# 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. As they were designed to accurately reflect syntactic relations, and not to make semantic relations explicit, these syntactic representations lack many explicit connections between content words that are needed for NLP applications. Proposals like 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 provide an easy to use and 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

The introduction of neural-network models into NLP brought with it a substantial increase in dependency parsing accuracy. We can now produce accurate syntactically annotated corpora at scale. However, the use of dependency structures remains limited. Basic syntactic dependency trees encode the functional connections between words but lack many relations between content words. Moreover, the use of strictly-syntactic relations result in structural diversity that undermines the efforts to effectively extract semantic information. "Users" of syntactic trees are thus required to devote substantial efforts to the processing of the trees in order to identify and extract the information needed for 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 by syntax. Schuster and Manning (2016) provide further enhancements targeted at English (Enhanced UD).[2] Candito et al. (2017) suggest even more, additional, enhancements.[3]

In this work we continue this line of thought, and take it a step further. We provide an easy to use Python library that converts English UD trees to a new representation which 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).

[1] universaldepdenencies.org

[2] While 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] The PropS (Stanovsky et al., 2016) and PredPatt White et al. (2016) efforts share a similar motivation, though outside of the UD framework.

[4] In a few examples, we omitted some of the preserved relations for readability.

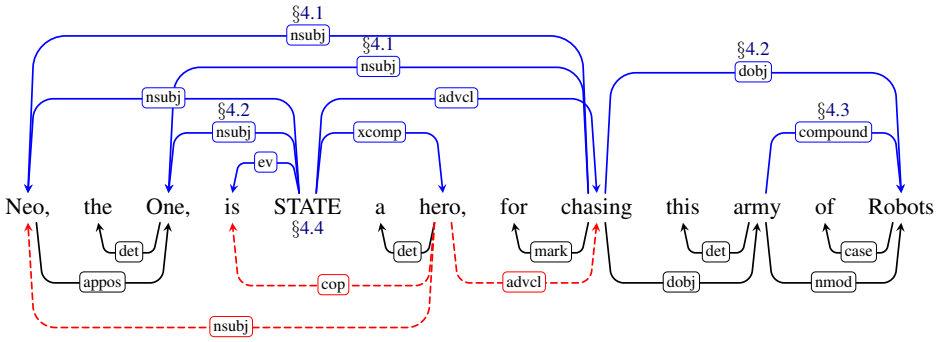



Figure 1: Representation of *Neo, the One, is a hero, for chasing this army of Robots*. The arcs above the sentence are BART additions. The ones below are EUD. Red arcs are removed in BART and black are retained.

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 integrates with the spaCy[5] library, and will be released as open-source under an Apache license. A web-based demo for experimenting with the converter is available at https://nlp.biu.ac.il/~aryeht/eud/.

## 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 favoring **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 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 sharing uncertain information that might be useful. We chose to explicitly expose also some relations which we believe to be useful but judge to be *uncertain*, yet 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 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, (Ntroduction, 2005)), which annotates relations and events, and a QA-SRL dataset (FitzGerald et al., 2018) which connects predicates to sentence segments that are perceived by people as their (possibly implied) arguments. For each of these corpora, we consider the de-

[5] https://spacy.io

pendency 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).

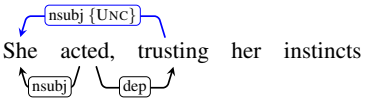
An innovative aspect of our approach is that each edge is associated with additional information beyond its dependency label. This information is structured as follows:

**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)}).

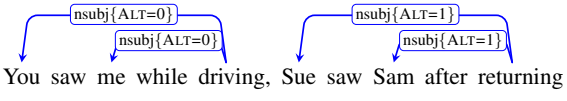
**UNC, ALT**: optional fields indicating uncertainty, described below.

### 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:




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.




```python
# Load a UD-based english model
nlp = spacy.load("en_ud_model")

# Add BART converter to spaCy's pipeline
from pybart.api import converter
converter = converter( ... )
nlp.add_pipe(converter, name="BART")

# Test the new converter component
doc = nlp("He saw me while driving")
me_token = doc[2]
for par_tok in me_token._.parent_list:
    print(par_tok)

# Output:
{'head': 2, 'rel':'dobj', 'src':'UD'}
{'head': 5, 'rel': 'nsubj',
  'src':('advcl','while'), 'alt':'0'}
```

Figure 2: Usage example of pyBART's spaCy-pipeline component.

## 3 Python code and Web-demo

The pyBART library provides a python converter from English UD trees to BART .[6] pyBART subsumes the enhancements of the EUD Java implementation provided in Stanford Core-NLP,[7] and extends them as described in §4. While the default behavior performs all enhancements, the converter is configurable to allow more selective behavior. pyBART has two modes: (1) a converter from CoNLLU-formatted UD trees to CoNLLU-formatted BART structures;[8] and (2) a spaCy (Honnibal and Montani, 2017) pipeline component.[9] After registering pyBART as a spaCy pipeline, tokens on the analyzed document will have a `._.parent_list` field, containing the 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, visualize them, and compares their outputs is also provided, and is available online at https://nlp.biu.ac.il/~aryeht/eud/. The dependency graph visualization component is using TextAnnotationGraphs (TAG) (Forbes et al., 2018).

[6] https://github.com/allenai/ud2ude

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

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

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

## 4 Coverage of Linguistic Phenomena

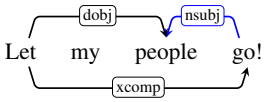
Our resulting conversion to BART consists of four conceptual changes from basic UD. The first type includes propagating shared arguments between external predicates and **nested structures**. The second type includes explicitly sharing arguments between **parallel structures**. The third type has to do with unifying **syntactic alternations** to reduce diversity, making structures that carry similar meaning also similar in structure. Finally, the forth type has to do with making **event structure explicit** in the syntactic representation, allowing main verbs that indicate event properties act as event modifiers rather than heads. In accordance with that, we further introduce a new STATE node, that acts as the main predicate node for *stative* (copular) sentences.

### 4.1 Nested Structures

In this type of conversions, we propagate 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 in *xcomp* clauses containing a *to* marker. We embrace this choice and extend it to cover also clauses without a marker including clauses with controlled gerunds.

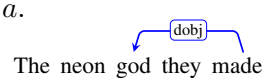
(1)



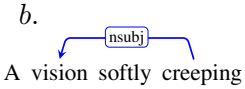

**Noun-modifying clauses:** Similarly, EUD links the empty subject of a finite relative clause to the corresponding argument of the external clause. We extend behavior also to **reduced relative clauses** (2a) and we follow Candito et al. (2017) in also including other relative clauses such as **noun-modifying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

(2)

*a*.



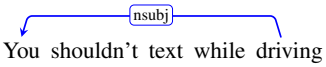

*b*.




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

[10] In clauses that includes an object, ambiguity may arise as to which participant is under modification. We propagate both the subject and the object, and mark the edges as alternative (ALT, see (§2.3)).

(3)



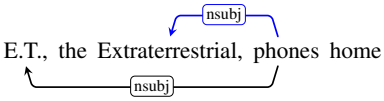

We observe empirically that many of the **dep** edges behave similarly to the adverbial clauses. We treat them similarly, while marking the resulting edges as uncertain.

### 4.2 Parallel structures

In this type of conversion, we identify parallel structures in which the latter instance is elliptical, and share with it the missing core argument contributed by the former instance.

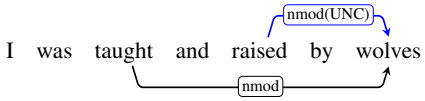
**Apposition:** Similarly to the PropS proposal (Stanovsky et al., 2016), we share relations across *apposition parts*, making the two, currently hierarchical, parts, more duplicate-like.

(4)

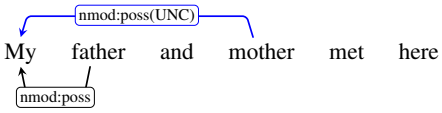



**Modifiers in conjunction:** In modified coordinated constructions, we share prepositional (5) and possessive (6) modifiers between the coordinated parts.

(5)



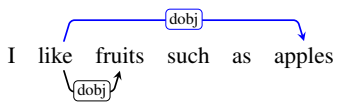

(6)



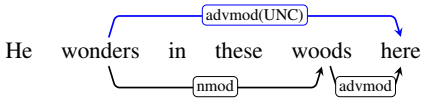

Since dependency structures are *inherently* ambiguous between conjoined modification and single-conjunt modification, (e.g, compare (5) to "Mogly was lost and raised by wolves", or (6) to "my Father and E.T."), we mark both as UNC.

**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.

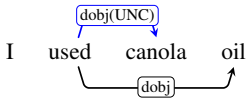
(7)




**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) 


**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.

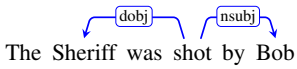
(9) 


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.

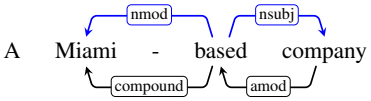
### 4.3 Syntactic Alternations

With this type of conversions we aim to unify syntactic variability. That is to say, we identify structures that are syntactically different but share (some) semantic structure, and add arcs or nodes that will expose the similarity. We detail on the particular phenomena we unify in turn.

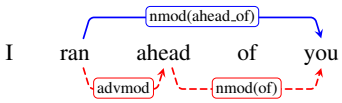
**The Passivization Alternation:** Following Candito et al. (2017) we relate the *passive* alteration to its *active* variant.

(10) 


**Hyphen reconstruction:** Noun-verb Hyphen Constructions (HC) which are modifying a nominal are effectively equivalent to a copular sentence wherein the subject is that nominal and the verb-part of the HC 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) 

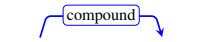

**Multi-word prepositions:** EUD transforms some adverbial cases to multi-word prepositions, using a white-list. We identify that we can do this systematically by considering all cases of advmod immediately followed by a preposition, and using a black-list for obvious verb-participle constructions (such as "back to" which in "going back to work" is modifying the verb).

(12) 


**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 "the apple is green"). To explicitly capture this productive implication, we add a subject relation from each adjectival modifier to its corresponding modified noun.

(13) 

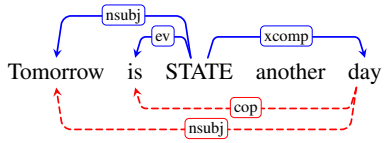

**Genitive Constructions:** Genitive cases can be alternatively expresses as a compound. We add a compound relation to unify the expression of genitives across these *X of Y* and *compound* structures.

(14) 


### 4.4 Event-Centered Representations

In sentences like "He started working", "He seems to be working", the main event indicated by the sentence is "work", while the syntactic head and root predicate is different. We present a chain of changes that aim to put emphasis on events by delegating copulas (is, was), evidentials (seem, say) and aspectuality (started, continued) to be clausal modifiers of events, rather than heads. 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 the terminology of Vendler's classes (Vendler, 1957)). We remedy this by adding a node that represents that STATE and allows for tense, aspect, modality and evidentiality information to be directly modifying it.

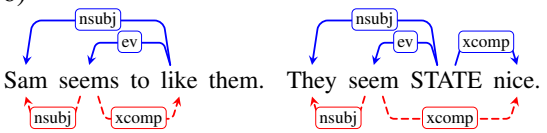
**Copular Sentences and Stative Predicates:** We added to all copula constructions new node named *STATE*, which represents the *stative* event introduced by the copular clause. This node becomes the root, and we rewire the entire clause around this *STATE*. By doing so we normalize the structure with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

(15) 

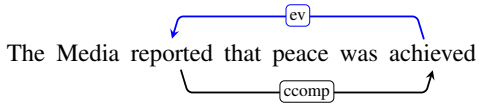

**Evidential reconstructions:** The previous alternation provides us with the opportunity to explicitly mark properties of this *stative* event as de-

pendent of the root. All edges that mark event properties are labeled as *ev*. We do so, for instance, for verbs explicitly marking evidentiality (16). We further expand this *ev* relation to mark evidentiality for reported-speech (17).[11]

(16)

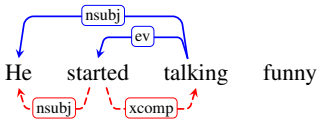



(17)




**Aspectual constructions:** we can now also mark aspectual verbs as modifying the complement verb. The complement (matrix) verb becomes the root, and we add the new *ev* relation to the various event-properties of the verb.

(18)




## 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 focus on pattern-based RE rather than on ML-based methods, because we can not distinguish if the representation is what being measured and not the overlying ML-model.

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).

[11] In all these, we follow white lists of evidential/reported-speech/aspectual verbs.

| 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.

We use this procedure to compare UD, Enhanced UD, our representation without Enhanced-UD's enhancements and our complete representation which is a superset of Enhanced UD (Table 1). Our representations 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.

An additional metric we consider is *economy*: how many different patterns are needed to achieve a given recall level (lower is better)? 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. Achieving 30.7% recall (the maximal recall of UD), requires 112 UD patterns, 77 Enhanced UD patterns, and only 52 BART patterns.

## 6 Conclusion

We propose a syntax-based representations 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*

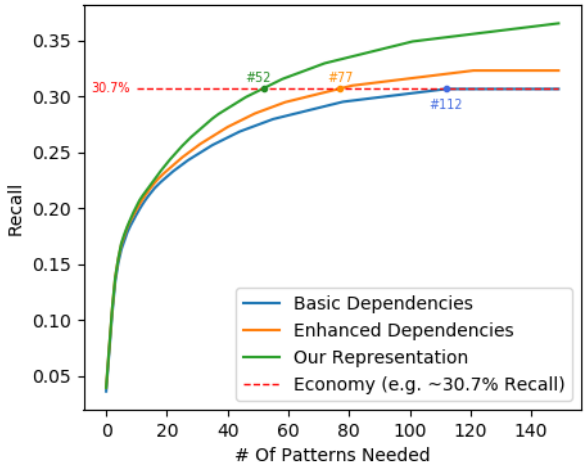



Figure 3: Economy comparison: Recall vs number of patterns, for the different representations.

*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.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale QA-SRL parsing. *CoRR*, abs/1805.05377.

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).

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.

Ii. I Ntroduction. 2005. The ace 2005 ( ace 05 ) evaluation plan evaluation of the detection and recognition of ace entities , values , temporal expressions , relations , and events 1.

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.

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.

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.