pyBART: Evidence-based Syntactic Transformations for IE

Aryeh Tiktinsky   **Yoav Goldberg**   **Reut Tsarfaty**
Allen Institute for AI, Tel Aviv, Israel
Bar Ilan University, Ramat-Gan, Israel
{aryeht,yoavg,reutt}@allenai.org

Abstract

Syntactic dependencies can be predicted with high accuracy, and are useful for both machine-learned and pattern-based information extraction tasks. However, their utility can be improved. These syntactic dependencies are designed to accurately reflect syntactic relations, and they do not make semantic relations explicit. Therefore, these representations lack many explicit connections between content words, that would be useful for downstream applications. Proposals like English Enhanced UD improve the situation by extending universal dependency trees with additional explicit arcs. However, they are not available to Python users, and are also limited in coverage. We introduce a broad-coverage, data-driven and linguistically sound set of transformations, that makes event-structure and many lexical relations explicit. We present pyBART, an easy-to-use open-source Python library for converting English UD trees either to Enhanced UD graphs or to our representation. The library can work as a standalone package or be integrated within a spaCy NLP pipeline. When evaluated in a pattern-based relation extraction scenario, our representation results in higher extraction scores than Enhanced UD, while requiring fewer patterns.

1 Introduction

Owing to neural-based advances in parsing technology, NLP researchers and practitioners can now accurately produce syntactically-annotated corpora at scale. However, the use and empirical benefits of the dependency structures themselves remain limited. Basic syntactic dependencies encode the functional connections between words but lack many functional and semantic relations that exist between the content words in the sentence. Moreover, the use of strictly-syntactic relations results in structural diversity, undermining the efforts to effectively extract coherent semantic information from the resulting structures.

Thus, human practitioners and applications that "consume" these syntactic trees are required to devote substantial efforts to processing the trees in order to identify and extract the information needed for downstream applications, such as information and relation extraction (IE). Meanwhile, semantic representations (Banarescu et al., 2013; Palmer et al., 2010; Abend and Rappoport, 2013; Oepen et al., 2014) are harder to predict with sufficient accuracy, calling for a middle ground.

Indeed, De Marneffe and Manning (2008) introduced *collapsed* and *propagated* dependencies, in an attempt to make some semantic-like relations more apparent. The Universal Dependencies (UD) project[1] similarly embraces the concept of *Enhanced Dependencies* (Nivre et al., 2018)), adding explicit relations that are otherwise left implicit. Schuster and Manning (2016) provide further enhancements targeted specifically at English (Enhanced UD).[2] Candito et al. (2017) suggest further enhancements to address diathesis alternations.[3]

In this work we continue this line of thought, and take it a step further. We present pyBART, an

---

[1] universaldepdenencies.org

[2] In this paper we do not distinguish between the Universal Enhanced UD and Schuster and Manning (2016)'s Enhanced++ English UD. We refer to their union on English as Enhanced UD.

[3] Efforts such as PropS (Stanovsky et al., 2016) and PredPatt (White et al., 2016), share our motivation of extracting predicate-argument structures from treebank-trainable trees, though outside of the UD framework. Efforts such as KNext (Durme and Schubert, 2008) automatically extract logic-based forms by converting treebank-trainable trees, for consumption by further processing. HLF (Rudinger and Van Durme, 2014), DepLambda (Reddy et al., 2016) and UDepLambda (Reddy et al., 2017) attempt to provide a formal semantic representation by converting dependency structures to logical forms. While they share a high-level goal with ours — exposing functional relations in a sentence in a unified way — their end result, logical forms, is substantially different from pyBART structures. While providing substantial benefits for semantic parsing applications, logical forms are less readable for non-experts than labeled relations between content words. As these efforts rely on dependency trees as a backbone, they could potentially benefit from pyBART's focus on syntactic enhancements on top of (E)UD.
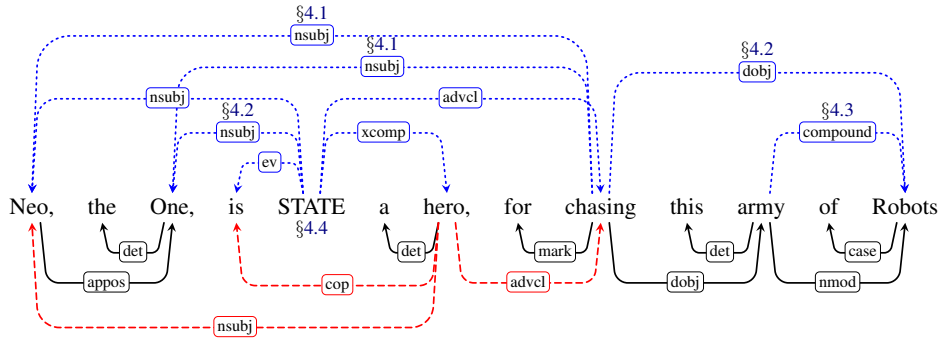
47

Figure 1: Representation of *Neo, the One, is a hero, for chasing this army of Robots*. The arcs above the sentence are BART additions. The ones below are EUD. Red arcs are removed in BART while black are retained.

easy-to-use Python library which converts English UD trees to a new representation that subsumes the English Enhanced UD representation and substantially extends it. We designed the representation to be linguistically sound and automatically recoverable from the syntactic structure, while exposing the kinds of relations required by IE applications. Some of these modifications are illustrated in Figure 1.[4] We aim to make event structure explicit, and cover as many linguistically plausible phenomena as possible. We term our representation BART (The BIU-AI2 Representation Transformation).

To assess the benefits of BART with respect to UD and other enhancements, we compare them in the context of a pattern-based relation extraction task, and demonstrate that BART achieves higher F_1 scores while requiring fewer patterns.

The python conversion library, `pyBART`, integrates with the spaCy[5] library, and is available under an open-source Apache license. A web-based demo for experimenting with the converter is also available. `https://allenai.github.io/pybart/`.

2 The BART Representation

We aim to provide a representation that will be useful for downstream NLP tasks, while retaining the following key properties. The proposal has to be **(i) based on syntactic structure** and **(ii) useful for information seeking applications**. As a consequence of (ii), we also want it to **(iii) make event structure explicit** and **(iv) allow favoring recall over precision**.

Being **based on syntax** as the backbone would allow us to capitalize on independent advances in

syntactic parsing, and on its relative domain independence. We want our representation to be not only accurate but also **useful for information seeking applications**. This suggests a concrete methodology (§2.1) and evaluation criteria (§5): we choose which relations to focus on based on concrete cases attested in relation extraction and QA-corpora, and evaluate the proposal based on the usefulness in a relation extraction task.

In general, information-seeking applications favor **making events explicit**. Current syntactic representations prefer to assign syntactic heads as root predicates, rather than actual eventive verb. In contrast, we aim to center our representation around the main event predicate in the sentence, while indicating event properties such as aspectuality (*Sam started walking*) or evidentiality (*Sam seems to like them*) as modifiers of rather than heads. To do this in a consistent manner, we introduce a new node of type STATE for copular sentences, making their event structure parallel to those containing finite eventive verbs (§4.4)

Finally, downstream users may prefer to **favor recall over precision** in some cases. To allow for this, we depart from previous efforts that refrain from providing any uncertain information. We chose to explicitly expose some relations which we believe to be useful but judge to be uncertain, while clearly marking their uncertainty in the output. This allows users to experiment with the different cases and assess the reliability of the specific constructions in their own application domain. We introduce two uncertainty marking mechanisms, discussed in §2.3.

2.1 Data-driven Methodology

Our departure point is the English EUD representation (Schuster and Manning, 2016) and related

---

[4]Some preserved UD relations are omitted for readability.
[5]`https://spacy.io`

efforts discussed above, which we seek to extend in a way which is useful to information seeking applications. To identify relevant constructions that are not covered by current representations, we use a data-driven process. We consider concrete relations that are expressed in annotated task-based corpora: a relation extraction dataset (ACE05, (Walker et al., 2006)), which annotates relations and events, and a QA-SRL dataset (He et al., 2015) which connects predicates to sentence segments that are perceived by people as their (possibly implied) arguments. For each of these corpora, we consider the dependency paths between the annotated elements, looking for cases where a direct relation in the corpus corresponds to an indirect dependency path in the syntactic graph. We identify recurring cases that we think can be shortened, and which can be justified linguistically and empirically. We then come up with proposed enhancements and modifications, and verify them empirically against a larger corpus by extracting cases that match the corresponding patterns and browsing the results.

2.2 Formal Structure

As is common in dependency-based representations, BART structures are labeled, directed multi-graphs whose nodes are the words of a sentence, and the labeled edges indicate the relations between them. Some constructions add additional nodes, such as copy-nodes (Schuster and Manning, 2016) and STATE nodes (§4.4).

An innovative aspect of our approach is that each edge is associated with additional information beyond its dependency label. This information is structured as follows:

SRC: a field indicating the origin of this edge—either "UD" for the original dependency edges, or a pair indicating the type and sub-type of the construction that resulted in the BART edge (e.g., {SRC=(conj,and)} or {SRC=(adv,while)}).

UNC, ALT: optional fields indicating uncertainty, described below.

2.3 Embracing uncertainty

Some syntactic constructions are ambiguous with respect to the ability to propagate information through them. Rather than giving up on all ambiguous constructions, we opted to generate the edges and mark them with an UNC=TRUE flag, deferring the decision regarding the validity of the edge to the user:

```
1  # Load a UD-based english model
2  nlp = spacy.load("en_ud_model")
3
4  # Add BART converter to spaCy's pipeline
5  from pybart.api import converter
6  converter = converter( ... )
7  nlp.add_pipe(converter, name="BART")
8
9  # Test the new converter component
10 doc = nlp("He saw me while driving")
11 me_token = doc[2]
12 for par_tok in me_token._.parent_list:
13     print(par_tok)
14
15 # Output:
16 {'head': 2, 'rel':'dobj', 'src':'UD'}
17 {'head': 5, 'rel': 'nsubj',
18   'src':('advcl','while'), 'alt':'0'}
```

Figure 2: Usage example of pyBART's spaCy-pipeline component.
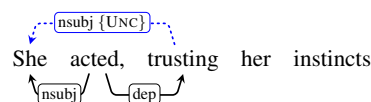
In some cases, we can identify that one of two options is possible, but cannot determine which. In these cases we report both edges, but mark them explicitly as alternatives to each other. This is achieved with an ALT=X field on both edges, with X being a number indicating the pair.
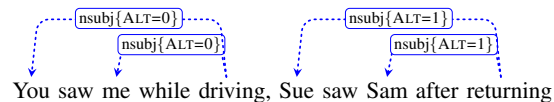
3 Python code and Web-demo

The pyBART library provides a Python converter from English UD trees to BART. pyBART subsumes the enhancements of the EUD Java implementation provided in Stanford Core-NLP,[6] and extends them as described in §4. While pyBART's default performs all enhancements, it can be configured to follow a more selective behavior. pyBART has two modes: (1) a converter from CoNLLU-formatted UD trees to CoNLLU-formatted BART structures;[7] and (2) a spaCy (Honnibal and Montani, 2017) pipeline component.[8] After registering pyBART as a spaCy pipeline, tokens on the analyzed document will have a `._.parent_list` field, containing the

---

[6]https://nlp.stanford.edu/software/stanford-dependencies.html

[7]The extra edge information is linearized into the dependency label after a '@' separator.

[8]This requires a spaCy model trained to produce UD trees, which we provide.

list of parents of the token in the BART structure. Each item is a dictionary specifying—in addition to the parent-token id and dependency label—also the extra information described in §2.2. See Figure 2 for an illustration of the API.

A web-based demo that parses sentences into both EUD and BART graphs, visualizes them, and compares their outputs, is also provided.[9]

4 Coverage of Linguistic Phenomena

BART conversion consists of four conceptual changes from basic UD. The first type propagates shared arguments between predicates in **nested structures**. The second type shares arguments between **parallel structures**. The third type attempts to unify **syntactic alternations** to reduce diversity, making structures that carry similar meaning also similar in structure. Finally, the forth type is designed to make **event structure** explicit in the syntactic representation, allowing finite verbs that indicate event properties to act as event modifiers rather than root predicates. In accordance with that, we further introduce a new STATE node, that acts as the main predicate node for *stative* (copular, verb-less) sentences.

4.1 Nested Structures

Our first type of conversions propagates an external core argument to be explicitly linked as the subject of a subordinate clause.

Complement control: The various EUD representations explicitly indicate the external subjects of *xcomp* clauses containing a *to* marker. We embrace this choice and extend it to cover also clauses without a *to* marker, including imperative clauses and clauses with controlled gerunds.

(1)

Let my people go!
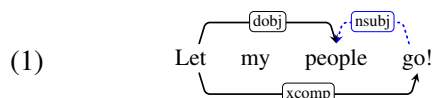
Noun-modifying clauses: Similarly, EUD links the empty subject of a finite relative clause to the corresponding argument of the external clause. We extend this behavior to also cover **reduced relative clauses** (2a), and we follow Candito et al. (2017) in also including other relative clauses such as **noun-modifying participles** (2b).
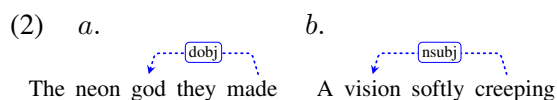
(2) a.

The neon god they made

b.
A vision softly creeping

Adverbial clauses and "dep": Adverbial modifier clauses that miss a subject, often modify the subject of the main clause. We propagate the external subject to be the subject of the internal clause.[10]

(3)

You shouldn't text while driving
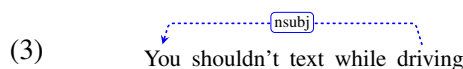
We observe that many **dep** edges empirically behave like adverbial clauses, and treat them similarly. We mark these edges as "uncertain".

4.2 Parallel structures

The second type of conversions identifies parallel structures in which the latter instance is elliptical, and share the missing core argument contributed by the former instance.

Apposition: Similarly to the PropS proposal (Stanovsky et al., 2016), we share relations across *apposition parts*, making the two, currently hierarchical, phrase, more duplicate-like.

(4)

E.T., the Extraterrestrial, phones home
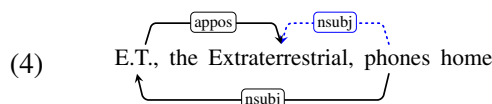
Modifiers in conjunction: In modified coordinated constructions, we share prepositional (5) and possessive (6) modifiers between the coordinated parts. Since dependency trees are *inherently* ambiguous between conjoined modification and single-conjunct modification, (e.g, compare (5) to "Mogly was lost and raised by wolves", or (6) to "my Father and E.T."), we mark both as UNC.

(5)

I was taught and raised by wolves

(6)

My father and mother met here
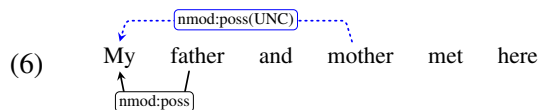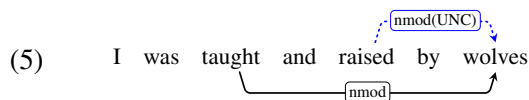
Elaboration/Specification Clauses: For noun nominal modifiers that have the form of an *elaboration* or *specification*, we share the head of the modified noun with its dependent modifier. That is, if the modification is marked by *like* or *such as* prepositions, we propagate the head noun to the nominal dependent.

(7)

I enjoy fruits such as apples

---
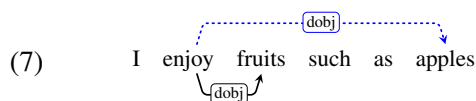
[10]In external clauses that include a subject and an object, ambiguity may arise as to which is to be modified. We propagate both and mark the edges as alternates (ALT, §2.3)).

Indexicals: the interpretation of locative and temporal indexicals such as *here*, *there* and *now* depends on the situation and the speaker, and often modify not only the predicate but the entire situation. We therefore share the adverbial modification from the noun to the main verb. Due to their situation-specific nature, we mark these as UNC.

(8) He wonders in these woods here
[dependency tree: advmod(UNC), nmod, advmod]
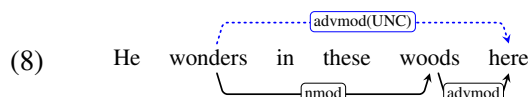
Compounds: Shwartz and Waterson (2018) show that in many cases, compounds can be seen as having a multiple-head. Therefore, we share the existing relations across the compound parts.

(9) I used canola oil
[dependency tree: dobj(UNC), dobj]
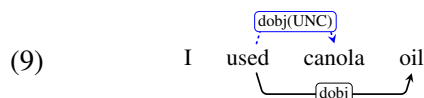
As many compounds *do* have a clear head (e.g. *I used baby oil*, where *baby* is clearly not the head), we mark these as uncertain.

4.3 Syntactic Alternations

This type of conversions aim to unify syntactic variability. We identify structures that are syntactically different but share (some) semantic structure, and add arcs or nodes to expose the similarity.

The Passivization Alternation: Following Candito et al. (2017) we relate the *passive* alteration to its *active* variant.

(10) The Sheriff was shot by Bob
*[dependency tree: dobj, nsubj, nsubjpass, nmod:by]*
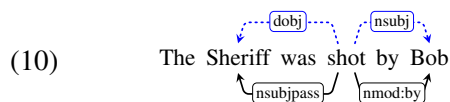
Hyphen reconstruction: Noun-verb Hyphen Constructions (HC) which are modifying a nominal can be seen as conveying the same information as a copular sentence wherein the noun is the subject and the verb is the predicate. To explicitly indicate this, we add to all modifying noun-verb HCs a *subject* and a *modifier* relation originating at the verb-part of the HC.

(11) A Miami - based company
[dependency tree: nmod, nsubj, compound, amod]
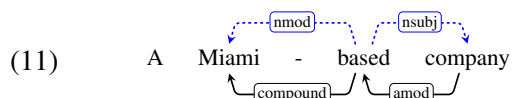
Adjectival modifiers: Adjectival modification can be viewed as capturing the same information as a predicative copular sentence conveying the same meaning (so, "a green apple" implies that "an apple is green"). To explicitly capture this productive implication, we add a subject relation from each adjectival modifier to its corresponding modified noun.

(12) I see dead people
*[dependency tree: nsubj]*
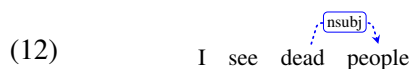
Genitive Constructions: Genitive cases can be alternatively expressed as a compound. We add a compound relation to unify the expression of genitives across *X of Y* and *compound* structures.

(13) Army of zombies
[dependency tree: compound]
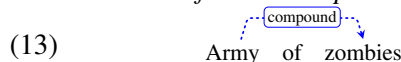
4.4 Event-Centered Representations

In many sentences, the finite root predicate does not indicate the main event. Instead, a verb in the subordinated clause expresses the event, and the finite verb acts as its modifier. For example, in sentences like "He started working", "He seems to work there", the main event indicated is "work", while the root predicates ("started", "seemed") modify this event. Here, we present a chain of changes that puts emphasis on *events* by delegating copular and tense auxiliaries (is, was), evidentials (seem, say) and various aspectual verbs (started, continued) to be clausal modifiers, rather than heads of the sentence. This creates a further challenge, since there is a prevalent discrepancy between predicative sentences such as "He works" and copular sentences as "He is smart". The UD structure for the latter lacks a node that clearly indicates a *stative* event (in Vendler (1957)'s terminology). We remedy this by adding a node to represent the STATE and have tense, aspect, modality and evidentiality directly modifying it.[11]

Copular Sentences and Stative Predicates: We added to all copula constructions new node named *STATE*, which represents the *stative* event introduced by the copular clause. This node becomes the root, and we rewire the entire clause around this *STATE*. By doing so we unify it with the structures of clauses with finite predicative. Once we added the *STATE* node, we form a new relation, termed *ev*, to mark event/state modifications. The resulting structure is as follows:

(14) Tomorrow is STATE another day
*[dependency tree: nsubj, ev, xcomp, cop, nsubj]*
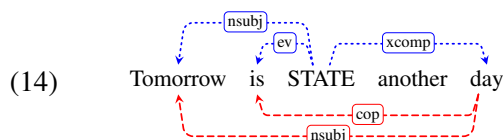
Evidential reconstructions: We can now explicitly mark properties of events as dependents of the verbal or stative root by means of the label *ev*. We do so, using verbs' white-lists, for verbs marking
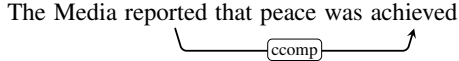
---

[11]Pragmatically, some users prefer to not have non-word nodes. pyBART supports this by providing a mode that treats the copula as the head, retaining the other modifications.

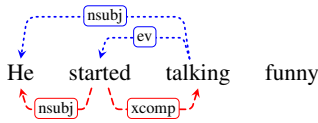evidentiality (15) and for reported-speech (16).

(15)
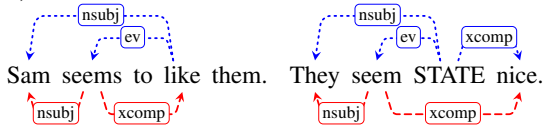
(16)

Aspectual constructions: Finally, we can now also mark aspectual verbs as modifying the complement (matrix) verb denoting the main event. The complement (matrix) verb becomes the root of the dependency structure, and we add the new *ev* relation to mark the aspectual modification of the event.

(17)

5 Evaluation

Our proposed representation attempts to target information-seeking applications, but is it effective? We evaluate the resulting graph structures against the UD and Enhanced UD representations, in the context of a relation-extraction (RE) task. Concretely, we evaluate the representations on their ability to perform pattern-based RE on the TACRED dataset (Zhang et al., 2017).

We use an automated and reproducible methodology: for each of the representations, we use the RE train-set to acquire extraction patterns. We then apply the patterns to the dev-set, compute F1-scores, and, for each relation, filter the patterns that hurt F1-score. We then apply the filtered pattern-set to the test-set, and report F1 scores.

To acquire extraction patterns, we use the following procedure: given a labeled sentence consisting of a relation name and the sentence indices of the two entities participating in the relation, we compute the shortest dependency path between the entities, ignoring edge directions. We then form an extraction pattern from the directed edges on this path. We consult a list of trigger words (Yu et al., 2015) collected for the different relations. If a trigger word or its lemma is found on the path, we form an unlexicalized path except for the trigger word (i.e. E1 <nsubj "founded" >dobj >compound E2). If no trigger-word is found, the path is lexicalized with the word's lemmas (i.e. E1 <nsubj "reduce" >dobj "activity" >compound E2).

Representation	Precision	Recall	F1
UD	76.53	30.65	43.77
Enhanced UD	77.63	32.37	45.69
Ours(w/o-Enhanced)	73.96	33.48	46.09
Ours	74.62	36.65	**49.15**

Table 1: Effectiveness of the different representations on the TACRED relation extraction task.
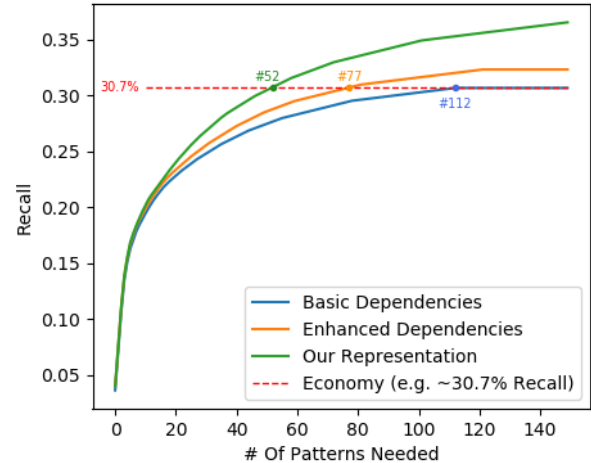

Figure 3: Economy comparison: Recall vs number of patterns, for the different representations.

We use this procedure to compare UD, Enhanced UD (EUD), BART without EUD enhancements, and full BART, which is a superset of Enhanced UD (Table 1). BART achieves a substantially higher F1 score of 49.15%, an increase of 5.5 F1 points over UD, and 3.5 F1 points above Enhanced UD. It does so by substantially improving recall while somewhat decreasing precision.

We also consider *economy*: the number of different patterns needed to achieve a given recall level. Figure 3 plots the achieved recall against the number of patterns. As the curves show, Enhanced UD is more economic than UD, and our representation is substantially more economic than both. To achieve 30.7% recall (the maximal recall of UD), UD requires 112 patterns, EUD requires 77 patterns, while BART needed only 52 patterns.

6 Conclusion

We propose a syntax-based representation that aims to make the event structure and as many lexical relations as possible explicit, for the benefit of downstream information-seeking applications. We provide a Python API that converts UD trees to this representation, and demonstrate its empirical benefits on a relation extraction task.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Marie Candito, Bruno Guillaume, Guy Perrier, and Djamé Seddah. 2017. Enhanced UD dependencies with neutralized diathesis alternation. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 42–53, Pisa,Italy. Linköping University Electronic Press.

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

Benjamin Durme and Lenhart Schubert. 2008. Open knowledge extraction through compositional language processing. *Proceedings of the 2008 Conference on Semantics in Text Processing*.

Angus Forbes, Kristine Lee, Gus Hahn-Powell, Marco A. Valenzuela-Escrcega, and Mihai Surdeanu. 2018. Text annotation graphs: Annotating complex natural language phenomena. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing universal dependency treebanks: A case study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107, Brussels, Belgium. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Siva Reddy, Oscar Tckstrm, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing.

Rachel Rudinger and Benjamin Van Durme. 2014. Is the Stanford dependency representation semantic? In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 54–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Sebastian Schuster and Christopher D Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378.

Vered Shwartz and Chris Waterson. 2018. Olive oil is made *of* olives, baby oil is made *for* babies: Interpreting noun compounds using paraphrases in a neural model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 218–224, New Orleans, Louisiana. Association for Computational Linguistics.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *arXiv preprint arXiv:1603.01648*.

Zeno Vendler. 1957. Verbs and times. *The Philosophical Review*, 66(2):143–160.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia, LDC2006T06*.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Dian Yu, Heng Ji, Sujian Li, and Chin-Yew Lin. 2015. Why read if you can scan? trigger scoping strategy for biographical fact extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1203–1208, Denver, Colorado. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

A   Appendix

A.1   Additional examples of BART structures

The following are additional examples that we could not fit into the space constraints of the paper.

Complement control   Example (1) shows an example of linking the external subject to a controlled finite verb. The following complementary example shows linking the subject to a controlled gerund:

(18)
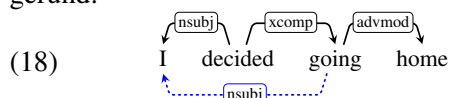
Elaboration/Specification Clauses   Example (7) shows a specification clause connected as an object to the root. The following is a complementary example of using the *like* elaboration-preposition, in which the modifier noun is a *subject* dependant of its head. The modified noun inherits the *subject* relation from its modifier head.
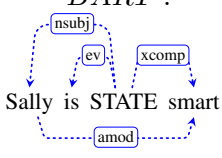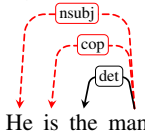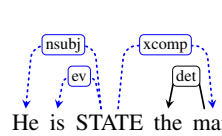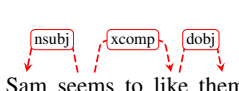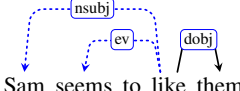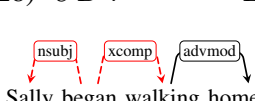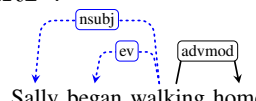
(19)

Adjectival modifier

(20)
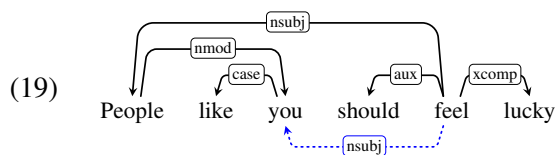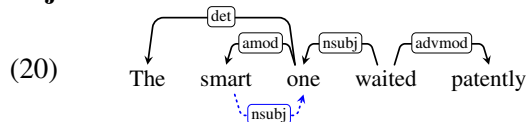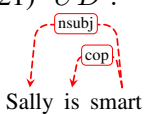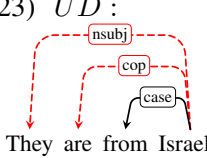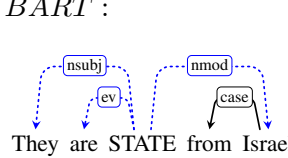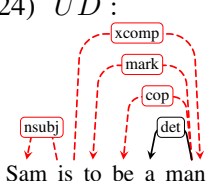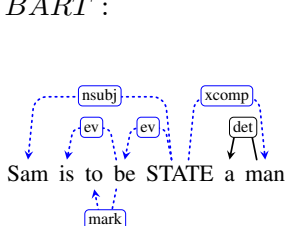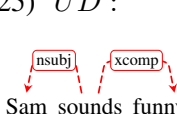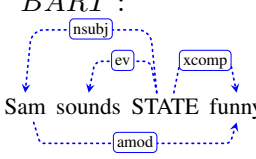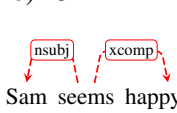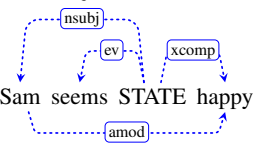
Copular Sentences and Stative Predicates   We show additional examples of these transformations, with explicit comparison to UD.

(21) $UD:$ $BART:$

Sally is smart   Sally is STATE smart
(nsubj, cop)   (nsubj, ev, xcomp, amod)

(22) $UD:$ $BART:$

He is the man   He is STATE the man
(nsubj, cop, det)   (nsubj, ev, xcomp, det)

(23) $UD:$ $BART:$

They are from Israel   They are STATE from Israel
(nsubj, cop, case)   (nsubj, ev, nmod, case)

(24) $UD:$ $BART:$

Sam is to be a man   Sam is to be STATE a man
(xcomp, mark, cop, nsubj, det)   (nsubj, ev, ev, xcomp, det, mark)

(25) $UD:$ $BART:$

Sam sounds funny   Sam sounds STATE funny
(nsubj, xcomp)   (nsubj, ev, xcomp, amod)

(26) $UD:$ $BART:$

Sam seems happy   Sam seems STATE happy
(nsubj, xcomp)   (nsubj, ev, xcomp, amod)

(27) $UD:$ $BART:$

Sam seems to like them   Sam seems to like them
(nsubj, xcomp, dobj)   (nsubj, ev, dobj)

(28) $UD:$ $BART:$

Sally began walking home   Sally began walking home
(nsubj, xcomp, advmod)   (nsubj, ev, advmod)