that of other systems specialized to exploit intra- or extra-sentential parallelism. We achieve a greater improvement despite the fact that our generative baseline provides a higher level of performance, and is presumably thus more difficult to improve upon (Table 8). These systems do not compare themselves against a reranked model that does not use parallelism as we do in this work.

During inference, the Viterbi algorithm recovers the most probable sequence of parses, and this means that we are relying on the generative parser to provide the context (i.e. the previous parses) when analyzing any given sentence. We do another type of oracle analysis in which we provide the parser with the correct, manually annotated parse tree of the previous sentence when extracting features for the current sentence during training and parsing. This "perfect context" model achieves F1s of 90.42% and 90.00% on sections 22 and 23 respectively, which is comparable to the best results of our reranking models. This indicates that the lack of perfect contextual information is not a major obstacle to further improving parsing performance.

3.4 Analysis

We now analyze several specific cases in the development set in which the reranker makes correct use of contextual information. They concretely illustrate how context can improve parsing performance, and confirm our initial intuition that extra-sentential context can be useful for parsing. The sentence in (3) and (4) is one such case.

(3) *Generative/Context-ignorant*: (S (S A BMA spokesman said "runaway medical costs" have made health insurance "a significant challenge) ," and (S margins also have been pinched ...) (. .))

(4) *Context-aware*: (S (NP A BMA spokesman) (VP said "runaway medical costs" have made health insurance "a significant challenge," and margins also have been pinched ...) (. .))

The baseline and the context-ignorant models parse the sentence as a conjunction of two S clauses, misanalyzing the scope of what is said by the BMA spokesman to the first part of the conjunct. By analyzing the features and feature weight values extracted from the parse sequence, we determined that the context-aware reranker is able to correct the analysis of the scoping due to a parallelism in the syntactic structure. Specifically, the substructure $S \rightarrow VP.$ is present in both this sentence and the previous sentence of the reranked sequence, which also contains a reporting verb.

(5) (S (NP BMA Corp., Kansas City, Mo.,) (VP said it's weighing "strategic alternatives" ... and is contacting possible buyers ...) (. .))

As a second example, consider the following sentence.

(6) *Generative/Context-ignorant*: To achieve maximum liquidity and minimize price volatility, (NP either all markets) (VP should be open to trading or none).

(7) *Context-aware*: To achieve maximum liquidity and minimize price volatility, (CC either) (S (NP all markets) should be open to trading or none).

The original generative and context-ignorant parses posit that "either all markets" is a noun phrase, which is incorrect. Syntactic parallelism corrects this for two reasons. First, the reranker prefers a determiner to start an NP in a consistent context, as both surround sentences also contain this substructure. Also, the previous sentence also contains a conjunction CC followed by a S node under a S node, which the reranker prefers.

While these examples show contextual features to be useful for parsing coordinations, we also found

context-awareness to be useful for other types of structural ambiguity such as PP attachment ambiguity. Notice that the method we employ to correct coordination errors is different from previous approaches which usually rely on lexical or syntactic similarity between conjuncts rather than between sentences. Our approach can thus broaden the range of sentences that can be usefully reranked. For example, there is little similarity between conjuncts to avail of in the second example (Sentences 6 and 7).

Based on these analyses, it appears that context awareness provides a source of information for parsing which is not available to context-ignorant parsers. We should thus consider integrating both types of features into the reranking parser to build on the advantages of each. Specifically, within-sentence features are most appropriate for lexical dependencies and some structural dependencies. Extra-sentential features, on the other hand, are appropriate for capturing the syntactic consistency effects as we have demonstrated in this paper.

4 Conclusions

In this paper, we have examined evidence for syntactic consistency between neighbouring sentences. First, we conducted a corpus analysis of the Penn Treebank WSJ, and shown that parallelism exists between sentences for productions with a variety of *LHS* types, generalizing previous results for noun phrase structure. Then, we explored a novel source of features for parsing informed by the extra-sentential context. We improved on the parsing accuracy over a generative baseline parser, and rival a similar reranking model that does not rely on extra-sentential context. By examining the subsets of the evaluation data on which each model performs best and also individual cases, we argue that context allows a type of structural ambiguity resolution not available to parsers which only rely on intra-sentential context.

Acknowledgments

We would like to thank the anonymous reviewers and Timothy Fowler for their comments. This work is supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

J.K. Bock. 1986. Syntactic persistence in language production. *Cognitive Psychology*, 18(3):355–387.

A. Buch and C. Pietsch. 2010. Measuring syntactic priming in dialog corpora. In *Proceedings of the Conference on Linguistic Evidence 2010: Empirical, Theoretical and Computational Perspectives*.

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd ACL*, pages 173–180. Association for Computational Linguistics.

K.W. Church. 2000. Empirical estimates of adaptation: the chance of two Noriegas is closer to $p/2$ than p^2. In *Proceedings of 18th COLING*, pages 180–186. Association for Computational Linguistics.

T. Cohn and P. Blunsom. 2005. Semantic role labelling with tree conditional random fields. In *Ninth Conference on Computational Natural Language Learning*, pages 169–172.

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

A. Dubey, P. Sturt, and F. Keller. 2005. Parallelism in coordination as an instance of syntactic priming: Evidence from corpus-based modeling. In *Proceedings of HLT/EMNLP 2005*, pages 827–834.

A. Dubey, F. Keller, and P. Sturt. 2006. Integrating syntactic priming into an incremental probabilistic parser, with an application to psycholinguistic modeling. In *Proceedings of the 21st COLING and the 44th ACL*, pages 417–424. Association for Computational Linguistics.

J.R. Finkel, A. Kleeman, and C.D. Manning. 2008. Efficient, feature-based, conditional random field parsing. *Proceedings of ACL-08: HLT*, pages 959–967.

S.T. Gries. 2005. Syntactic priming: A corpus-based approach. *Journal of Psycholinguistic Research*, 34(4):365–399.

D. Hogan. 2007. Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of 45th ACL*, volume 45, pages 680–687.

F. Jousse, R. Gilleron, I. Tellier, and M. Tommasi. 2006. Conditional random fields for XML trees. In *ECML Workshop on Mining and Learning in Graphs*.

S. Kübler, W. Maier, E. Hinrichs, and E. Klett. 2009. Parsing coordinations. In *Proceedings of the 12th EACL*, pages 406–414. Association for Computational Linguistics.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proceedings of HLT-NAACL 2006*.

S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL 2007*, pages 404–411. Association for Computational Linguistics.

M.J. Pickering and H.P. Branigan. 1999. Syntactic priming in language production. *Trends in Cognitive Sciences*, 3(4):136–141.

D. Reitter. 2008. *Context Effects in Language Production: Models of Syntactic Priming in Dialogue Corpora*. Ph.D. thesis, University of Edinburgh.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, pages 213–220.

Y. Tsuruoka, J. Tsujii, and S. Ananiadou. 2009. Fast full parsing by linear-chain conditional random fields. In *Proceedings of the 12th EACL*, pages 790–798. Association for Computational Linguistics.