More constructions, more genres: Extending Stanford Dependencies

Marie-Catherine de Marneffe*, Miriam Connor, Natalia Silveira, Samuel R. Bowman, Timothy Dozat and Christopher D. Manning

*Linguistics Department
The Ohio State University
Columbus, OH 43210
mcdm@ling.osu.edu

Linguistics Department
Stanford University
Stanford, CA 94305
{mkconnor,natalias,sbowman,
tdozat,manning}@stanford.edu

Abstract

The Stanford dependency scheme aims to provide a simple and intuitive but linguistically sound way of annotating the dependencies between words in a sentence. In this paper, we address two limitations the scheme has suffered from: First, despite providing good coverage of core grammatical relations, the scheme has not offered explicit analyses of more difficult syntactic constructions; second, because the scheme was initially developed primarily on newswire data, it did not focus on constructions that are rare in newswire but very frequent in more informal texts, such as casual speech and current web texts. Here, we propose dependency analyses for several linguistically interesting constructions and extend the scheme to provide better coverage of modern web data.

1 Introduction

The Stanford dependency representation (de Marneffe et al. 2006, de Marneffe and Manning 2008b, henceforth SD) has seen wide usage within the Natural Language Processing (NLP) community as a standard for English grammatical relations, and its leading ideas are being adapted for other languages. This adaptation seems to be motivated by two principal advantages: (i) it provides a richer, more linguistically faithful typology of dependencies than the main alternatives and (ii) it adopts a simple, understandable, and uniform notation of dependency triples, close to traditional grammar. This combination makes it effective both for use by non-linguists working directly with linguistic information in the development of natural language understanding applications and also as a source of features for machine learning approaches. As a result, the representation has been variously used in relation extraction, text understanding, and machine translation applications.

While SD provides good coverage of core grammatical relations, such as subject, object, internal noun phrase relations, and adverbial and subordinate clauses, the standard remains underdeveloped and agnostic as to the treatment of many of the more difficult—albeit rarer—constructions that tend to dominate discussions of syntax in linguistics, such as *tough* adjectives, free relatives, comparative constructions, and small clauses. These constructions have been analyzed many times in various frameworks for constituency representation, and some have had some limited treatment in dependency grammar frameworks. Nevertheless, it is often not obvious how to analyze them in terms of dependencies, and currently the SD scheme does not offer explicit, principled analyses of these constructions.

Further, a current practical limitation is that the SD scheme was developed against newswire data, namely the Wall Street Journal portion of the Penn Treebank. It therefore gave relatively little consideration to constructions that are absent or rare in newswire, such as questions, imperatives, discourse particles, sentence fragments, ellipsis, and various kinds of list structures. Such constructions are, however, abundant in modern web texts. Emails, blogs, forum posts, and product reviews show a greater use of informal constructions, slang, and emoticons. It is important to handle these new genres by providing adequate dependency representations of the constructions which appear in such important modern genres.

Our goal in this paper is to address these two current limitations of Stanford dependencies. We extend the scheme to handle a wider array of linguistic constructions, both linguistically interesting constructions and those necessary to resolve practical problems in providing analyses for language use in modern web data.

Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), pages 187–196,
Prague, August 27–30, 2013.

2 The Stanford dependencies

The set of grammatical relations used by SD is principally drawn from the grammatical-relation oriented traditions in American linguistics: Relational Grammar (Perlmutter 1983), Head-driven Phrase Structure Grammar (HPSG, Pollard and Sag 1994), and particularly Lexical-Functional Grammar (LFG, Bresnan 2001). However, the actual syntactic representation adopted follows the functional dependency grammar tradition (Tesnière 1959, Sgall et al. 1986, Mel'cuk 1988) and other dependency grammars such as Word Grammar (Hudson 2010) in representing a sentence as a set of grammatical relations between its words. The SD scheme deviates from its LFG roots in trying to achieve the correct balance between linguistic fidelity and human interpretability of the relations, particularly in the context of relation extraction tasks. This leads it to sometimes stay closer to the descriptions of traditional grammar (such as for *indirect object*) in order to avoid making unnecessary theoretical claims that detract from broad interpretability. The focus of the SD scheme is on semantically useful relations.

Automatic annotation of dependencies using the SD scheme can be obtained for English text with a tool distributed with the Stanford Parser.[1] The tool uses a rule-based strategy to extract grammatical relations as defined in the SD scheme via structural configurations in Penn Treebank-style phrase-structure trees. The tool performs well, but as with all automatic parsing, it is important to maintain a distinction between the annotations it produces and the theoretical standard of the SD scheme: there can be a difference between the relation that the scheme would assign to two words and the relation that gets assigned by the tool. In this paper, we address the ideal relation structures rather than discussing parser performance directly. The Stanford dependency representation makes available several variants, suited to different goals. One, the *basic* representation, is a simple dependency tree over all the words in the sentence, which is useful when a close parallelism to the source text words must be maintained, such as when used as a representation for direct dependency parsing (Kübler et al. 2009). The expanded representation adds additional relations that cannot be expressed by a tree structure but may be useful for capturing semantic relations between entities in the sentence. Here, we will draw such additional dependencies as dashed arcs.

[1] `http://nlp.stanford.edu/software/lexparser.shtml`

3 Data

We have started an annotation effort to construct a gold-standard corpus of web data annotated with this extended SD scheme.[2] To provide the community with a gold-standard corpus that better captures linguistic phenomena present in casual text genres, we are annotating the parsed section of the Google Web Treebank (Petrov and McDonald 2012). This corpus contains about 250,000 words of unedited web text and covers five domains: questions and answers, emails, newsgroups, local business reviews and blogs. For each domain, between 2,000 and 4,000 sentences have been annotated with phrase-structure trees in the style of OntoNotes 4.0 by professional annotators from the Linguistic Data Consortium.

4 Linguistic analyses adopted for different constructions

The SD scheme has been in use for seven years, but still lacks principled analyses of many of the difficult English constructions that have been a staple of the formal linguistic literature. However, we have found in our annotation work that some of these constructions now arise prominently in terms of cases for which the correct analysis is unclear. Here we try to resolve several of these interesting corner-cases of English grammar. Some of these cases, such as tough adjectives and free relatives, were also discussed in recent evaluations of dependency extraction systems (Rimell et al. 2009, Bender et al. 2011) where the goal was to recover long dependencies. The family of CoNLL dependency schemes for English (Buchholz and Marsi 2006, Johansson and Nugues 2007), another common dependency representation in NLP, largely does not provide satisfying analyses for any of the cases presented here. Small clauses are the one exception, and the CoNLL treatment of small clauses is similar to ours.

4.1 Tough adjectives

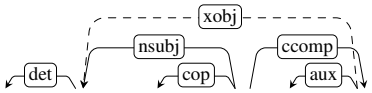
Tough adjectives, discussed in Bender et al. (2011), have posed challenges to nearly every syn-

[2] To date, except for the BioInfer corpus of biomedical texts (Pyysalo et al. 2007) and a small set of chosen long-distance dependency constructions (Rimell et al. 2009), there are no gold standard Stanford dependency annotations.

tactic formalism. For example, in (1a), the object of *find* can be "raised" to subject position in the main clause to form a tough adjective construction, as in (1b). One of the difficulties for generative grammar in modeling this construction is that the object being raised can be embedded arbitrarily deeply in the sentence, as in (1c).

(1) a. It was hard (for me) to find this address.
b. This address was hard (for me) to find.
c. This address was hard (for me) to work up the motivation to try to explain how to find.

In (1b), *this address* functions syntactically as the subject of *was hard*, but thematically as the object of *find*, and we want to represent both of these dependencies at some level. We simply give the surface subject (here *this address*) the expected *nsubj* label coming off the main predicate. We want to represent its relationship to the embedded verb as well though, since the surface subject is its thematic argument. Paralleling the existing *xsubj* dependency for the relationship between a verb and its controlling subject (which breaks the tree dependency structure), we introduce the *xobj* dependency to capture the relationship between a verb and its logical object when it breaks the tree dependency structure. So (1b) will have the dependency representation in (2), with an additional *xobj* dependency from *find* to *address*:



(2) This address was hard to find.

Further, there are two competing structural analyses for the optional *for NP* phrase. In one, the *for NP* is a PP complement of the main predicate, and in the other, *for* is a complementizer that takes a tense-less sentence. Chomsky (1973) argues on the basis of sentences like (3a–3b) that the experiencer *for NP* must be a true PP, not part of a complementizer.

(3) a. It is easy [PP for the rich] [SBAR for the poor to do all the work].
b. *All the work is easy [PP for the rich] [SBAR for the poor to do _]

When the *for* introduces an SBAR proposition (which is quite rare), the whole clause can "move" as a unit, as demonstrated in (4a), but when the *for* introduces a PP experiencer, the PP can "move" separately (4b), both supporting the hypothesis that the experiencer is not part of an SBAR.

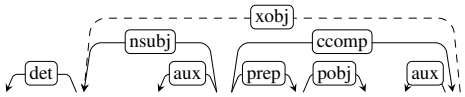
(4) a. [SBAR For the poor to do all the work] is easy [PP for the rich].
b. X-server is difficult [S to set up] [PP for everyone].[3]

We conclude from data like this that the PP is usually a separate constituent, and should be annotated as such; (3a) and (4b) should therefore be assigned the dependencies shown in (5a–5b).

(5) a. It is easy for the rich for the poor to do the work.

b. X-server is difficult to set up for everyone.

We opt to analyze *to find* in (1b) and *to set up* in (4b) as clausal complements (*ccomp*). Faithful to the original SD scheme, we reserve the use of the *xcomp* label for controlled complements in the LFG sense of functional control (Bresnan 1982) – where the subject of the complement is necessarily controlled by an argument of the governing verb. This is not the case here: the subject can be viewed as a covert PRO, which is coreferent with the *for PP* complement. We now have a complete analysis for *tough* adjectives. The dependency relations for the sentence in (1b) are given in (6) below.



(6) This address was hard for me to find.

4.2 Free relatives

Free relatives, which are discussed in Rimell et al. (2009), are likewise challenging because while their surface resemblance to embedded interrogatives invites a transformational treatment parallel to *wh*-questions, certain of their syntactic properties point to an analysis in which the *wh*-phrase serves as the head rather than as a subordinate element. To illustrate these two conflicting analyses, we will explore the implications of each treatment using the free relative phrases (italicized) in (7) below as our chief examples.

[3] https://mail.gnome.org/archives/gnome-accessibility-list/2003-May/msg00421.html

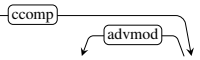
(7) a. Put your bag down *wherever you see a spot*.

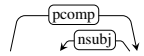
b. Sarah will talk to *whoever comes her way*.

c. I'll dress *however nicely you tell me to dress*.

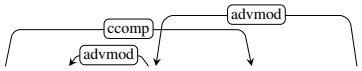
An initially attractive approach to analyzing (7a–7c) is to treat the free relatives identically to embedded *wh*-interrogative complements. On this approach, *wherever* is an *advmod* of *see*, *whoever* is the *nsubj* of *comes*, and *however nicely* (with *nicely* being the head of the *wh*-phrase) is an *advmod* of *dress*, resulting in the following dependency structures:

(8) a. Put your bag down wherever you see a spot





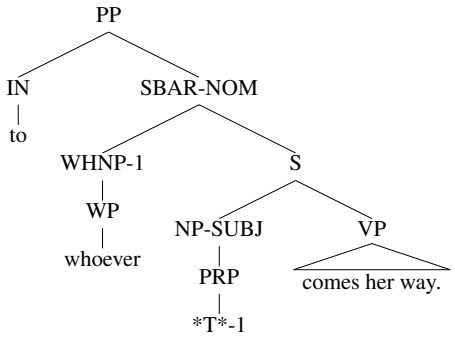
b. Sarah will talk to whoever comes her way



c. I'll dress however nicely you tell me to dress

The above treatment is analogous to a transformational analysis of free relatives in a phrase structure formalism. In such a treatment, the *wh*-phrase is generated inside the clause and moved to the clause-initial position through A′-movement. The Treebank II bracketing guidelines (Bies et al. 1995) take this approach, inserting a *T* node, indicating the trace of A′-movement, in the tree position where the *wh*-word was generated and coindexing it with the *wh*-word—see (9).

(9) Sarah will talk . . .



Under this analysis, we must treat *see* as a sentential complement of *put*, *comes* as a prepositional complement of *to*, and *tell* as a sentential complement of *dress*, as shown in (8a–8c) and (9).

However, as Bresnan and Grimshaw (1978) point out, this transformational analysis fails to capture certain key syntactic properties of free relatives. In particular, the free relative phrases in (7) do not really behave like sentential complements—in fact, substituting other sentential *wh*-complements for the free relative phrases in (7) leads to ungrammatical constructions:

(10) a. *Put your bag down *what table Al put his on*.

b. *Sarah will talk to *which person Fred talked to*.

c. *I'll dress *what dress I wore last time*.

We make better predictions if we analyze the free relatives like those in (7a), (7b), and (7c) as locative adverbial phrases, nominal phrases, and adverbial phrases, respectively. Substituting these phrase types for the free relatives in the original examples leads to perfectly natural constructions:

(11) a. Put your bag down *on the table*.

b. Sarah will talk to *that man over there*.

c. I'll dress *very nicely*.

In each example, the syntactic category assigned to the free relative phrase is identical to that of the *wh*-phrase within the free relative: *wherever* being locative, *whoever* being nominal, and *however nicely* being adverbial. Based on this observation (among others), Bresnan and Grimshaw (1978) argue for treating the *wh*-phrase as the head of the free relative. In their 1978 transformation grammar analysis, they then account for the appearance of movement with a deleted pronoun whose trace is coindexed with the *wh*-phrase and stipulate that the coindexed nodes must agree in certain grammatical features:

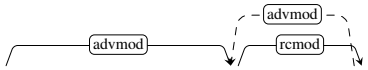
(12) a. Put your bag down [$_{\text{LocP}}$ wherever$_1$ [$_{\text{S}}$ you see a spot [*there*$\rightarrow \emptyset_1$]]].

b. Sarah will talk to [$_{\text{NP}}$ whoever$_1$ [$_{\text{S}}$ [*s/he*$\rightarrow \emptyset_1$] comes her way]].

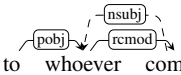
c. I'll dress [$_{\text{AdvP}}$ [however nicely]$_1$ [$_{\text{S}}$ you tell me to dress [*so*$\rightarrow \emptyset_1$]]].

So rather than follow the Treebank II guidelines, we adopt the approach of Bresnan and Grimshaw (1978), analyzing the *wh*-phrase as the head of the free relative and treating the sentential portion of the free relative phrase as a relative clause modifier on the head. We also mark the relationships inside the relative clause, between the

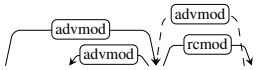
verb and the head of the *wh*-phrase, with additional dependencies, to preserve the semantic relationship between the two entities. The grammatical relations between the verb of the relative and the head of the *wh*-phrase correspond to the ones the traces would receive. Thus, we decompose the examples in (7a–7c) as follows:



(13) Put your bag down wherever you see a spot.



(14) Sarah will talk to whoever comes her way.



(15) I'll dress however nicely you tell me to dress.

4.3 Comparative constructions

The syntax of comparative constructions in English poses various challenges for linguistic theory, many of which are discussed in Bresnan (1973). We devoted special attention to canonical (in)equality comparisons between two elements, of the form: as_1 *X* as_2 *Y* and *more X than Y*.

4.3.1 *as ... as* constructions

In constructions of the form as_1 *X* as_2 *Y*, *X* and *Y* can be of a range of syntactic types, leading to surface forms such as those exemplified below:

(16) a. Commitment is as important as a player's talent.
b. Get the cash to him as soon as possible.
c. I put in as much flour as the recipe called for.

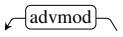
Note that there are analogous constructions with inequality comparatives for all of these, briefly discussed below; the analysis argued for in this subsection will largely extend to those. *X* takes the form of an AdjP, an AdvP, and an NP in (16a), (16b) and (16c), respectively. We analyze the as_1 *X* as_2 *Y* expression as modification on the *X* phrase; notice that preserving only the head of the *X* phrase always yields a grammatical sentence, indicating that this head determines the syntactic type of the whole phrase:

(17) a. Commitment is important.
b. Get the cash to him soon.
c. I put in flour.

This suggests that the head of *X* is the head of the whole structure (and therefore depends on nothing inside it) and that the $as_1 \ldots as_2$ *Y* phrase modifies the inner *X* phrase. Our analysis expresses this by making as_1 dependent on a head inside *X*. However, clearly $as_1 \ldots as_2$ *Y* is a comparative modifier, and it modifies a gradable property. That property is not always denoted by the head of *X*; *flour*, for example, does not seem to be the target of the comparison in (16c). To reflect that, our next analytic decision is to make as_1 dependent on the adjective, adverb or quantifier that represents the gradable property targeted by the comparison. The relation is *advmod*, consistent with other types of degree modification, such as (18).

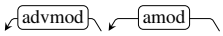
(18) a. Commitment is crucially important.
b. Get the cash to him very soon.
c. I put in too much flour.

With that, for (16a) we have:



(19) as important

For (16c), we make as_1 dependent on *much*, not *flour*, as it is the quantity of flour that is the target of the comparison:



(20) as much flour

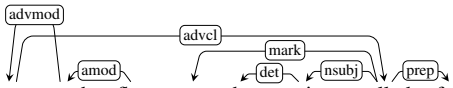
These decisions address the question of what the head of the entire phrase is, and how the comparative modifier interfaces with it. Next, we turn to questions about the internal structure of the comparative. It seems that as_1 has a privileged status over as_2, since it is possible to drop as_2 *Y* (21), but not as_1 (22):

(21) a. Commitment is (just) as important.
b. ?Get the cash to him (just) as soon.
c. I put in (just) as much flour.

(22) a. *Commitment is important as a player's talent.
b. *Get the cash to him soon as possible.
c. *I put in much flour as the recipe called for.

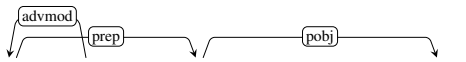
For this reason, and following other authors' syntactic analyses of the secondary term of a comparative as a complement (Huddleston and Pullum

2002), we make *as*$_2$ *Y* dependent on *as*$_1$. This still leaves the question of how to link *Y* with the rest of the phrase. It is clear that the material in *as*$_2$ *Y* can be clausal, as exemplified by (16c); it is also optional, as exemplified by (21). For that reason, we make it an *advcl*, dependent on *as*$_1$, with *as*$_2$ as a *mark*. This is consistent with the Penn Treebank annotations for these constructions. That gives us:



(23) as much flour as the recipe called for

In the case when *Y* is an NP, to remain consistent with the Penn Treebank annotations, we treat *as*$_2$ *Y* as a prepositional phrase. So we have:



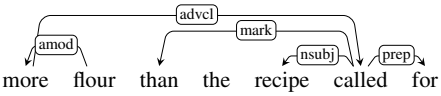
(24) as important as a player 's talent

4.3.2 *more . . . than* constructions

The analysis we give to expressions like *more . . . than* or *less . . . than*, as in (25), is very similar to the analysis of *as . . . as* discussed above.

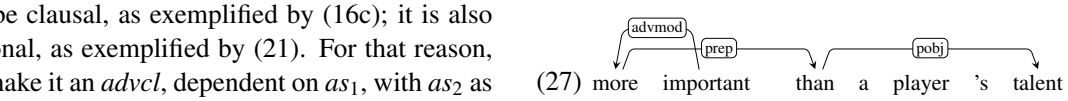
(25) I put in more flour than the recipe called for.

Again, we analyze the head of the *more X than Y* expression as the head of the *X* phrase, since keeping the head will yield a grammatical sentence, which in this case is exactly (17c). Also in parallel with the constructions above, we note that the relation between *more . . . than Y* and *X* has a parallel with other types of adverbial modifiers, as was shown in (18c). Therefore, we again label that relation *advmod*. As for *than Y*, again we take it to be an adverbial clause if *Y* is anything other than an NP. So we will have analyses such as:



(26) more flour than the recipe called for

When *Y* is an NP, we essentially adopt the analysis of Bresnan (1973), in which an *-er* morpheme that expresses the comparative value combines with *much* to form *more*; this provides an explanation for why *much* appears in (16c), where it combines with *as*, but not in (25), where it combines with *-er*. This is relevant because the resulting *more* in (25) is, syntactically, an adjectival modifier, as is *much* in (16c). Also, in parallel with our analysis of *as*$_1$ *X as*$_2$ *Y* and consistently with the Penn Treebank analysis, we call *than Y* a prepositional phrase when *Y* is a noun phrase. We therefore arrive at the following analysis for the comparative expression below:



(27) more important than a player 's talent

4.4 Small clauses

In the world of phrase structure, *small clauses*, like the bracketed example in (28), have two competing analyses: in the analysis correlated with the lexicalist approach (28a), both the entity and the predicate depend on the main verb; and in the one correlated with the transformational approach (28b), the entity depends on the predicate.

(28) a. We made [them leave].

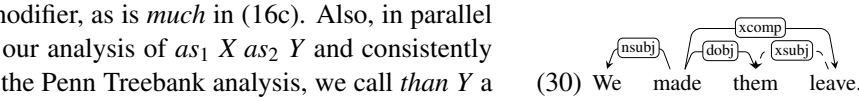
b. We made [them leave].

There is a substantial literature on small clauses and evidence for and against each structure (Borsley 1991, Culicover and Jackendoff 2005, Matthews 2007). The optimal analysis largely depends on the assumptions of the theory in question. The Penn Treebank adopts the analysis in (28b), putting both arguments of the main verb under an S node. Empirically, though, the small clause as a unit fails a considerable number of constituency tests, such as those in (29) (adapted from (Culicover and Jackendoff 2005)), which show that the small clause cannot move around in the sentence as a unit. So in the system we have been developing—which we aim to make as empirically motivated as possible—we choose to have both the entity and the predicate depend on the main verb (28a) as is also done in the CoNLL scheme, leading to the analysis in (30). This analysis also allows us to add an additional subject relation between the two components of the small clause when the small clause contains a verb (which CoNLL does not have). Adopting the other analysis, we would lose the link between the object and the higher verb.

(29) a. *What we made was them leave.

b. *We made without difficulty them leave.

c. *Them leave is what we made.

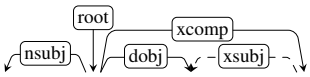


(30) We made them leave.

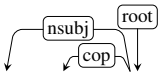
The Penn Treebank also recognizes small clause constructions where the predicate is a nominal or adjectival expression as in (31b) and (31c) respectively. We can extend the *xcomp* analysis to them by regarding the noun or adjective as also a predicate with a controlled subject. This is consistent with both the LFG analysis where the grammatical function XCOMP originated (Bresnan 1982) and the treatment of predicate nouns and adjectives in copula constructions in SD (de Marneffe and Manning 2008a).

(31) a. We made them leave.
b. We made them martyrs.
c. We made them noticeable.

For example (32a) has a parallel analysis to (32b):



(32) a. We made them martyrs.



b. They are martyrs.

Assigning the *xcomp* label offers a consistent analysis across all uses of the small clause construction and also emphasizes the fact that the second noun phrase is being used non-referentially, as a predicate instead of as an entity.

5 Extensions to the Stanford dependencies

In the process of annotating the Google Web Treebank, we also discovered a number of ways in which the SD standard needs to be modified to capture the syntax of a broader range of text genres. These changes, described in the following paragraphs, led to a new version of the extended SD scheme with 56 relations, listed in Figure 1.

5.1 New relations

discourse Colloquial writing contains interjections, emoticons, and other discourse markers which are not linked to their host sentences by any existing relation. We add a discourse element relation *discourse* which encompasses these constructions, including emoticons and all phrases headed by words that the Penn Treebank tags INTJ: interjections (*oh*, *uh-huh*, *Welcome*), fillers (*um*, *ah*), and discourse markers (*well*, *like*, *actually*).

(33) Hello, my name is Vera.

root - root
dep - dependent
  aux - auxiliary
    auxpass - passive auxiliary
    cop - copula
  arg - argument
    agent - agent
    comp - complement
      acomp - adjectival complement
      ccomp - clausal complement with internal subject
      xcomp - clausal complement with external subject
      obj - object
        dobj - direct object
        iobj - indirect object
        pobj - object of preposition
    subj - subject
      csubj - clausal subject
        csubjpass - passive clausal subject
      nsubj - nominal subject
        nsubjpass - passive nominal subject
  cc - coordination
  conj - conjunct
  expl - expletive (expletive "there")
  list - list item
  mod - modifier
    advmod - adverbial modifier
      neg - negation modifier
    amod - adjectival modifier
    appos - appositional modifier
    advcl - adverbial clause modifier
    det - determiner
    discourse - discourse element
    goeswith - goes with
    predet - predeterminer
    preconj - preconjunct
    mwe - multi-word expression modifier
    mark - marker (word introducing an *advcl* or *ccomp*)
    nn - noun compound modifier
    npadvmod - noun phrase adverbial modifier
      tmod - temporal modifier
    num - numeric modifier
    number - element of compound number
    prep - prepositional modifier
    poss - possession modifier
    possessive - possessive modifier (*'s*)
    prt - phrasal verb particle
    quantmod - quantifier modifier
    rcmod - relative clause modifier
    vmod - verbal modifier
    vocative - vocative
  parataxis - parataxis
  punct - punctuation
  ref - referent
  sdep - semantic dependent (breaking tree structure)
    xsubj - (controlled) subject
    xobj - (controlled) object

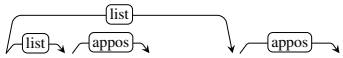
Figure 1: Extended Stanford dependencies.

goeswith Unedited text often contains multiple tokens that correspond to a single standard English word, as a result of reanalysis of compounds ("hand some" for "handsome") or input error ("othe r" for "other"). The non-head portions of these broken words are tagged GW in the treebank. We cannot expect preprocessing steps

(tokenization and normalization) to fix all of these errors, so we introduce the relation *goeswith* to reconnect these non-heads to their heads—usually the initial pieces of the words.

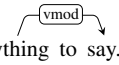
(34) They come here with out legal permission

list Web text often contains passages which are meant to be interpreted as lists of comparable items, but are parsed as single sentences. Email signatures in particular contain these structures, in the form of contact information. We label the contact information *list* as in (35). For the key-value pair relations that often occur in these contexts, we use the *appos* relation.



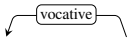
(35) Steve Jones Phone: 555-9814 Email: jones@abc.edf

vmod Since the distinction between *partmod* and *infmod* is straightforwardly reflected in the part-of-speech of the verb, we choose to cease duplicating information by merging these relations into a single one, *vmod*. We intend this to cover all cases of verb-headed phrases acting as modifiers, which are not full clauses.



(36) I don't have anything to say.

vocative In writing that directly addresses a dialog participant (e.g., emails and newsgroup postings), it is common to begin sentences by naming that other participant. We introduce the *vocative* relation to link these names to their host sentences.



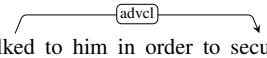
(37) Tracy, do we have concerns here?

xobj We introduce the relation *xobj* to capture the relationship between a verb and its displaced logical object. For further explanation, see the discussion of *tough* adjectives in section 4.1 above.

(38) Those were hard for him to find.

5.2 Modified and deleted relations

advcl Purpose clauses (*purpcl*) were singled out based on a semantic distinction, but distinctions were not made for other types of adverbial clause (temporal, causal, etc.). We make the scheme more uniform by collapsing purpose clauses with general adverbial clauses (*advcl*).



(39) She talked to him in order to secure the account.

amod Parenthetically marked ages have been treated as appositives (and marked *appos*), but we find that this violates the otherwise largely sound generalization that appositives fill the same semantic role as the NPs they modify, and essentially serve as alternative ways to identify the entities named by those NPs. Since, for example, it is not reasonable to infer (41) from (40), we choose to re-classify these cases as displaced adjectival modifiers, and label them *amod*.

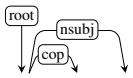
(40) John Smith (33) was from Kansas City, MO.

(41) 33 is from Kansas City, MO.

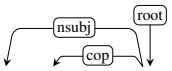
appos We abandon the *abbrev* relation and substitute *appos*: *abbrev* captured parenthetical expressions indicating abbreviations, but was used rarely, and provided little information not also captured by the more general *appos*.

(42) The Australian Broadcasting Corporation (ABC)

attr In *wh*-questions such as (43), we treated the copular verb as the root, and the *wh*-word was an *attr*. We are abandoning the *attr* relation, leading to the following analysis which parallels that of affirmative copular sentences like (44) where the predicate is the root. Copular sentences are now treated more uniformly than before.

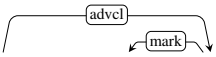


(43) What is that?

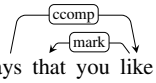


(44) That is a sturgeon.

mark The former *complm* relation captured overt complementizers like "that" in complement clauses (*ccomp*). We follow the intuition from HPSG that this relation captures approximately the same structural relation as *mark* in adverbial clauses (45), and provides no information that *mark* would not also provide. We thus abandon the *complm* relation, and substitute *mark* (46).



(45) I like to swim when it rains.



(46) He says that you like to swim.

mwe We have found several additional constructions that we believe meet the criteria to be considered multi-word expressions for the purposes of the *mwe* relation, which is intended for "multi-word idioms that behave like a single function word." We add the following constructions:

> *at least, at most, how about, how come, in case, in order (to), of course, prior to, so as (to), so that, what if, whether or not*

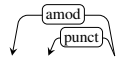
This is in addition to already-recognized constructions such as:

> *rather than, as well as, instead of, such as, because of, instead of, in addition to, all but, such as, because of, instead of, due to*

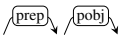
(47) Of course I'll go!

Ultimately, the choice of what to count as a *mwe* reflects a cut across the continuous cline of grammaticalization, and is necessarily arbitrary.

punct We do not follow Choi and Palmer (2012) in using the relations *hmod* and *hyph* for the non-head words of split-up hyphenated words and the hyphens respectively. We find the usage of hyphens is very inconsistent, and so we prefer to apply the most appropriate general relation that holds between the hyphenated components rather than adopt these labels. For the hyphen, when it is used to construct compound words (48), we treated it as punctuation and assign the *punct* relation, but when it is used in place of an en dash to indicate a range, as in (49), we treat it as a preposition and assign the *prep* relation.



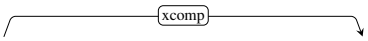
(48) short - term humanitarian crisis



(49) French Indochina War (1946 – 1954)

rel *rel* has been used in a small number of constructions to mark the head words of *wh*-phrases introducing relative clauses. We are retiring the relation: we will mark the heads of *wh*-phrases in accordance with their role in the relative clause (usually *nsubj*, *dobj*, *pobj*, or *prep*), and any such head whose role cannot be identified will be marked with the generic relation *dep*.

xcomp The *xcomp* relation is specified in de Marneffe and Manning (2008a) to apply to any non-finite complement clause which has its subject controlled by the subject of the next higher verb. However, complement clauses with object control—wherein the object of the higher verb controls the subject of an embedded clause, as in (50)—structurally have more in common with subject control cases rather than with the canonical *ccomp* complement clause with which it would otherwise be classified. Further, these cases are grouped as XCOMP in LFG. In order to ensure that this crucial notion of external control is reliably captured, we expand the definition of *xcomp* to include cases of both subject and object control.



(50) It allowed material previously stored to decompose.

6 Conclusions

We extend the coverage of the SD scheme by presenting principled analyses of linguistically interesting constructions and by positing new relations to capture frequent constructions in modern web data. Our approach has been empirical: all the construction types discussed here appear in the Google English Web Treebank data that we are annotating. We are currently incorporating our extensions of the SD standard into the freely available converter tool associated with the scheme. So far, there has not been any quantitative evaluation of the tool: there has only been some qualitative analysis as well as a focus on some relations as reported in (Rimell et al. 2009, Bender et al. 2011), but ultimately the annotated gold standard corpus we are creating will enable a thorough evaluation of the converter tool.

Acknowledgment

Annotation of the English Web Treebank with gold Stanford dependencies has been supported by a gift from Google Inc. We thank the anonymous reviewers, John Bauer, and Joakim Nivre for helpful comments on the analyses we propose.

References

Bender, Emily M., Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 397–408.

Bies, Ann, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger, 1995. Bracketing guidelines for Treebank II style Penn Treebank project.

Borsley, Robert D. 1991. *Syntactic Theory: A Unified Approach*. Edward Arnold.

Bresnan, Joan. 1973. Syntax of the comparative clause construction in English. *Linguistic Inquiry* 4.

Bresnan, Joan. 1982. Control and complementation. In Joan Bresnan (ed.), *The Mental Representation of Grammatical Relations*, pp. 282–390. MIT Press.

Bresnan, Joan. 2001. *Lexical-functional syntax*. Blackwell.

Bresnan, Joan, and Jane Grimshaw. 1978. The syntax of free relatives in English. *Linguistic Inquiry* 9(3):331–391.

Buchholz, Sabine, and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pp. 149–164.

Choi, Jinho D., and Martha Palmer. 2012. Guidelines for the Clear style constituent to dependency conversion. Technical report, University of Colorado Boulder, Institute of Cognitive Science.

Chomsky, Noam. 1973. Conditions on transformations. In Stephen Anderson and Paul Kiparsky (eds.), *A Festschrift for Morris Halle*, pp. 232–286. New York: Holt, Rinehart & Winston.

Culicover, Peter W., and Ray Jackendoff. 2005. *Simpler syntax*. Oxford University Press.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pp. 449–454.

de Marneffe, Marie-Catherine, and Christopher D. Manning. 2008a. Stanford typed dependencies manual. Technical report, Stanford University.

de Marneffe, Marie-Catherine, and Christopher D. Manning. 2008b. The Stanford typed dependencies representation. In *Proceedings of the COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*, pp. 1–8.

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

Hudson, Richard A. 2010. *An Introduction to Word Grammar*. Cambridge University Press.

Johansson, Richard, and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*.

Kübler, Sandra, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*, volume 2 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool.

Matthews, Peter H. 2007. *Syntactic relations: A critical survey*. Cambridge University Press.

Mel'cuk, Igor A. 1988. *Dependency syntax: Theory and practice*. SUNY Press.

Perlmutter, David M. (ed.). 1983. *Studies in Relational Grammar*, volume 1. University of Chicago Press.

Petrov, Slav, and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *First Workshop on Syntactic Analysis of Non-Canonical Language*.

Pollard, Carl, and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. University of Chicago Press.

Pyysalo, Sampo, Filip Ginter, Katri Haverinen, Juho Heimonen, Tapio Salakoski, and Veronika Laippala. 2007. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. In *Proceedings of BioNLP 2007: Biological, translational, and clinical language processing (ACL07)*, pp. 25–32.

Rimell, Laura, 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*, pp. 813–821.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. D. Reidel Publishing Company.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Librairie C. Klincksieck.