Parser Evaluation over Local and Non-Local Deep Dependencies in a Large Corpus

Emily M. Bender[♠], **Dan Flickinger**[♡], **Stephan Oepen**[♣], **Yi Zhang**[♢]
[♠]Dept of Linguistics, University of Washington, [♡]CSLI, Stanford University
[♣]Dept of Informatics, Universitetet i Oslo, [♢]Dept of Computational Linguistics, Saarland University
ebender@uw.edu, danf@stanford.edu, oe@ifi.uio.no, yzhang@coli.uni-sb.de

Abstract

In order to obtain a fine-grained evaluation of parser accuracy over naturally occurring text, we study 100 examples each of ten reasonably frequent linguistic phenomena, randomly selected from a parsed version of the English Wikipedia. We construct a corresponding set of gold-standard target dependencies for these 1000 sentences, operationalize mappings to these targets from seven state-of-the-art parsers, and evaluate the parsers against this data to measure their level of success in identifying these dependencies.

1 Introduction

The terms "deep" and "shallow" are frequently used to characterize or contrast different approaches to parsing. Inevitably, such informal notions lack a clear definition, and there is little evidence of community consensus on the relevant dimension(s) of depth, let alone agreement on applicable metrics. At its core, the implied dichotomy of approaches alludes to differences in the interpretation of the parsing task. Its abstract goal, on the one hand, could be pre-processing of the linguistic signal, to enable subsequent stages of analysis. On the other hand, it could be making explicit the (complete) contribution that the grammatical form of the linguistic signal makes to interpretation, working out who did what to whom. Stereotypically, one expects corresponding differences in the choice of interface representations, ranging from various levels of syntactic analysis to logical-form representations of semantics.

In this paper, we seek to probe aspects of variation in automated linguistic analysis. We make the assumption that an integral part of many (albeit not all) applications of parsing technology is the recovery of structural relations, i.e. dependencies at the level of interpretation. We suggest a selection of ten linguistic phenomena that we believe (a) occur with reasonably high frequency in running text and (b) have the potential to shed some light on the depths of linguistic analysis. We quantify the frequency of these constructions in the English Wikipedia, then annotate 100 example sentences for each phenomenon with gold-standard dependencies reflecting core properties of the phenomena of interest. This gold standard is then used to estimate the recall of these dependencies by seven commonly used parsers, providing the basis for a qualitative discussion of the state of the art in parsing for English.

In this work, we answer the call by Rimell et al. (2009) for "construction-focused parser evaluation", extending and complementing their work in several respects: (i) we investigate both local and non-local dependencies which prove to be challenging for many existing state-of-the-art parsers; (ii) we investigate a wider range of linguistic phenomena, each accompanied with an in-depth discussion of relevant properties; and (iii) we draw our data from the 50-million sentence English Wikipedia, which is more varied and a thousand times larger than the venerable WSJ corpus, to explore a more level and ambitious playing field for parser comparison.

2 Background

All parsing systems embody knowledge about possible and probable pairings of strings and corresponding linguistic structure. Such linguistic and probabilistic knowledge can be hand-coded (e.g., as grammar rules) or automatically acquired from labeled or

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 397–408,
Edinburgh, Scotland, UK, July 27–31, 2011.

unlabeled training data. A related dimension of variation is the type of representations manipulated by the parser. We briefly review some representative examples along these dimensions, as these help to position the parsers we subsequently evaluate.[1]

2.1 Approaches to parsing

Source of linguistic knowledge At one end of this dimension, we find systems whose linguistic knowledge is encoded in hand-crafted rules and lexical entries; for English, the ParGram XLE system (Riezler et al., 2002) and DELPH-IN English Resource Grammar (ERG; Flickinger (2000))—each reflecting decades of continuous development—achieve broad coverage of open-domain running text, for example. At the other end of this dimension, we find fully unsupervised approaches (Clark, 2001; Klein and Manning, 2004), where the primary source of linguistic knowledge is co-occurrence patterns of words in unannotated text. As Haghighi and Klein (2006) show, augmenting this knowledge with hand-crafted prototype "seeds" can bring strong improvements. Somewhere between these poles, a broad class of parsers take some or all of their linguistic knowledge from annotated treebanks, e.g. the Penn Treebank (PTB), which encodes "surface grammatical analysis" (Marcus et al., 1993). Such approaches include those that directly (and exclusively) use the information in the treebank (e.g. Charniak (1997), Collins (1999), Petrov et al. (2006), inter alios) as well as those that complement treebank structures with some amount of hand-coded linguistic knowledge (e.g. O'Donovan et al. (2004), Miyao et al. (2004), Hockenmaier and Steedman (2007), inter alios). Another hybrid in terms of its acquisition of linguistic knowledge is the RASP system of Briscoe et al. (2006), combining a hand-coded grammar over PoS tag sequences with a probabilistic tagger and statistical syntactic disambiguation.

Design of representations Approaches to parsing also differ fundamentally in the style of representation assigned to strings. These vary both in their formal nature and the "granularity" of linguistic information (i.e. the number of distinctions assumed), encompassing variants of constituent structure, syntactic dependencies, or logical-form representations of semantics. Parser interface representations range between the relatively simple (e.g. phrase structure trees with a limited vocabulary of node labels as in the PTB, or syntactic dependency structures with a limited vocabulary of relation labels as in Johansson and Nugues (2007)) and the relatively complex, as for example elaborate syntactico-semantic analyses produced by the ParGram or DELPH-IN grammars.

There tends to be a correlation between the methodology used in the acquisition of linguistic knowledge and the complexity of representations: in the creation of a mostly hand-crafted treebank like the PTB, representations have to be simple enough for human annotators to reliably manipulate. Deriving more complex representations typically presupposes further computational support, often involving some hand-crafted linguistic knowledge—which can take the form of mappings from PTB-like representations to "richer" grammatical frameworks (as in the line of work by O'Donovan et al. (2004), and others; see above), or can be rules for creating the parse structures in the first place (i.e. a computational grammar), as for example in the treebanks of van der Beek et al. (2002) or Oepen et al. (2004).[2]

In principle, one might expect that richer representations allow parsers to capture complex syntactic or semantic dependencies more explicitly. At the same time, such "deeper" relations may still be recoverable (to some degree) from comparatively simple parser outputs, as demonstrated for unbounded dependency extraction from strictly local syntactic dependency trees by Nivre et al. (2010).

2.2 An armada of parsers

Stanford Parser (Klein and Manning, 2003) is a probabilistic parser which can produce both phrase structure trees and grammatical relations (syntactic dependencies). The parsing model we evaluate is the

[1] Additional sources of variation among extant parsing technologies include (a) the behavior with respect to ungrammatical inputs and (b) the relationship between probabilistic and symbolic knowledge in the parser, where parsers with a hand-coded grammar at their core typically also incorporate an automatically trained probabilistic disambiguation component.

[2] A noteworthy exception to this correlation is the annotated corpus of Zettlemoyer and Collins (2005), which pairs surface strings from the realm of natural language database interfaces directly with semantic representations in lambda calculus. These were hand-written on the basis of database query statements distributed with the original datasets.

English factored model which combines the preferences of unlexicalized PCFG phrase structures and of lexical dependencies, trained on sections 02–21 of the WSJ portion of the PTB. We chose Stanford Parser from among the state-of-the-art PTB-derived parsers for its support for grammatical relations as an alternate interface representation.

Charniak&Johnson Reranking Parser (Charniak and Johnson, 2005) is a two-stage PCFG parser with a lexicalized generative model for the first-stage, and a discriminative MaxEnt reranker for the second-stage. The models we evaluate are also trained on sections 02–21 of the WSJ. Top-50 readings were used for the reranking stage. The output constituent trees were then converted into Stanford Dependencies. According to Cer et al. (2010), this combination gives the best parsing accuracy in terms of Stanford dependencies on the PTB.

Enju (Miyao et al., 2004) is a probabilistic HPSG parser, combining a hand-crafted core grammar with automatically acquired lexical types from the PTB.[3] The model we evaluate is trained on the same material from the WSJ sections of the PTB, but the treebank is first semi-automatically converted into HPSG derivations, and the annotation is enriched with typed feature structures for each constituent. In addition to HPSG derivation trees, Enju also produces predicate argument structures.

C&C (Clark and Curran, 2007) is a statistical CCG parser. Abstractly similar to the approach of Enju, the grammar and lexicon are automatically induced from CCGBank (Hockenmaier and Steedman, 2007), a largely automatic projection of (the WSJ portion of) PTB trees into the CCG framework. In addition to CCG derivations, the C&C parser can directly output a variant of grammatical relations.

RASP (Briscoe et al., 2006) is an unlexicalized robust parsing system, with a hand-crafted "tag sequence" grammar at its core. The parser thus analyses a lattice of PoS tags, building a parse forest from which the most probable syntactic trees and sets of corresponding grammatical relations can be extracted. Unlike other parsers in our mix, RASP did not build on PTB data in either its PoS tagging or syntactic disambiguation components.

MSTParser (McDonald et al., 2005) is a data-driven dependency parser. The parser uses an edge-factored model and searches for a maximal spanning tree that connects all the words in a sentence into a dependency tree. The model we evaluate is the second-order projective model trained on the same WSJ corpus, where the original PTB phrase structure annotations were first converted into dependencies, as established in the CoNLL shared task 2009 (Johansson and Nugues, 2007).

XLE/ParGram (Riezler et al., 2002, see also Cahill et al., 2008) applies a hand-built Lexical Functional Grammar for English and a stochastic parse selection model. For our evaluation, we used the Nov 4, 2010 release of XLE and the Nov 25, 2009 release of the ParGram English grammar, with c-structure pruning turned off and resource limitations set to the maximum possible to allow for exhaustive search. In particular, we are evaluating the f-structures output by the system.

Each parser, of course, has its own requirements regarding preprocessing of text, especially tokenization. We customized the tokenization to each parser, by using the parser's own internal tokenization or pre-tokenizing to match the parser's desired input. The evaluation script is robust to variations in tokenization across parsers.

3 Phenomena

In this section we summarize the ten phenomena we explore and our motivations for choosing them. Our goal was to find phenomena where the relevant dependencies are relatively subtle, such that more linguistic knowledge is beneficial in order to retrieve them. Though this set is of course only a sampling, these phenomena illustrate the richness of structure, both local and non-local, involved in the mapping from English strings to their meanings. We discuss the phenomena in four sets and then briefly review their representation in the Penn Treebank.

3.1 Long distance dependencies

Three of our phenomena can be classified as involving long-distance dependencies: finite *that*-less relatives clauses ('barerel'), tough adjectives ('tough') and right node raising ('rnr'). These are illustrated

[3] This hand-crafted grammar is distinct from the ERG, despite sharing the general framework of HPSG. The ERG is not included in our evaluation, since it was used in the extraction of the original examples and thus cannot be fairly evaluated.

in the following examples:[4]

(1) barerel: This is the second *time* in a row Australia **has lost** their home tri-nations' series.

(2) tough: Original *copies* are very hard to **find**.

(3) rnr: Ilúvatar, as his names imply, exists **before** and **independently of** *all else*.

While the majority of our phenomena involve local dependencies, we include these long-distance dependency types because they are challenging for parsers and enable more direct comparison with the work of Rimell et al. (2009), who also address right node raising and bare relatives. Our barerel category corresponds to their "object reduced relative" category with the difference that we also include adverb relatives, where the head noun functions as a modifier within the relative clause, as does *time* in (1). In contrast, our rnr category is somewhat narrower than Rimell et al. (2009)'s "right node raising" category: where they include raised modifiers, we restrict our category to raised complements.

Part of the difficulty in retrieving long-distance dependencies is that the so-called extraction site is not overtly marked in the string. In addition to this baseline level of complication, these three construction types present further difficulties: Bare relatives, unlike other relative clauses, do not carry any lexical cues to their presence (i.e., no relative pronouns). *Tough* adjective constructions require the presence of specific lexical items which form a subset of a larger open class. They are rendered more difficult by two sources of ambiguity: alternative subcategorization frames for the adjectives and the purposive adjunct analysis (akin to *in order to*) for the infinitival VP. Finally, right node raising often involves coordination where one of the conjuncts is in fact not a well-formed phrase (e.g., *independently of* in (3)), making it potentially difficult to construct the correct coordination structure, let alone associate the raised element with the correct position in each conjunct.

3.2 Non-dependencies

Two of our phenomena crucially look for the lack of dependencies. These are *it* expletives ('itexpl') and verb-particle constructions ('vpart'):

(4) itexpl: Crew negligence is blamed, and *it* **is suggested** that the flight crew were drunk.

(5) vpart: He once **threw out** two *baserunners* at home in the same inning.

The English pronoun *it* can be used as an ordinary personal pronoun or as an expletive: a placeholder for when the language demands a subject (or occasionally object) NP but there is no semantic role for that NP. The expletive *it* only appears when it is licensed by a specific construction (such as extraposition, (4)) or selecting head. If the goal of parsing is to recover from the surface string the dependencies capturing who did what to whom, expletive *it* should not feature in any of those dependencies. Likewise, instances of expletive *it* should be detected and discarded in reference resolution. We hypothesize that detecting expletive *it* requires encoding linguistic knowledge about its licensers.

The other non-dependency we explore is between the particle in verb-particle constructions and the direct object. Since English particles are almost always homophonous with prepositions, when the object of the verb-particle pair follows the particle, there will always be a competing analysis which analyses the sequence as V+PP rather than V+particle+NP. Furthermore, since verb-particle pairs often have non-compositional semantics (Sag et al., 2002), misanalyzing these constructions could be costly to downstream components.

3.3 Phrasal modifiers

Our next category concerns modifier phrases:

(6) ned: *Light* ***colored*** **glazes** also have softening effects when painted over dark or bright images.

(7) absol: The format **consisted** of 12 games, each *team* ***facing*** the other teams twice.

The first, ('ned'), is a pattern which to our knowledge has not been named in the literature, where a noun takes the typically verbal *-ed* ending, is modified by another noun or adjective, and functions as a modifier or a predicate. We believe this phenomenon to be interesting because its unusual morphology is likely to lead PoS-taggers astray, and because the often-hyphenated Adj+N-ed constituent has productive internal structure constraining its interpretation.

The second phrasal modifier we investigate is the absolutive construction. An absolutive consists of an

[4] All examples are from our data. Words involved in the relevant dependencies are highlighted in italics (dependents) and boldface (heads).

NP followed by a non-finite predicate (such as could appear after the copula *be*). The whole phrase modifies a verbal projection that it attaches to. Absolutives may be marked with *with* or unmarked. Here, we focus on the unmarked type as this lack of lexical cue can make the construction harder to detect.

3.4 Subtle arguments

Our final three phenomena involve ways in which verbal arguments can be more difficult to identify than in ordinary finite clauses. These include detecting the arguments of verbal gerunds ('vger'), the interleaving of arguments and adjuncts ('argadj') and raising/control ('control') constructions.

(8) vger: ***Accessing*** *the website* without the "www" subdomain **returned** a copy of the main site for "EP.net".

(9) argadj: The story **shows**, *through* flashbacks, the different *histories* of the characters.

(10) control: *Alfred* "retired" in 1957 at age 60 but **continued** to **paint** full time.

In a verbal gerund, the *-ing* form a verb retains verbal properties (e.g., being able to take NP complements, rather than only PP complements) but heads a phrase that fills an NP position in the syntax (Malouf, 2000). Since gerunds have the same morphology as present participle VPs, their role in the larger clause is susceptible to misanalysis.

The argadj examples are of interest because English typically prefers to have direct objects directly adjacent to the selecting verb. Nonetheless, phenomena such as parentheticals and heavy-NP shift (Arnold et al., 2000), in which "heavy" constituents appear further to the right in the string, allow for adjunct-argument order in a minority of cases. We hypothesize that the relative infrequency of this construction will lead parsers to prefer incorrect analyses (wherein the adjunct is picked up as a complement, the complement as an adjunct or the structure differs entirely) unless they have access to linguistic knowledge providing constraints on possible and probable complementation patterns for the head.

Finally, we turn to raising and control verbs ('control') (e.g., Huddleston and Pullum (2002, ch. 14)). These verbs select for an infinitival VP complement and stipulate that another of their arguments (subject or direct object in the examples we explore) is identified with the unrealized subject position of the infinitival VP. Here it is the dependency between the infinitval VP and the NP argument of the "upstairs" verb which we expect to be particularly subtle. Getting this right requires specific lexical knowledge about which verbs take these complementation patterns. This lexical knowledge needs to be represented in such a way that it can be used robustly even in the case of passives, relative clauses, etc.[5]

3.5 Penn Treebank representations

We investigated the representation of these 10 phenomena in the PTB (Marcus et al., 1993) in two steps: First we explored the PTB's annotation guidelines (Bies et al., 1995) to determine how the relevant dependencies were intended to be represented. We then used Ghodke and Bird's (2010) Treebank Search to find examples of the intended annotations as well as potential examples of the phenomena annotated differently, to get a sense of the consistency of the annotation from both precision and recall perspectives. In this study, we take the phrase structure trees of the PTB to represent dependencies based on reasonable identification of heads.

The barerel, vpart, and absol phenomena are completely unproblematic, with their relevant dependencies explicitly and reliably represented. In addition, the tough construction is reliably annotated, though one of the dependencies we take to be central is not directly represented: The missing argument is linked to a null *wh* head at the left edge of the complement of the tough predicate, rather than to its subject. Two further phenomena (rnr and vger) are essentially correctly represented: the representations of the dependencies are explicit and mostly but not entirely consistently applied. Two out of a sample of 20 examples annotated as containing rnr did not, and two out of a sample of 35 non-rnr-annotated coordinations actually contained rnr. For vger the primary problem is with the PoS tagging, where the gerund is sometimes given a nominal tag, contrary to PTB guidelines, though the structure above it conforms.

The remaining four constructions are more problematic. In the case of object control, while the guide-

[5] Distinguishing between raising and control requires further lexical knowledge and is another example of a "non-dependency" (in the raising examples). We do not draw that distinction in our annotations.

lines specify an analysis in which the shared NP is attached as the object of the higher verb, the PTB includes not only structures conforming to that analysis but also "small clause" structures, with the latter obscuring the relationship of the shared argument to the higher verb. In the case of itexpl, the adjoined `(S(-NONE- *EXP*))` indicating an expletive use of *it* is applied consistently for extraposition (as prescribed in the guidelines). However, the set of lexical licensers of the expletive is incomplete. For argadj we run into the problem that the PTB does not explicitly distinguish between post-verbal modifiers and verbal complements in the way that they are attached. The guidelines suggest that the function tags (e.g., `PP-LOC`, etc.) should allow one to distinguish these, but examination of the PTB itself suggests that they are not consistently applied. Finally, the ned construction is not mentioned in the PTB guidelines nor is its internal structure represented in the treebank. Rather, strings like *gritty-eyed* are left unsegmented and tagged as `JJ`.

We note that the PTB representations of many of these phenomena (barerel, tough, rnr, argadj, control, itexpl) involve empty elements and/or function tags. Systems that strip these out before training, as is common practice, will not benefit from the information that is in the PTB.

Our purpose here is not to criticize the PTB, which has been a tremendously important resource to the field. Rather, we have two aims: The first is to provide context for the evaluation of PTB-derived parsers on these phenomena. The second is to highlight the difficulty of producing consistent annotations of any complexity as well as the hurdles faced by a hand-annotation approach when attempting to scale a resource to more complex representations and/or additional phenomena (though cf. Vadas and Curran (2008) on improving PTB representations).

4 Methodology

4.1 Data extraction

We processed 900 million tokens of Wikipedia text using the October 2010 release of the ERG, following the work of the WikiWoods project (Flickinger et al., 2010). Using the top-ranked ERG derivation trees as annotations over this corpus and simple patterns using names of ERG-specific constructions or lexical types, we randomly selected a set of candidate sentences for each of our ten phenomena. These candidates were then hand-vetted in sequence by two annotators to identify, for each phenomenon, 100 examples that do in fact involve the phenomenon in question and which are both grammatical and free of typos. Examples that were either deemed overly basic (e.g. plain V + V coordination, which the ERG treats as rnr) or inappropriately complex (e.g. non-constituent coordination obscuring the interleaving of arguments and adjuncts) were also discarded at this step. Table 1 summarizes relative frequencies of each phenomenon in about 47 million parsed Wikipedia sentences, as well as the total size of the candidate sets inspected. For the control and tough phenomena hardly any filtering for complexity was applied, hence these can serve as indicators of the rate of genuine false positives. For phenomena that partially overlap with those of Rimell et al. (2009), it appears our frequency estimates are comparable to what they report for the Brown Corpus (but not the WSJ portion of the PTB).

Phenomenon	Frequency	Candidates
barerel	2.12%	546
tough	0.07%	175
rnr	0.69%	1263
itexpl	0.13%	402
vpart	4.07%	765
ned	1.18%	349
absol	0.51%	963
vger	5.16%	679
argadj	3.60%	1346
control	3.78%	124

Table 1: Relative frequencies of phenomena matches in Wikipedia, and number of candidate strings vetted.

4.2 Annotation format

We annotated up to two dependency triples per phenomenon instance, identifying the heads and dependents by the surface form of the head words in the sentence suffixed with a number indicating word position (see Table 2).[6] Some strings contain more than one instance of the phenomenon they illustrate; in these cases, multiple sets of dependencies are

[6] As the parsers differ in tokenization strategies, our evaluation script treats these position IDs as approximate indicators.

Item ID	Phenomenon	Polarity	Dependency
1011079100200	absol	1	having-2|been-3|passed-4 ARG act-1
1011079100200	absol	1	withdrew-9 MOD having-2|been-3|passed-4
1011079100200	absol	1	carried+on-12 MOD having-2|been-3|passed-4

Table 2: Sample annotations for sentence # 1011079100200: *The-0 act-1 having-2 been-3 passed-4 in-5 that-6 year-7 Jessop-8 withdrew-9 and-10 Whitworth-11 carried-12 on-13 with-14 the-15 assistance-16 of-17 his-18 son-19.*

Phenomenon	Head	Type	Dependent	Distance
Bare relatives	gapped predicate in relative	ARG2/MOD	modified noun	3.0 (8)
(barerel)	modified noun	MOD	top predicate of relative	3.3 (8)
Tough adjectives	*tough* adjective	ARG2	*to*-VP complement	1.7 (5)
(tough)	gapped predicate in *to*-VP	ARG2	subject/modifiee of adjective	6.4 (21)
Right Node Raising	verb/prep2	ARG2	shared noun	2.8 (9)
(rnr)	verb/prep1	ARG2	shared noun	6.1 (12)
Expletive It	*it*-subject taking verb	!ARG1	*it*	1.2 (3)
(itexpl)	raising-to-object verb	!ARG2	*it*	–
Verb+particle constructions	particle	!ARG2	complement	2.7 (9)
(vpart)	verb+particle	ARG2	complement	3.7 (10)
Adj/Noun2 + Noun1-*ed*	head noun	MOD	Noun1-*ed*	2.4 (17)
(ned)	Noun1-*ed*	ARG1/MOD	Adj/Noun2	1.0 (1.5)
Absolutives	absolutive predicate	ARG1	subject of absolutive	1.7 (12)
(absol)	main clause predicate	MOD	absolutive predicate	9.8 (26)
Verbal gerunds	selecting head	ARG[1,2]	gerund	1.9 (13)
(vger)	gerund	ARG2/MOD	first complement/modifier of gerund	2.3 (8)
Interleaved arg/adj	selecting verb	MOD	interleaved adjunct	1.2 (7)
(argadj)	selecting verb	ARG[2,3]	displaced complement	5.9 (26)
Control	"upstairs" verb	ARG[2,3]	"downstairs" verb	2.4 (23)
(control)	"downstairs" verb	ARG1	shared argument	4.8 (17)

Table 3: Dependencies labeled for each phenomenon type, including average and maximum surface distances.

recorded. In addition, some strings evince more than one of the phenomena we are studying. However, we only annotate the dependencies associated with the phenomenon the string was selected to represent. Finally, in examples with coordinated heads or dependents, we recorded separate dependencies for each conjunct. In total, we annotated 2127 dependency triples for the 1000 sentences, including 253 negative dependencies (see below). Table 3 outlines the dependencies annotated for each phenomenon.

To allow for multiple plausible attachment sites, we give disjunctive values for heads or dependents in several cases: (i) with auxiliaries, (ii) with complementizers (*that* or *to*, as in Table 2), (iii) in cases of measure or classifier nouns or partitives, (iv) with multi-word proper names and (v) where there is genuine attachment ambiguity for modifiers. As these sets of targets are disjunctive, these conventions should have the effect of increasing measured parser performance. 580 (27%) of the annotated dependencies had at least one disjunction.

4.3 Annotation and reconciliation process

The entire data set was annotated independently by two annotators. Both annotators were familiar with the ERG, used to identify these sentences in the WikiWoods corpus, but the annotation was done without reference to the ERG parses. Before beginning annotation on each phenomenon, we agreed on which dependencies to annotate. We also communicated with each other about annotation conventions as the need for each convention became clear. The annotation conventions address how to handle coordination, semantically empty auxiliaries, passives and similar orthogonal phenomena.

Once the entire data set was dual-annotated, we compared annotations, identifying the following sources of mismatch: typographical errors, incompletely specified annotation conventions, inconsistent application of conventions (101 items, dropping in frequency as the annotation proceeded), and genuine disagreement about what to annotate, either different numbers of dependencies of interest identified

in an item (59 items) or conflicting elements in a dependency (54 items).[7] Overall, our initial annotation pass led to agreement on 79% of the items, and a higher per-dependency level of agreement. Agreement could be expected to approach 90% with more experience in applying annotation conventions.

We then reconciled the annotations, using the comparison to address all sources of difference. In most cases, we readily agreed which annotation was correct and which was in error. In a few cases, we decided that both annotations were plausible alternatives (e.g., in terms of alternative attachment sites for modifiers) and so created a single merged annotation expressing the disjunction of both (cf. § 4.2).

5 Evaluation

With the test data consisting of 100 items for each of our ten selected phenomena, we ran all seven parsing systems and recorded their dependency-style outputs for each sentence. While these outputs are not directly comparable with each other, we were able to associate our manually-annotated target dependencies with parser-specific dependencies, by defining sets of phenomenon-specific regular expressions for each parser. In principle, we allow this mapping to be somewhat complex (and forgiving to non-contentful variation), though we require that it work deterministically and not involve specific lexical information. An example set is given in Fig. 2.

```
"absol" =>
{'ARG1' => [
'\(ncsubj \W*{W1}\W*_(\d+) \W*{W2}\W*_(\d+) _\)',
'\(ncmod _ \W*{W2}\W*_(\d+) \W*{W1}\W*_(\d+)\)'],
'ARG' => [
'\(ncsubj \W*{W1}\W*_(\d+) \W*{W2}\W*_(\d+) _\)',
'\(ncmod _ \W*{W1}\W*_(\d+) \W*{W2}\W*_(\d+)\)'],
'MOD' => [
'\(xmod _ \W*{W1}\W*_(\d+) \W*{W2}\W*_(\d+)\)',
'\(ncmod _ \W*{W1}\W*_(\d+) \W*{W2}\W*_(\d+)\)',
'\(cmod _ \W*{W1}\W*_(\d+) \W*{W2}\W*_(\d+)\)']}
```

Figure 2: Regexp set to evaluate C&C for absol.

These expressions fit the output that we got from the C&C parser, illustrated in Fig. 3 with a relevant portion of the dependencies produced for the example in Table 2. Here the C&C dependency `(ncsubj passed_4 Act_1 _)` matches the first target in the gold-standard (Table 2), but no matching C&C dependency is found for the other two targets.

```
(xmod _ Act_1 passed_4)
(ncsubj passed_4 Act_1 _)
(ncmod _ withdrew,_9 Jessop_8)
(dobj year,_7 withdrew,_9)
```

Figure 3: Excerpts of C&C output for item in Table 2.

The regular expressions operate solely on the dependency labels and are not lexically-specific. They are specific to each phenomenon, as we did not attempt to write a general dependency converter, but rather to discover what patterns of dependency relations describe the phenomenon when it is correctly identified by each parser. Thus, though we did not hold out a test set, we believe that they would generalize to additional gold standard material annotated in the same way for the same phenomena.[8]

In total, we wrote 364 regular expressions to handle the output of the seven parsers, allowing some leeway in the role labels used by a parser for any given target dependency. The supplementary materials for this paper include the test data, parser outputs, target annotations, and evaluation script.

Fig. 1 provides a visualization of the results of our evaluation. Each column of points represents one dependency type. Dependency types for the same phenomenon are represented by adjacent columns. The order of the columns within a phenomenon follows the order of the dependency descriptions in Table 3: For each pair, the dependency type with the higher score for the majority of the parsers is shown first (to the left). The phenomena themselves are also arranged according to increasing (average) difficulty. itexpl only has one column, as we annotated just one dependency per instance here. (The two descriptions in Table 3 reflect different, mutually-incompatible instance types.) Since expletive *it* should not be the semantic dependent of any head, the targets are generalized for this phenomenon and the evaluation script counts as incor-

[7] We do not count typographical errors or incompletely specified conventions as failures of inter-annotator agreement.

[8] In the case of the XLE, our simplistic regular-expression approach to the interpretation of parser outputs calls for much more complex patterns than for the other parsers. This is owed to the rich internal structure of LFG f-structures and higher granularity of linguistic analysis, where feature annotations on nodes as well as reentrancies need to be taken into account. Therefore, our current results for the XLE admit small amounts of both over- and under-counting.

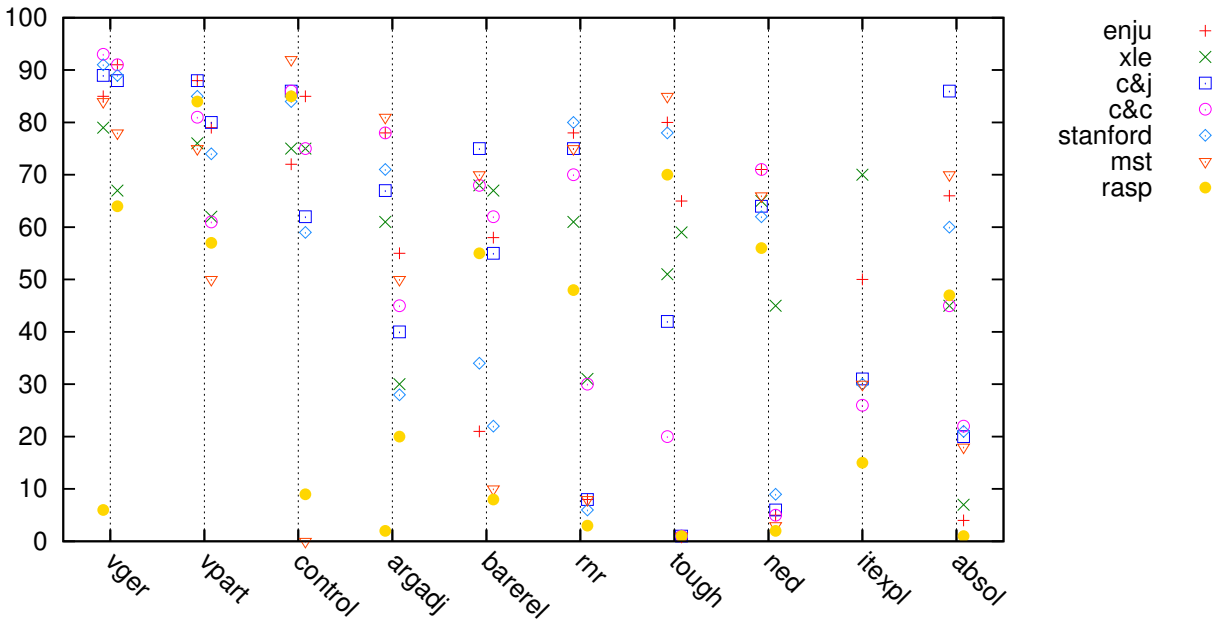

Figure 1: Individual dependency recall for seven parsers over ten phenomena.

rect any dependency involving referential *it*.

We observe fairly high recall of the dependencies for vpart and vger (with the exception of RASP), and high recall for both dependencies representing control for five systems. While Enju, Stanford, MST, and RASP all found between 70 and 85% of the dependency between the adjective and its complement in the tough construction, only Enju and XLE represented the dependency between the subject of the adjective and the gap inside the adjective's complement. For the remaining phenomena, each parser performed markedly worse on one dependency type, compared to the other. The only exceptions here are XLE and C&C's (and to a lesser extent, C&J's) scores for barerel. No system scored higher than 33% on the harder of the two dependencies in rnror absol, and Stanford, MST, and RASP all scored below 25% on the harder dependency in barerel. Only XLE scored higher than 10% on the second dependency for ned and higher than 50% for itexpl.

6 Discussion

From the results in Fig. 1, it is clear that even the best of these parsers fail to correctly identify a large number of relevant dependencies associated with linguistic phenomena that occur with reasonable frequency in the Wikipedia. Each of the parsers attempts with some success to analyze each of these phenomena, reinforcing the claim of relevance, but they vary widely across phenomena. For the two long-distance phenomena that overlap with those studied in Rimell et al. (2009), our results are comparable.[9] Our evaluation over Wikipedia examples thus shows the same relative lack of success in recovering long-distance dependencies that they found for WSJ sentences. The systems did better on relatively well-studied phenomena including control, vger, and vpart, but had less success with the rest, even though all but two of those remaining phenomena involve syntactically local dependencies (as indicated in Table 3).

Successful identification of the dependencies in these phenomena would, we hypothesize, benefit from richer (or deeper) linguistic information when parsing, whether it is lexical (tough, control, itexpl, and vpart), or structural (rnr, absol, vger, argadj, and barerel), or somewhere in between, as with ned. In the case of treebank-trained parsers, for the information to be available, it must be consistently encoded in the treebank and attended to during training. As

[9] Other than Enju, which scores 16 points higher in the evaluation of Rimell et al., our average scores for each parser across the dependencies for these phenomena are within 12 points of those reported by Rimell et al. (2009) and Nivre et al. (2010).

noted in Sections 2.1 and 3.5, there is tension between developing sufficiently complex representations to capture linguistic phenomena and keeping an annotation scheme simple enough that it can be reliably produced by humans, in the case of hand-annotation.

7 Related Work

This paper builds on a growing body of work which goes beyond (un)labeled bracketing in parser evaluation, including Lin (1995), Carroll et al. (1998), Kaplan et al. (2004), Rimell et al. (2009), and Nivre et al. (2010). Most closely related are the latter two of the above, as we adopt their "construction-focused parser evaluation methodology".

There are several methodological differences between our work and that of Rimell et al. First, we draw our evaluation data from a much larger and more varied corpus. Second, we automate the comparison of parser output to the gold standard, and we distribute the evaluation scripts along with the annotated corpus, enhancing replicability. Third, where Rimell et al. extract evaluation targets on the basis of PTB annotations, we make use of a linguistically precise broad-coverage grammar to identify candidate examples. This allows us to include both local and non-local dependencies not represented or not reliably encoded in the PTB, enabling us to evaluate parser performance with more precision over a wider range of linguistic phenomena.

These methodological innovations bring two empirical results. The first is qualitative: Where previous work showed that overall Parseval numbers hide difficulties with long-distance dependencies, our results show that there are multiple kinds of reasonably frequent *local* dependencies which are also difficult for the current standard approaches to parsing. The second is quantitative: Where Rimell et al. found two phenomena which were virtually unanalyzed (recall below 10%) for one or two parsers each, we found eight phenomena which were virtually unanalyzed by at least one system, including two unanalyzed by five and one by six. Every system had at least one virtually unanalyzed phenomenon. Thus we have shown that the dependencies being missed by typical modern approaches to parsing are more varied and more numerous than previously thought.

8 Conclusion

We have presented a detailed construction-focused evaluation of seven parsers over 10 phenomena, with 1000 examples drawn from English Wikipedia. Gauging recall of such "deep" dependencies, in our view, can serve as a proxy for downstream processing involving semantic interpretation of parser outputs. Our annotations and automated evaluation script are provided in the supplementary materials, for full replicability. Our results demonstrate that significant opportunities remain for parser improvement, and highlight specific challenges that remain invisible in aggregate parser evaluation (e.g. Parseval or overall dependency accuracy). These results suggest that further progress will depend on training data that is both more extensive and more richly annotated than what is typically used today (seeing, for example, that a large part of more detailed PTB annotation remains ignored in much parsing work).

There are obvious reasons calling for diversity in approaches to parsing and for different trade-offs in, for example, the granularity of linguistic analysis, average accuracy, cost of computation, or ease of adaptation. Our proposal is not to substitute construction-focused evaluation on Wikipedia data for widely used aggregate metrics and reference corpora, but rather to augment such best practices in the spirit of Rimell et al. (2009) and expand the range of phenomena considered in such evaluations. Across frameworks and traditions (and in principle languages), it is of vital importance to be able to evaluate the quality of parsing (and grammar induction) algorithms in a maximally informative manner.

Acknowledgments

We are grateful to Tracy King for her assistance in setting up the XLE system and to three anonymous reviewers for helpful comments. The fourth author thanks DFKI and the DFG funded Excellence Cluster on MMCI for their support of the work. Data preparation on the scale of Wikipedia was made possible through access to large-scale HPC facilities, and we are grateful to the Scientific Computing staff at UiO and the Norwegian Metacenter for Computational Science.

References

Jennifer E. Arnold, Thomas Wasow, Anthony Losongco, and Ryan Ginstrom. 2000. Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering. *Language*, 76(1):28–55.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for treebank II style Penn treebank project. Technical report, University of Pennsylvania.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 77–80, Sydney, Australia.

Aoife Cahill, John T. Maxwell III, Paul Meurer, Christian Rohrer, and Victoria Rosén. 2008. Speeding up LFG parsing using c-structure pruning. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 33–40, Manchester, England, August. Coling 2008 Organizing Committee.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: A survey and a new proposal. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 447–454, Granada.

Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to Stanford dependencies: Trade-offs between speed and accuracy. In *7th International Conference on Language Resources and Evaluation (LREC 2010)*, Malta.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598 – 603, Providence, RI.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the 5th Conference on Natural Language Learning*, pages 105–112. Toulouse, France.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. WikiWoods. Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Valletta, Malta.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):15 – 28.

Sumukh Ghodke and Steven Bird. 2010. Fast query for large treebanks. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 267–275, Los Angeles, California, June. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2006. Prototype-driven grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 881–888, Sydney, Australia, July. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, UK.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *In Proceedings of NODALIDA 2007*, pages 105–112, Tartu, Estonia.

Ron Kaplan, Stefan Riezler, Tracy H King, John T Maxwell III, Alex Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 97–104, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*, pages 3–10, Cambridge, MA. MIT Press.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 478–485, Barcelona, Spain.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*, pages 1420–1425, Montreal, Canada.

Robert Malouf. 2000. Verbal gerunds as mixed categories in HPSG. In Robert Borsley, editor, *The Nature*

and Function of Syntactic Categories, pages 133–166. Academic Press, New York.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, Canada.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-oriented grammar development for acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pages 684–693, Hainan Island, China.

Joakim Nivre, Laura Rimell, Ryan McDonald, and Carlos Gómez Rodríguez. 2010. Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 833–841, Beijing, China.

Ruth O'Donovan, Michael Burke, Aoife Cahill, Josef Van Genabith, and Andy Way. 2004. Large-scale induction and evaluation of lexical resources from the penn-ii treebank. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 367–374, Barcelona, Spain.

Stephan Oepen, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Research on Language and Computation*, 2(4):575 – 596.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics*, Philadelphia, PA.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 813–821, Singapore. Association for Computational Linguistics.

Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions. A pain in the neck for NLP. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 189–206. Springer, Berlin, Germany.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of ACL-08: HLT*, pages 335–343, Columbus, Ohio, June. Association for Computational Linguistics.

L. van der Beek, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino Dependency Treebank. In Mariet Theune, Anton Nijholt, and Hendri Hondorp, editors, *Computational Linguistics in the Netherlands*, Amsterdam, The Netherlands. Rodopi.

Luke Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence*, pages 658–666, Arlington, Virginia. AUAI Press.