Who Did What to Whom?
A Contrastive Study of Syntacto-Semantic Dependencies

Angelina Ivanova♣, Stephan Oepen♣, Lilja Øvrelid♣, and Dan Flickinger♡
♣ University of Oslo, Department of Informatics
♡ Stanford University, Center for the Study of Language and Information
{angelii|oe|liljao}@ifi.uio.no, danf@stanford.edu

Abstract

We investigate aspects of interoperability between a broad range of common annotation schemes for syntacto-semantic dependencies. With the practical goal of making the LinGO Redwoods Treebank accessible to broader usage, we contrast seven distinct annotation schemes of functor–argument structure, both in terms of syntactic and semantic relations. Drawing examples from a multi-annotated gold standard, we show how abstractly similar information can take quite different forms across frameworks. We further seek to shed light on the representational 'distance' between pure bilexical dependencies, on the one hand, and full-blown logical-form propositional semantics, on the other hand. Furthermore, we propose a fully automated conversion procedure from (logical-form) meaning representation to bilexical semantic dependencies.[†]

1 Introduction—Motivation

Dependency representations have in recent years received considerable attention from the NLP community, and have proven useful in diverse tasks such as Machine Translation (Ding & Palmer, 2005), Semantic Search (Poon & Domingos, 2009), and Sentiment Analysis (Wilson et al., 2009). Dependency representations are often claimed to be more 'semantic' in spirit, in the sense that they directly express predicate–argument relations, i.e. *Who did What to Whom?* Several of the shared tasks of the Conference on Natural Language Learning (CoNLL) in the past years have focused on data-driven dependency parsing—producing both syntactic (Nivre et al., 2007) and semantic dependencies (Hajič et al., 2009)—and have made available gold standard data sets (dependency banks) for a range of different languages. These data sets have enabled rigorous evaluation of parsers and have spurred considerable progress in the field of data-driven dependency parsing (McDonald & Nivre, 2011).

Despite widespread use, dependency grammar does not represent a unified grammatical framework and there are large representational differences across communities, frameworks, and languages. Moreover, many of the gold-standard dependency banks were created by automated conversion from pre-existing constituency treebanks—notably the venerable Penn Treebank for English (PTB; Marcus et al., 1993)—and there exist several conversion toolkits which convert from constituent structures to dependency structures. This conversion is not always trivial, and the outputs can differ notably in choices concerning head status, relation inventories, and formal graph properties of the resulting depedency structure. Incompatibilty of representations and differences in the 'granularity' of linguistic information hinder the evaluation of parsers across communities (Sagae et al., 2008).

In this paper, we pursue theoretical as well as practical goals. First, we hope to shed more light on commonalities and differences between a broad range of dependency formats—some syntactic, others semantic in spirit. Here, divergent representations are in part owed to relatively superficial design decisions, as well as in part to more contentful differences in underlying linguistic assumptions; thus, for some classes of syntagmatic relations there may be one-to-one correspondences across families of dependency formats, while for other classes (or other subsets of formats), interconversion may not be possible in general. Building on freely available gold-standard annotations in seven different formats, we contrast these representations both qualitatively and quantitatively. A better understanding

[†]We are indebted to Emily Bender, Rebecca Dridan, and four anonymous reviewers for their feedback on this work.

Proceedings of the 6th Linguistic Annotation Workshop, pages 2–11,
Jeju, Republic of Korea, 12-13 July 2012.

of such cross-representational relations and related trade-offs will be beneficial to creators and users of syntacto-semantic annotations alike.

Our notion of *syntacto-semantic* information encompasses dependencies ranging from 'classic' syntactically defined grammatical functions (like *subject*, *complement*, or *adjunct*) to more abstract (proto-)roles in propositional semantics (like *agent* or *location*). Indeed, we observe that 'syntactic' vs. 'semantic' representations are anything but clearly separated universes, and that dependency schemes often seek to bring into equilibrium syntactic as well as semantic considerations. At the same time, our focus (and that of much recent and current work in annotation and parsing) is on *bilexical* dependencies, i.e. representations that limit themselves to directed and labeled relations between observable, lexical units of the linguistic signal.

Second, our work is grounded in the practical goal of making pre-existing, large and framework-specific treebanks accessible to a broader range of potential users. Specifically, the Deep Linguistic Processing with HPSG Initiative (DELPH-IN[1]) has produced both manually and automatically annotated resources making available comparatively fine-grained syntactic and semantic analyses in the framework of Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag, 1994). For English, the so-called LinGO Redwoods Treebank (Oepen et al., 2004) contains gold-standard annotations for some 45,000 utterances in five broad genres and domains; comparable resources exist for Japanese (Bond et al., 2004) and are currently under construction for Portuguese and Spanish (Branco et al., 2010; Marimon, 2010). We develop an automated, parameterizable conversion procedure for these resources that maps HPSG analyses into either syntactic or semantic bilexical dependencies. Similar conversion procedures have recently been formulated for functional structures within the LFG framework (Øvrelid et al., 2009; Cetinoglu et al., 2010). In the design of this unidirectional (i.e. lossy) mapping, we apply and corroborate the cross-framework observations made in the more linguistic part of this study.

The paper has the following structure: Section 2 introduces the corpus and annotations we take as our point of departure; Section 3 contrasts analyses of select linguistic phenomena by example; and Section 4 develops an automated conversion from HPSG analyses to bilexical dependencies.

[1]See `http://www.delph-in.net` for background.

2 The Multi-Annotated PEST Corpus

At the 2008 Conference on Computational Linguistics (COLING), a workshop on *Cross-Framework and Cross-Domain Parser Evaluation* organized a shared task on comparing different target representations for grammatical analysis (Bos et al., 2008). For a selection of ten sentences from the PTB, the organizers encouraged contrastive studies over a set of parallel, gold-standard annotations in eight different formats. This collection, dubbed PEST (*Parser Evaluation Shared Task*), remains a uniquely valuable resource, despite its small size, for its careful selection of grammatical phenomena, broad coverage across frameworks, and general availability.[2] In the following we briefly review our selection of dependency representations from the PEST data set that provide the vantage point for the current work—using the dimensions identified earlier: head status, relation types, and graph properties.

In the dependency parsing community, it is commonly assumed that dependency structures are directed trees: labeled, directed graphs, where the word tokens in a sentence constitute the nodes, and (i) every token in the sentence is a node in the graph (combined with a designated root node, conventionally numbered as 0), (ii) the graph is (weakly) connected, (iii) every node in the graph has at most one head, and (iv) the graph is acyclic (Nivre et al., 2007). Although these formal constraints facilitate efficient syntactic parsing, they are not necessarily warranted from a pure linguistic point of view. In fact, many of the more theoretical accounts of dependency grammar do not adhere to these requirements (Hudson, 1984). The choice of heads in a dependency representation is another area where individual schools differ substantially. Generally speaking, we may distinguish between formats that take

[2]See `http://lingo.stanford.edu/events/08/pe/` for details. Note that, in addition to the gold-standard 'core' of ten PTB sentences, the full PEST collection includes another three dozen sentences from other corpora with some cross-framework annotations, though not in all of the formats and in some cases not manually validated.

	Description	H	T	C
CD	CoNLL Syntactic Dependencies	F	+	+
CP	CoNLL PropBank Semantics	S	–	–
SB	Stanford Basic Dependencies	S	+	+
SD	Stanford Collapsed Dependencies	S	–	–
EP	Enju Predicate – Argument Structures	S	–	+
DT	DELPH-IN Syntactic Derivation Tree	F	+	+
DM	DELPH-IN Minimal Recursion Semantics	S	–	–

Table 1: Summary of dependency formats, where the columns labeled *H* indicate the head status (functional vs. substantive), *T* whether or not structures are acyclic trees, and *C* whether or not all tokens are weakly connected.

a largely functional view of head status—e.g. functional elements like auxiliaries, subjunctions, and infinitival markers are heads—and more substantive or content-centered approaches where the lexical verbs or arguments of the copula are heads. The inventory of dependency relations constitutes another dimension of variation between frameworks. Typically, these relations are largely based on syntactic functions; however, there is also a tradition for using relations more akin to semantic roles, e.g. in the so-called tectogrammatical layer of the Prague Dependency Treebank (Sgall et al., 1986).

Specifically, we look at the 'core' part of the PEST data set that contains ten sentences from the Wall Street Journal portion of the PTB in the following formats: CoNLL 2008 (a) Syntactic Dependencies and (b) PropBank Semantics, Stanford (c) basic and (d) collapsed dependencies, and (e) Enju predicate–argument structures. For comparison to the DELPH-IN HPSG resources, we augment these annotations with gold-standard (f) syntactic and (g) semantic analyses from the LinGO English Resource Grammar (ERG; Flickinger, 2000).[3]

CoNLL Syntactic Dependencies (CD) As discussed earlier, several of the CoNLL shared tasks in the past decade addressed the identification of syntactic or semantic bilexical relations. For English, syntactic dependencies in the PEST collection were obtained by converting PTB trees with the PennConverter software (Johansson & Nugues, 2007), which relies on head finding rules (Magerman, 1994; Collins, 1999) and the functional annotation already present in the PTB annotation. In this format, the dependency representations adhere to the formal graph constraints mentioned above and syntactic heads are largely functional. The dependency relations are mostly syntactic, but also express a few more semantic distinctions like different types of adverbial modification—temporal, locative, etc.

CoNLL PropBank Semantics (CP) For the 2008 CoNLL shared task on joint learning of syntactic and semantic dependencies (Surdeanu et al., 2008), the PropBank and NomBank annotations 'on top' of the PTB syntax (Palmer et al., 2005; Meyers et al., 2004) were converted to bilexical dependency form. This conversion was based on the dependency syntax already obtained for the same data set (CD, above) and heuristics which identify the semantic head of an argument with its syntactic head.The conversion further devotes special attention to arguments with several syntactic heads, discontinuous arguments, and empty categories (Surdeanu et al., 2008). The representation does not adhere to the formal constraints posed above; it lacks a designated root node, the graph is not connected, and the graph is not acyclic. The choices with respect to head status are largely substantive. The dependency relations employed for this representation are PropBank semantic roles, such as `A0` (proto-agent), `A1` (proto-patient), and various modifier roles.

Stanford Basic Dependencies (SB) The Stanford Dependency scheme, a popular alternative to CoNLL-style syntactic dependencies (CD), was originally provided as an additional output format for the Stanford parser (Klein & Manning, 2003). It is a result of a conversion from PTB-style phrase structure trees (be they gold standard or automatically produced)—combining 'classic' head finding rules with rules that target specific linguistic constructions, such as passives or attributive adjectives (Marneffe et al., 2006). The so-called *basic* format provides a dependency graph which conforms to the criteria listed above, and the heads are largely content rather than function words. The grammatical relations are organized in a hierarchy, rooted in the generic relation 'dependent' and containing 56 different relations (Marneffe & Manning, 2008), largely based on syntactic functions.

[3] Among others, annotations in the Prague Dependency format would be interesting to compare to, but currently these are unfortunately not among the formats represented in the PEST corpus.

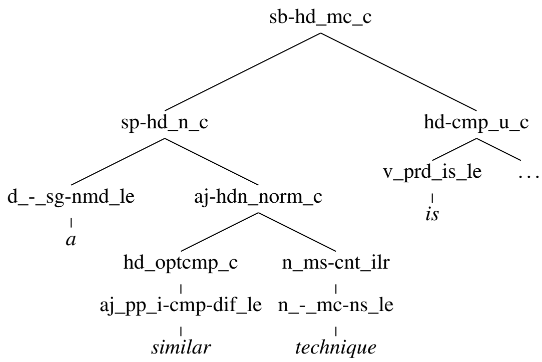


Figure 1: Syntactic derivation tree from the ERG.

{ e_{12}
$_{1}$:_a_q(BV x_6)
e_9:_similar_a_to(ARG1 x_6)
x_6:_technique_n_1
e_{12}:_almost_a_1(ARG1 e_3)
e_3:_impossible_a_for(ARG1 e_{18})
e_{18}:_apply_v_to(ARG2 x_6, ARG3 x_{19})
$_{2}$:udef_q(BV x_{19})
e_{25}:_other_a_1(ARG1 x_{19})
x_{19}:_crop_n_1
e_{26}:_such+as_p(ARG1 x_{19}, ARG2 x_{27})
$_{3}$:udef_q(BV x_{27})
$_{4}$:udef_q(BV x_{33})
x_{33}:_cotton_n_1
$_{5}$:udef_q(BV i_{38})
x_{27}:implicit_conj(L-INDEX x_{33}, R-INDEX i_{38})
$_{6}$:udef_q(BV x_{43})
x_{43}:_soybeans/nns_u_unknown
i_{38}:_and_c(L-INDEX x_{43}, R-INDEX x_{47})
$_{7}$:udef_q(BV x_{47})
x_{47}:_rice_n_1
}

Figure 2: ERG Elementary Dependency Structure.

Stanford Collapsed Dependencies (SD) Stanford Dependencies also come in a so-called *collapsed* version[4], where certain function words, such as prepositions, introduce dependency relations (rather than acting as nodes in the graph). Moreover, certain dependents—such as subjects of control verbs—have more than one head. The collapsed representation thus does not meet the formal graph criteria mentioned above: it is not connected, since not all tokens in the sentence are connected to the graph, a node may have more than one head, and there may also be cycles in the graph.

Enju Predicate–Argument Structures (EP) The Enju system is a robust, statistical parser obtained by learning from a conversion of the PTB into HPSG (Miyao, 2006). Enju outputs so-called predicate–argument structures (often dubbed PAS, but in our context henceforth EP), which primarily aim to capture semantic relations and hence prefer substantive heads over functional ones and encode most types of syntactic modifiers as predicates (i.e. heads) rather than arguments. The gold-standard Enju predicate–argument structures in the PEST collection were obtained semi-automatically from the HPSG conversion of the PTB;[5] they do not obey our formal graph constraints, much for the same reasons as we see in CP or SD.

[4]The collapsed scheme actually is the default option when running the Stanford converter, whereas the basic format must be requested by a specific command-line flag (‘-basic’).

[5]For unknown reasons, the original PEST release lacks Enju annotations for one of the ten ‘core’ sentences. We were able to obtain a stand-in analysis with the help of Prof. Yusuke Miyao (one of the original PEST coordinators), however, which we will include in our re-release of the extended resource.

DELPH-IN Syntactic Derivation Tree (DT) Similar to Enju, the LinGO English Resource Grammar (ERG; Flickinger, 2000) is couched in the HPSG framework; in contrast to Enju, however, the ERG has been engineered fully analytically (fully independent of the PTB), growing grammatical coverage continuously since the early 1990s. Figure 1 shows the ERG derivation tree for part of our running example (see below), which provides a compact ‘recipe’ for construction of the full HPSG analysis. Internal nodes in the tree are labeled with identifiers of HPSG constructions (subject–head, specifier–head, and head–complement, in the top of the tree), leaf nodes with types of lexical entries. In Section 4 below, we convert DELPH-IN derivations into syntactic bilexical dependencies.

DELPH-IN Minimal Recursion Semantics (DM) As part of the full HPSG sign, the ERG also makes available a logical-form representation of propositional semantics in the format of Minimal Recursion Semantics (MRS; Copestake et al., 2005). While MRS proper utilizes a variant of predicate calculus that affords underspecification of scopal relations, for our goal of projecting semantic forms onto bilexical dependencies, we start from the reduction of MRS into the Elementary Dependency Structures (EDS) of Oepen & Lønning (2006), as shown in Figure 2. EDS is a lossy (i.e. non-reversible) conversion from MRS into a variable-free dependency graph; graph nodes (one per line in Figure 2) correspond to elementary predications from the original logical form and are connected by arcs labeled with MRS argument indices: ARG1, ARG2, etc. (where BV is reserved for what is the bound variable of a quantifier in the full MRS).[6] Note that, while EDS already brings us relatively close to the other formats, there are graph nodes that do not correspond to individual words from our running example, for example the underspecified quantifiers for the bare noun phrases (udef_q) and the binary conjunction implicit_conj that ties together *cotton* with *soybeans and rice*. Furthermore, some words are semantically empty (the predicative copula, infinitival *to*, and argument-marking preposition), and the EDS does not form a tree (*technique*, for example, is the ARG1 of *similar*, ARG2 of *apply*, and bound variable of *a*). In Section 4 below, we develop a mapping from DELPH-IN Elementary Dependency Structures to 'pure' bilexical semantic dependencies.

3 Contrasting Analyses by Example

Availability of the ten PEST sentences in different dependency representations allows us to observe and visualize cross-format differences both qualitatively and quantitatively.[7] To illustrate some pertinent contrasts, Figure 3 visualizes syntacto-semantic dependencies in seven formats for the PEST example:

(1) *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.*

For the CoNLL, Stanford, and DELPH-IN formats, which each come in two variants, we present the more syntactic dependencies above (in red) and the more semantic dependencies below (blue) the actual string. This running example illustrates a range of linguistic phenomena such as coordination, verbal chains, argument and modifier prepositional phrases, complex noun phrases, and the so-called *tough* construction.

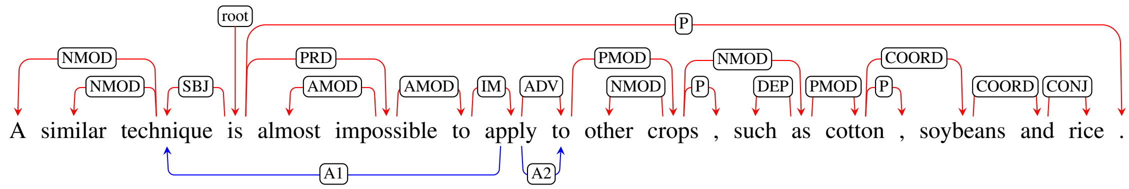
Figure 3 reveals a range of disagreements across formats. The analysis of coordination represents a well-known area of differences between various dependency schemes, and this is also the case for our example. Strikingly, none of the formats agree on the analysis of the coordination *cotton, soybeans and rice*. CoNLL Syntactic Dependencies (CD) exhibit the so-called Mel'čuk-style analysis of coordination (Mel'čuk, 1988), where the first conjunct is regarded as the head of coordinated structures, and the consequent conjuncts and coordinating construction are sequentially linked to each other. DELPH-IN MRS (DM) is similar, but the coordinating conjunction is treated as functional and therefore does not contribute a dependency node. CoNLL Propositional Semantic (CP) has no analysis for the coordinated structure, since it only analyzes main arguments in the sentence. In both Stanford schemes, the first conjunct is the head of the coordination construction, and the other conjuncts depend on it—but the basic (SB) and collapsed (SD) representations differ because a coordination relation is propagated to all conjuncts in SD. In the DELPH-IN Derivation (DT), finally, the coordinating conjunction is the head for all conjuncts.

Above, we proposed a distinction between more functional vs. more substantive dependency formats, and although this distinction does not clearly separate the different analyses in Figure 3, it points to some interesting differences. Where the majority of the schemes identify the root of the sentence as the finite verb *is*, the Stanford schemes—being largely substantive in their choices concerning syntactic heads—annotate the predicative adjective *impossible* as the root. Further, the infinitive *to apply* receives different interpretations in the formats. The infinitival marker depends on the main verb in CP, SB, and SD—whereas CD, EP, and DT regard it as the head. In CD, SB, and DT, prepositions are dependents, as illustrated by *(such) as*; in EP and DM, prepositional modifiers are heads; and SD 'collapses' prepositions to yield direct relations between the nominal head of the preposition (*crops*) and its internal argument.

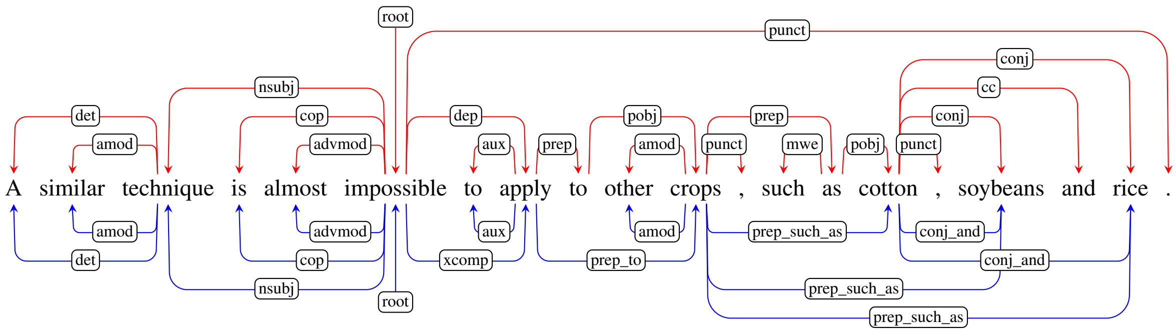
The treatment of noun phrases cuts across this distinction between functional and substantive ap-

[6] In the textual rendering of our EDS in Figure 2, nodes are prefixed with unique identifiers, which serve to denote node reentracy and the targets of outgoing dependency arcs.

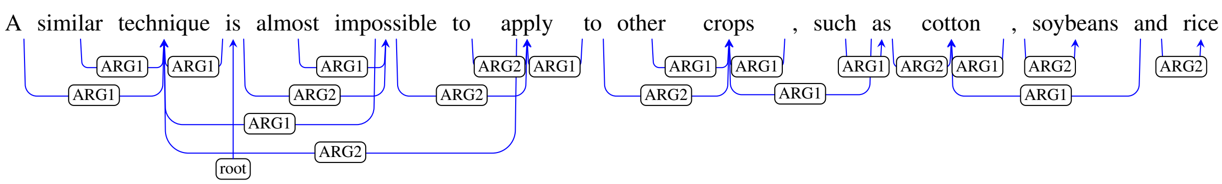
[7] In this section, we use bilexical dependency variants of the DELPH-IN analyses, anticipating the conversion procedure sketched in Section 4 below.



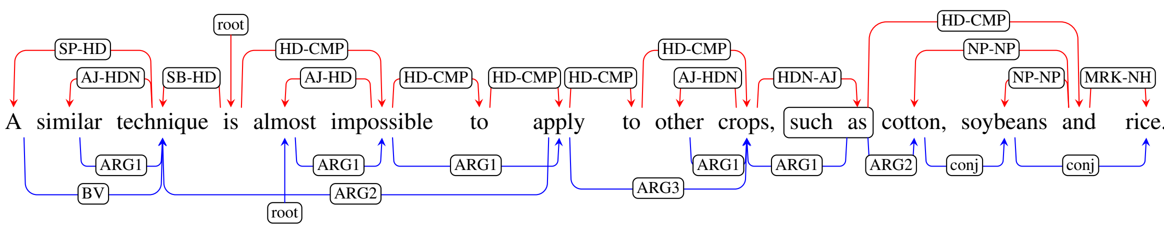
(a) CoNLL 2008 *syntactic dependencies* (CD; top) and *propositional semantics* (CP; bottom).



(b) Stanford Dependencies, in the so-called *basic* (SB; top) and *collapsed & propagated* (SD; bottom) variants.



(c) Enju *predicate-argument structures* (EP).



(d) DELPH-IN *syntactic derivation tree* (DT; top) and *Minimal Recursion Semantics* (DM; bottom).

Figure 3: Dependency representations in (a) CoNLL, (b) Stanford, (c) Enju, and (d) DELPH-IN formats.

proaches. In *a similar technique*, one can treat the determiner and attributive adjective as dependents of the noun, which is what we find in the CD, SB, SD, and DT schemes. Alternatively, one may consider the noun to be a dependent of both the determiner and the adjective, as is the case in the schemes deriving from predicate logic (EP and DM).

Our running example also invokes the so-called *tough* construction, where a restricted class of adjectives (*impossible* in our case) select for infinitival VPs containing an object gap and, thus, create a long-distance dependency (Rosenbaum, 1967; Nanni, 1980, inter alios). In the dependency analyses in Figure 3, we observe three possible heads for the noun *technique*, viz. *is* (CD, EP, DT), *impossible* (SB and SD), and *apply* (CP, EP, and DM). The long-distance dependency (between *technique* and *apply*) is marked only in the more semantic schemes: CP, EP, and DM.

Our comparison shows a range of pertinent qualitative differences. To further quantify the degree of overlap between different analyses in the PEST data, we abstract from tokenization subtleties by aligning representations across formats in terms of native PTB tokenization. For example, in the ERG punctuation is attached to the words (e.g. *crops,*), multiword expressions (*such as*) act as one entity, and unlike in the PTB hyphenated words (like *arms-*

	CD	CP	SB	SD	EP	DT	DM
CD	*19*	1	12	5	6	12	2
CP	1	*2*	1	0	1	1	1
SB	12	1	*19*	10	4	7	3
SD	5	0	10	*14*	2	4	3
EP	6	1	4	2	*20*	6	8
DT	12	1	7	4	6	*15*	0
DM	2	1	3	3	8	0	*11*

Table 2: Pairwise unlabelled dependency overlap.

	CD	CP	SB	SD	EP	DT	DM
CD		.171	.427	.248	.187	.488	.115
CP	.171		.171	.177	.122	.158	.173
SB	.427	.171		.541	.123	.319	.147
SD	.248	.177	.541		.14	.264	.144
EP	.187	.122	.123	.14		.192	.462
DT	.488	.158	.319	.264	.192		.13
DM	.115	.173	.147	.144	.462	.13	

Table 3: Pairwise Jaccard similarity on PEST 'core'.

control, not present in our example) are split into component parts (a similar, but not identical splitting is also used in CD). Conversely, in the PTB-derived formats punctuation marks are separate tokens, but EP consistently drops sentence-final punctuation.

Table 2 shows how many *unlabelled* dependency arcs each pair of formats have in common for our running example. The most similar pairs here are CD and DT, CD and SB, and SB and SD. The values in the diagonal of the table expose the total number of dependencies in a given representation.

For each pair of formats we computed its Jaccard similarity index $\frac{|A \cap B|}{|A \cup B|}$, by macro-averaging over all ten sentences, i.e. computing total counts for the union and intersection for each pair of formats $\langle A, B \rangle$. The results are presented in Table 3, where we observe that Jaccard indices are comparatively low across the board and do not exceed 55 % for any pair. This measure (unsurprisingly) shows that SB and SD are the most similar formats among all seven. The DELPH-IN dependency representations demonstrate comparatively strong interoperability with other schemes, since CD corresponds well with DT syntactically, while EP correlates with DM among the more semantic formats.

4 Automated Conversion from HPSG

In the following paragraphs, we outline an automated, parameterizable, and lossy conversion from the native DELPH-IN analyses to bilexical dependencies, both syntactic and semantic ones.

Background: LinGO Redwoods The LinGO Redwoods Treebank (Oepen et al., 2004) is a collection of English corpora annotated with gold-standard HPSG analyses from the ERG. The annotations result from manual disambiguation among candidate analyses by the grammar, such that the treebank is entirely composed of structures derived from the ERG as an explicit, underlying model.[8] Synchronized to major releases of the ERG, Redwoods has been continuously updated to take advantage of improved coverage and precision of the grammar. The current, so-called Seventh Growth provides manually validated analyses for some 45,000 sentences from five domains, which also represent different genres of text. Automatically parsed and disambiguated versions of the English Wikipedia and comprehensive samples of user-generated web content are available in the exact same formats (so-called *treecaches*; Flickinger et al., 2010; Read et al., 2012).

Syntax: Derivations to Dependencies The transformation of DELPH-IN derivation trees to syntactic dependencies is, in principle, straightforward—as the HPSG constructions labeling internal nodes of the tree (see Figure 1) directly determine syntactic head daughters. Thus, for the conversion it is sufficient to (a) eliminate unary nodes and (b) extract bilexical dependencies in a single tree traversal. Here, HPSG constructions (like sb-hd_mc_c in Figure 1, i.e. a subject–head combination in a main clause) introduce dependency relations holding between the (lexical head of) the head daughter and (that of) each non-head daughter. Note that we further generalize the 150 or so fine-grained ERG constructions to 50 major construction types, e.g. sb-hd_mc_c to just sb-hd in Figure 3d.

Semantics: Logical Form to Dependencies The complete ERG Elementary Dependency Structure for our running example is shown in Figure 2 (al-

[8] The downside of this grammar-driven approach to treebank creation is that the collection can contain gaps for inputs that the grammar is unable to analyze and ones where annotators reject all available candidates as inadequate. At present, between ten and twenty percent of all utterances in Redwoods lack a gold-standard analysis for one of these reasons.

```
[transparent]
implicit_conj L-INDEX
/_c$/ L-INDEX
[relational]
/_c$/ conj L-INDEX R-INDEX
implicit_conj conj L-INDEX R-INDEX
```

Figure 4: Excerpt from the ERG configuration file.

though we are not showing some additional information on each node, relating EDS components to input tokens). The conversion procedure for 'regular' *lexical* relations, i.e. ones that correspond to actual tokens, is simple. For example, _other_a_1(ARG1 x_{19}) in Figure 2 contributes an ARG1 dependency between *other* and *crops* in Figure 3d, because x_{19} is the identifier of the EDS node labelled _crop_n_1.

Besides this basic mechanism, our converter supports three 'special' classes of relations, which we call (a) *transparent*, (b) *relational*, and (c) *redundant*. The latter class is of a more technical nature and avoids duplicates in cases where the EDS gave rise to multiple dependencies that only differ in their label (and where labels are considered equivalent), as can at times be the case in coordinate structures.[9]

Our class of so-called *transparent* relations includes the semantic relation associated with, for example, nominalization, where in the underlying logic a referential instance variable is explicitly derived from an event. In terms of bilexical dependencies, however, we want to conceptually equate the two EDS nodes involved. In our running example, in fact, coordination provides an example of transparency: in the EDS, there are two binary conjunction relations (implicit_conj and _and_c), which conceptually correspond to group formation; node i_{38} (corresponding to *and*) is the second argument of the implicit conjunction. For our semantic bilexical dependencies, however, we opt for the analysis of Mel'čuk (see Section 3 above), which we achieve by making interchangeable conjunction nodes with their left arguments, i.e. nodes i_{38} and x_{43}, as well as x_{27} and x_{33}, in Figure 2.

Finally, somewhat similar to the 'collapsing' available in Stanford Dependencies, our class of so-called *relational* predicates allows the creation of dependency labels transcending EDS role indices, which we apply for, among others, possession, subordination, apposition, and conjunction. The two conj dependencies in Figure 3d, for example, hold between left and right arguments of the two conjunctions, as per the excerpt from the ERG-specific conversion specification shown in Figure 4.[10].

[9]The full underlying logical forms make a distinction between scopal vs. non-scopal arguments, which is washed out in the EDS. The existence of seemingly redundant links in coordinate structures is owed to this formal reduction.

5 Conclusions—Outlook

With the goal of making the Redwoods Treebank resources accessible to the broader NLP community, we have presented both a qualitative and quantitative comparison of a range of syntacto-semantic dependency formats, in order to make explicit the information contained in the treebank representations, as well as contrasting these to already existing formats. Our comparative analysis shows a large variation across formats and—although this is not surprising per se—highlights the importance of contrastive studies. In this article we have furthermore presented an automatic conversion procedure, which converts the HPSG representations in the treebanks to a set of syntactic and semantic dependencies.

In terms of next steps, we will release the transformed Redwoods Treebank and conversion software in the hope that the new resources will enable various follow-up activities. Both the CoNLL and Stanford formats have been used to train data-driven dependency parsers, and it is a natural next step to train and evaluate parsers on the converted DELPH-IN formats. In order to do so, further adjustments may have to be made to the DM format to convert it into a dependency tree. In light of the broader variety of domains available in Redwoods, the converted data will enable experimentation in domain and genre adaptation for parsers. As a further step in gauging the utility of the various dependency formats, it would also be interesting to contrast these in a downstream application making use of dependency representations.

[10]Our conversion software is fully parameterizable in terms of the different classes of relations, to allow for easy experimentation and adaptation to other DELPH-IN grammars. We plan to contribute the converter, extended PEST collection, and a version of Redwoods transformed to bilexical dependencies into the open-source DELPH-IN repository; see `http://www.delph-in.net/lds/` for details and access.

References

Bond, F., Fujita, S., Hashimoto, C., Kasahara, K., Nariyama, S., Nichols, E., Ohtani, A., Tanaka, T., & Amano, S. (2004). The Hinoki treebank. A treebank for text understanding. In *Proceedings of the 1st International Joint Conference on Natural Language Processing* (pp. 158 – 167). Hainan Island, China.

Bos, J., Briscoe, E., Cahill, A., Carroll, J., Clark, S., Copestake, A., Flickinger, D., Genabith, J. van, Hockenmaier, J., Joshi, A., Kaplan, R., King, T. H., Kuebler, S., Lin, D., Loenning, J. T., Manning, C., Miyao, Y., Nivre, J., Oepen, S., Sagae, K., Xue, N., & Zhang, Y. (Eds.). (2008). *Proceedings of the COLING 2008 Workshop on Cross-Framework and Cross-Domain Parser Evaluation.* Manchester, UK: Coling 2008 Organizing Committee.

Branco, A., Costa, F., Silva, J., Silveira, S., Castro, S., Avelãs, M., Pinto, C., & Graça, J. (2010). Developing a deep linguistic databank supporting a collection of treebanks. The CINTIL DeepGramBank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation.* Valletta, Malta.

Cetinoglu, O., Foster, J., Nivre, J., Hogan, D., Cahill, A., & Genabith, J. van. (2010). Lfg without c-structures. In *Proceedings of the 9th international workshop on treebanks and linguistic theories.* Tartu, Estonia.

Collins, M. J. (1999). *Head-driven statistical models for natural language parsing.* Unpublished doctoral dissertation, University of Pennsylvania, Philadelphia.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics. An introduction. *Journal of Research on Language and Computation*, *3*(4), 281 – 332.

Ding, Y., & Palmer, M. (2005). Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics* (pp. 541 – 548). Ann Arbor, MI, USA.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering*, *6 (1)*, 15 – 28.

Flickinger, D., Oepen, S., & Ytrestøl, G. (2010). WikiWoods. Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation.* Valletta, Malta.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpanek, J., Straňàk, P., Surdeanu, M., Xue, N., & Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Natural Language Learning.*

Hudson, R. A. (1984). *Word grammar.* Blackwell.

Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In J. Nivre, H.-J. Kaalep, & M. Koit (Eds.), *Proceedings of NODALIDA 2007* (pp. 105 – 112).

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics* (pp. 423 – 430).

Magerman, D. M. (1994). *Natural language parsing as statistical pattern recognition.* Unpublished doctoral dissertation, Stanford University.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics*, *19*, 313 – 330.

Marimon, M. (2010). The Spanish Resource Grammar. In *Proceedings of the 7th International Conference on Language Resources and Evaluation.* Valletta, Malta.

Marneffe, M.-C. de, MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation.* Genova, Italy.

Marneffe, M.-C. de, & Manning, C. D. (2008). The Stanford typed dependencies representation.

In *Proceedings of the COLING08 Workshop on Cross-Framework Parser Evaluation* (pp. 1 – 8).

McDonald, R., & Nivre, J. (2011). Analyzing and integrating dependency parsers. *Computational Linguistics*, *37*(1), 197 – 230.

Mel'čuk, I. (1988). *Dependency syntax: Theory and practice.* Albany: SUNY Press.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., , & Grishman, R. (2004). The NomBank project: An interim report. In *Proceedings of HLT-EACL Workshop: Frontiers in Corpus Annotation.*

Miyao, Y. (2006). *From linguistic theory to syntactic analysis. Corpus-oriented grammar development and feature forest model.* Unpublished doctoral dissertation, University of Tokyo, Tokyo, Japan.

Nanni, D. (1980). On the surface syntax of constructions with *easy*-type adjectives. *Language*, *56*(3), 568–591.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., & Yuret, D. (2007). CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task session of EMNLP-CoNLL 2007* (pp. 915 – 932).

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S., & Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, *13*(2), 95 – 135.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2004). LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Research on Language and Computation*, *2*(4), 575 – 596.

Oepen, S., & Lønning, J. T. (2006). Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (pp. 1250 – 1255). Genoa, Italy.

Øvrelid, L., Kuhn, J., & Spreyer, K. (2009). Improving data-driven dependency parsing using large-scale lfg grammars. In *Proceedings of the 47th Meeting of the Association for Computational Linguistics.*

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71 – 105.

Pollard, C., & Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar.* Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Poon, H., & Domingos, P. (2009). Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing.* Singapore.

Read, J., Flickinger, D., Dridan, R., Oepen, S., & Øvrelid, L. (2012). The WeSearch Corpus, Treebank, and Treecache. A comprehensive sample of user-generated content. In *Proceedings of the 8th International Conference on Language Resources and Evaluation.* Istanbul, Turkey.

Rosenbaum, P. S. (1967). *The grammar of English predicate complement constructions* (Vol. 46). MIT Press.

Sagae, K., Miyao, Y., Matsuzaki, T., & Tsujii, J. (2008). Challenges in mapping of syntactic representations for framework-independent parser evaluation. In *Proceedings of the Workshop on Automated Syntactic Annotations for Interoperable Language Resources at ICGL'08.* Hong Kong.

Sgall, P., Hajičová, E., & Panevová, J. (1986). *The meaning of the sentence in its pragmatic aspects.* Dordrecht: Reidel.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., & Nivre, J. (2008). The CoNLL-2008 Shared Task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Natural Language Learning* (p. 159-177).

Wilson, T., Wiebe, J., & Hoffman, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, *35*(3), 399 – 433.