Universal Natural Language Understanding:
Combining Universal Dependencies, Glue Semantics, and machine learning

Jamie Findlay and Dag Haug

University of Oslo

Helgaker, 4 May 2022

Universal natural language understanding

- *Universal natural language understanding* is an (ambitiously named) RCN-funded project on computational semantics
- Four people involved:
  - Jamie Findlay (postdoc)
  - Dag Haug (PI)
  - Ahmet Yıldırım (programmer)
  - Saeedeh Salimifar (PhD fellow)

Formal and computational semantics

- 30 years ago, computational semantics would just be formal semantics implemented on a computer.
- Now the fields have drifted apart and the connections are often tangential.
- So why care about combining the two?
  - From the computational side, *natural language inference* is still an unsolved problem and machine learned models are often wrong, or at best right for the wrong reasons.
  - From the linguistic side, we get the chance to study large-scale rule systems and discover unexpected interactions.
- Also, we discover how much we take for granted as input to the semantics:
  - detailed syntax trees/LFs complete with empty categories, coindexation patterns, etc., etc.
  - rich information about lexical items (particularly for function words)

Natural language inference

- An in-demand NLP task: given a text, what follows?
  - Especially important for question-answering systems.

(1) The high price of corn affected the profits.
$\Rightarrow$ The price of corn is high.

- Can be straightforward given an appropriate logical representation of the premise:

(2) $\mathit{price}(x) \wedge \mathit{of}(x, y) \wedge \mathit{corn}(y) \wedge \mathit{high}(x) \wedge \mathit{profit}^*(z) \wedge \mathit{affect}(e, x, z) \ldots$

- But purely logical entailments are rare – we often need access to e.g. lexical information (Bos & Markert, 2005).

Natural language inference

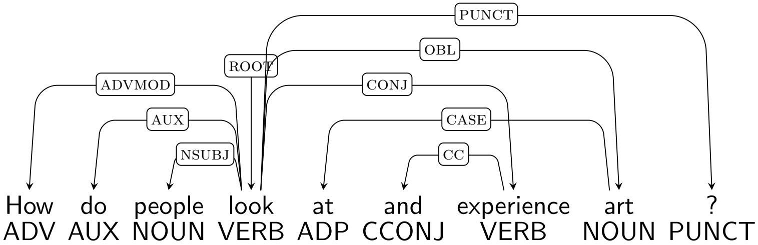
- Modern computational approaches treat NLI as a classification problem over sentence pairs, without explicit modelling of semantic structure (e.g. Bowman et al. 2015; Jiang & de Marneffe 2019).

- Results can be good, but seem often to rely on quirks of the datasets to extract incorrect generalisations that are effective, but brittle and easily misled (e.g. by passivisation: see McCoy et al. 2019)
  $\rightarrow$ a combination of logical and statistical/neural methods

- An interesting approach is to teach neural nets to produce DRSs (van Noord et al., 2018).

- Our approach is to use a shallow, machine-learned syntactic representation, use rule-based methods to turn that into a DRS, and then enrich the result using machine-learning again.

What you get from computational syntax

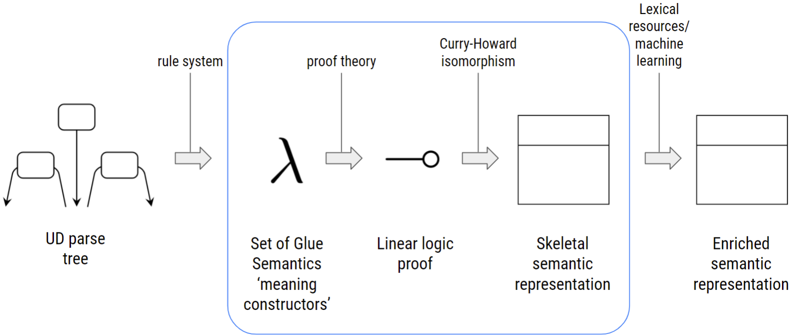


- This is a shallow syntactic representation, which is hard to convert into a meaningful semantics.
- On the plus side, structures like this are available for 100+ languages
  - via the Universal Dependencies project (Nivre et al., 2016, 2020).
  - *Enhanced* Universal Dependencies add more semantically-relevant information, but have their limitations (Findlay & Haug, 2021).

→ interest in deriving semantic representations from UD structures, ideally in a language-independent way.

A universal pipeline

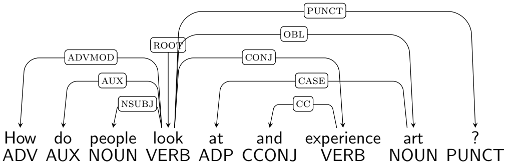
- For some languages there is detailed lexical information from VerbNets, WordNets, FrameNets etc., but for others there isn't.
  - So, for our universal pipeline, we postpone the use of lexical information as long as possible.



Target representations

- Our target representations for sentence meanings are DRSs.
- In particular, we have chosen the format produce by Boxer (Bos, 2008) in the Parallel Meaning bank.

From UD to DRT



$e1\ e2\ {}^{*}t1\ {}^{*}t2\ x1\ x2\ x3$
$(x3 = \text{‘?‘})$
$Manner(e1, x3)$
$Manner(e2, x3)$
$manner.n.01(x3)$
$Time(e1, t1)$
$Time(e2, t2)$
$person.n.01(x1)$
$Agent(e1, x1)$
$Theme(e1, x2)$
$look\_at.v.02(e1)$
$Experiencer(e2, x1)$
$Stimulus(e2, x2)$
$experience.v.02(e2)$
$art.n.01(x2)$
$(t2 = \text{‘}now\text{‘})$
${}^{*}time.n.08(t2)$
$(t1 = \text{‘}now\text{‘})$
${}^{*}time.n.08(t1)$

Challenges

- Fundamental problems:
  - We do not have meanings for the words.
  - We do not know how to compose the words, since the tree is too flat.
- Solution: We map the words to Glue Semantics entries (*meaning constructors*), which consist of two things:
  - a meaning
  - a type, which specifies how the meaning combines with others

(3) $$\textit{John} \rightsquigarrow \lambda P.\begin{array}{|c|}\hline x \\ \hline \mathit{Name}(x, \mathit{John}) \\ \hline\end{array} + P(x) : (e_1 \multimap t_2) \multimap t_2$$

- Right-hand side:
  - the types of ordinary Montague grammar
  - plus indices to keep track of what combines with what

In this case 1 is the index of *John* and 2 is the index of its scope point.

How good does it get?

- The entry from *John* is predictable from the POS-tag + the tree topography (scope point is the root verb).
- Similarly, we can provide reasonable entries for common nouns, adjectives, verbs, manner adverbs.
- They won't be detailed but also not worse than $[\![\mathit{life}]\!] = \mathit{life}'$.
- The challenge lies in functional items and relations, and in constructional aspects of meaning.
- Two main approaches:
  - limited use of lemma-specific meanings for closed-class "logic words" (e.g. determiners, connectives etc.)
  - pass on the problem to post-processing with machine learning (e.g. the mapping from syntactic relations to semantic roles, multiword expressions that are not annotated as such in UD)
- Before we look at the details we need a little bit of Glue.

What is Glue? (Background)

- A theory of the syntax/semantics interface, originally developed for LFG (Dalrymple et al., 1993, 1999), and now the *de facto* standard there (Dalrymple et al., 2019, ch. 8.5).
  - (See Findlay 2021, sec. 3 for a survey of approaches to meaning representation in LFG.)
- Also combined with other frameworks:
  - HPSG (Asudeh & Crouch, 2002),
  - LTAG (Frank & van Genabith, 2001), and
  - Minimalism (Gotham, 2018).

What is Glue? (Theory)

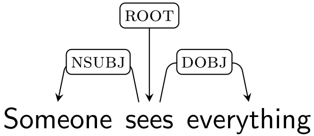
- Expressions in some meaning language are paired with linear logic formulae (Girard, 1987) as *meaning constructors*.
- You can think of the linear logic as providing a categorial grammar/Lambek calculus over meanings (ignoring word order).
- Technically, semantic composition is linear logic deduction mediated by the Curry-Howard isomorphism (Curry & Feys, 1958; Howard, 1980).
- Linear logic deduction is a computationally well-understood process (Hepple, 1996; Lev, 2007) with an efficient implementation (Messmer & Zymla, 2018) that we have also contributed to.

Scope ambiguity as an example

(4) Someone sees everything.

- Two interpretations:
  1. There is someone who sees everything. (surface scope, $\exists > \forall$)
  2. Everything is seen. (inverse scope, $\forall > \exists$)

- This if often seen as a syntactic ambiguity, but the dependency tree is compatible with both interpretations because there is no VP constituent:



- Syntax *constrains* what combines with what, but doesn't *determine* it.

What you need from syntax

LABEL	e_1	e_3	t_2
ASSIGNED TO	the subject argument of *sees*	the object argument of *sees*	the sentence as a whole
	someone	*everything*	

$\Downarrow$

$\lambda v.\lambda u.[\ |see(u,v)] : e_3 \multimap (e_1 \multimap t_2)$ — type $e{\rightarrow}(e{\rightarrow}t)$

$\lambda Q.[x_1 \mid person(x_1)]; Q(x_1) : (e_1 \multimap t_2) \multimap t_2$ — type $(e{\rightarrow}t){\rightarrow}t$

$\lambda P.[\ |[x_1|\ ] \Rightarrow P(x_1)] : (e_3 \multimap t_2) \multimap t_2$ — type $(e{\rightarrow}t){\rightarrow}t$

- An informal explanation:
  - $[\![\textit{sees}]\!]$ applies to e_3, then applies to e_1, to form t_2.
  - $[\![\textit{someone}]\!]$ applies to (something that applies to e_1 to form t_2) to form t_2.
  - $[\![\textit{everything}]\!]$ applies to (something that applies to e_3 to form t_2) to form t_2.
- There's more than one way to put $[\![\textit{someone}]\!]$, $[\![\textit{sees}]\!]$ and $[\![\textit{everything}]\!]$ together, while obeying these constraints, to form t_2.
- The different ways:
  - Give the different interpretations of (4).
  - Correspond to different proofs from the same premises in linear logic.

Surface scope interpretation

$$
\dfrac{\begin{array}{c}[\![\mathit{someone}]\!] :\\ (e_1 \multimap t_2) \multimap t_2\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}[\![\mathit{everything}]\!] :\\ (e_3 \multimap t_2) \multimap t_2\end{array} \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}[\![\mathit{sees}]\!] :\\ e_3 \multimap (e_1 \multimap t_2)\end{array} \quad [z : e_3]^1}{[\![\mathit{sees}]\!](z) : e_1 \multimap t_2}\multimap_E \qquad [w : e_1]^2}{[\![\mathit{sees}]\!](z)(w) : t_2}\multimap_E}{\lambda z.[\![\mathit{sees}]\!](z)(w) : e_3 \multimap t_2}\multimap_{I,1}}{[\![\mathit{everything}]\!](\lambda z.[\![\mathit{sees}]\!](z)(w)) : t_2}\multimap_E}{\lambda w.[\![\mathit{everything}]\!](\lambda z.[\![\mathit{sees}]\!](z)(w)) : e_1 \multimap t_2}\multimap_{I,2}}{[\![\mathit{someone}]\!](\lambda w.[\![\mathit{everything}]\!](\lambda z.[\![\mathit{sees}]\!](z)(w))) : t_2}\multimap_E
$$

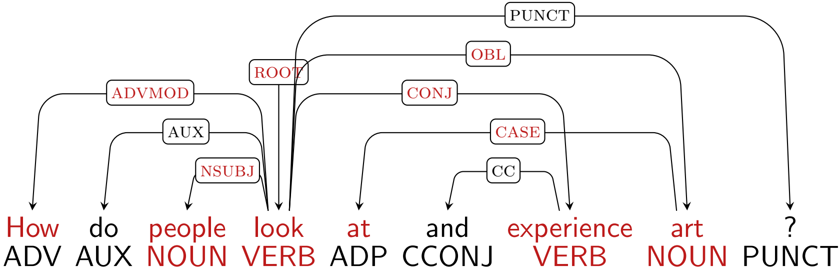
$$
\rightsquigarrow_\beta \begin{array}{|l|}\hline x_1 \\ \hline \mathit{person}(x_1) \\ \begin{array}{|c|}\hline x_2 \\ \hline \\ \hline\end{array} \Rightarrow \begin{array}{|c|}\hline \\ \hline \mathit{see}(x_1, x_2) \\ \hline\end{array} \\ \hline\end{array}
$$

Inverse scope interpretation

$$\dfrac{[\![everything]\!] : (e_3 \multimap t_2) \multimap t_2 \qquad \dfrac{\dfrac{[\![someone]\!] : (e_1 \multimap t_2) \multimap t_2 \qquad \dfrac{[\![sees]\!] : e_3 \multimap (e_1 \multimap t_2) \qquad [z : e_3]^1}{[\![sees]\!](z) : e_1 \multimap t_2}\multimap_E}{[\![someone]\!]([\![sees]\!](z)) : t_2}\multimap_E}{\lambda z.[\![someone]\!]([\![sees]\!](z)) : e_3 \multimap t_2}\multimap_{I,1}}{[\![everything]\!](\lambda z.[\![someone]\!]([\![sees]\!](z))) : t_2}\multimap_E$$

$$\rightsquigarrow_\beta \begin{array}{|c|}\hline \\ \hline \begin{array}{|c|}\hline x_1 \\ \hline \\ \hline\end{array} \Rightarrow \begin{array}{|l|}\hline x_2 \\ \hline person(x_2) \\ see(x_2, x_1) \\ \hline\end{array} \\ \hline\end{array}$$

A worked example



$$\lambda V.\lambda F.V(\lambda E(([X],[(X = \text{‘?‘}), advmod(E,X)]) + F(E))) :$$
$$(x_4 \multimap x_4)$$

$$\lambda Q.\lambda V.\lambda F.(Q(\lambda X.(V(\lambda E.(([\,],[nsubj(E,X)]) + F(E)))))) :$$
$$(((e_3 \multimap p_4) \multimap p_4) \multimap (x_4 \multimap x_4))$$

$$\lambda X.([\,],[person(X)]) :$$

How far did we get?

e_1 e_2 x_1 x_2 x_3
$person(x_1)$
$art(x_2)$
$look(e_1)$
$experience(e_2)$
$at(e_1, x_2)$
$(x_3 = \text{'?'})$
$advmod(e_1, x_3)$
$nsubj(e_1, x_1)$

Missing information

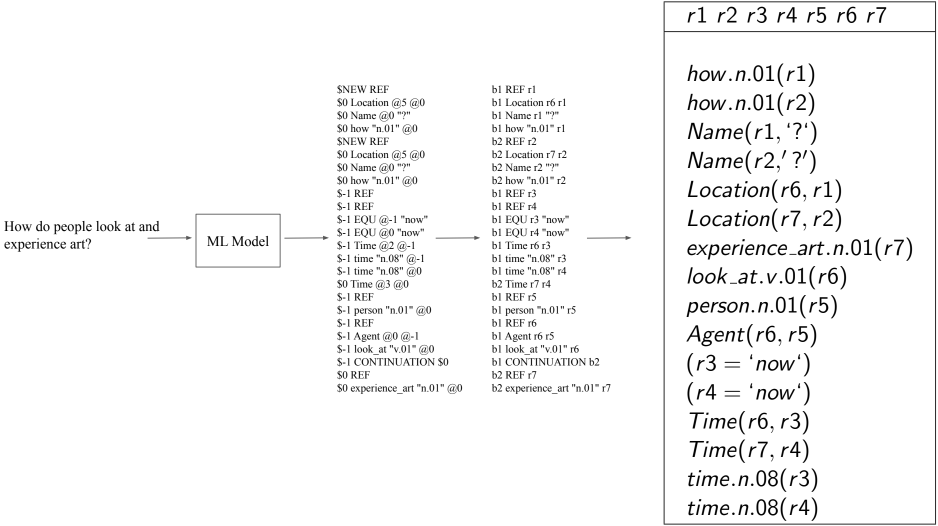
- "look at" as MWE
- purely syntactic roles
- no time information
- no roles for second conjunct

$e1$ $e2$ $^*t1$ $^*t2$ $x1$ $x2$ $x3$
$(x3 = \text{'?'})$
$Manner(e1, x3)$
$Manner(e2, x3)$
$manner.n.01(x3)$
$Time(e1, t1)$
$Time(e2, t2)$
$person.n.01(x1)$
$Agent(e1, x1)$
$Theme(e1, x2)$
$look\_at.v.02(e1)$
$Experiencer(e2, x1)$
$Stimulus(e1, x2)$
$experience.v.02(e2)$
$art.n.01(x2)$
$(t2 = \text{'}now\text{'})$
$^*time.n.08(t2)$
$(t1 = \text{'}now\text{'})$
$^*time.n.08(t1)$

Using machine learning (ML)

- Machine learning learns patterns from data
- Latest deep learning models enable us to train a model to obtain DRT of a sentence
- Training an ML model for this task is very similar to training an ML model for machine translation (i.e. translating an English sentence to Norwegian)
- Doing this requires many examples of sentence-DRT pairs.
- The model extract patterns from the pairs
- When we use the model, we show the model a sentence and based on the knowledge of previous examples in training, it produces a DRT for the sentence

Using machine learning - Example



$r1\ r2\ r3\ r4\ r5\ r6\ r7$
$how.n.01(r1)$
$how.n.01(r2)$
$Name(r1, \text{'?'})$
$Name(r2, \text{'?'})$
$Location(r6, r1)$
$Location(r7, r2)$
$experience\_art.n.01(r7)$
$look\_at.v.01(r6)$
$person.n.01(r5)$
$Agent(r6, r5)$
$(r3 = \text{'}now\text{'})$
$(r4 = \text{'}now\text{'})$
$Time(r6, r3)$
$Time(r7, r4)$
$time.n.08(r3)$
$time.n.08(r4)$

$e1\ e2\ {}^{*}t1\ {}^{*}t2\ x1\ x2\ x3$
$(x3 = \text{'?'})$
$Manner(e1, x3)$
$Manner(e2, x3)$
$manner.n.01(x3)$
$Time(e1, t1)$
$Time(e2, t2)$
$person.n.01(x1)$
$Agent(e1, x1)$
$Theme(e1, x2)$
$look\_at.v.02(e1)$
$Experiencer(e2, x1)$
$Stimulus(e1, x2)$
$experience.v.02(e2)$
$art.n.01(x2)$
$(t2 = \text{'}now\text{'})$
${}^{*}time.n.08(t2)$
$(t1 = \text{'}now\text{'})$
${}^{*}time.n.08(t1)$

What's next in the rules system

- We have basic coverage of adjuncts and arguments, including clausal complementation and relativization
- Coordination is covered for the most important categories (nouns, verbs, adjectives)
- We now need to look at harder phenomena, including various sorts of "constructions" given ad hoc analyses in UD
  - ellipsis
  - resumption
  - multiword expressions
  - ...
- Basic tense coverage should also be doable

What the rules won't cover

$x_1 \;\; e_1$
$named(x_1, Peter)$ $arrive(e_1)$ $theme(e_1, x_1)$

- What kind of θ-role is 'nsubj'?
  - A syntactic name, lifted from the arc label.
  - In and of itself, uninformative.
- What we have in the DRS above is as much information as can be extracted from the UD tree alone, without lexical knowledge.
- The theoretical challenges do not lie in depth of analysis, but in universality of coverage
- To get further, we either need reliable lexical databases, or use ML to learn patterns in the assignment of thematic roles
- A similar approach would be useful to deal with e.g. presupposition

How to combine with ML

- Pointwise enrichment of conditions?
- Or feed structured data into the ML encoder–decode set up?
- But how to present hierarchical data to neural nets?

how	$(x_3 = \text{'?'})$
	$\mathit{advmod}(e_1, x_3)$
people	$\mathit{person}(x_1)$
	$\mathit{nsubj}(e_1, x_1)$
look	$\mathit{look}(e_1)$
and	
experience	$\mathit{experience}(e_2)$
art	$\mathit{art}(x_2)$
at	$\mathit{at}(e_1, x_2)$

Conclusions

- NLI is a big unsolved problem in NLP, both for logic-based and ML-based approaches
- Our approach is primarily logic-based, but is based on a shallow syntax which is available for more languages than e.g. CCG parsers that previous logic-based approaches use
- Interesting to see how wide coverage we can achieve in this setup
- Interaction with ML definitely needed for post-processing

References I

Asudeh, Ash & Richard Crouch. 2002. Glue Semantics for HPSG. In Frank van Eynde, Lars Hellan & Dorothee Beermann (eds.), *Proceedings of the 8th international HPSG conference*, Stanford, CA: CSLI Publications.

Bos, Johan. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 conference on semantics in text processing* STEP '08, 277–286. Stroudsburg, PA, USA: Association for Computational Linguistics. `http://dl.acm.org/citation.cfm?id=1626481.1626503`.

Bos, Johan & Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 628–635. Vancouver, British Columbia, Canada: Association for Computational Linguistics. `https://aclanthology.org/H05-1079`.

References II

Bowman, Samuel R., Gabor Angeli, Christopher Potts & Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 632–642. Lisbon, Portugal: Association for Computational Linguistics. doi:10.18653/v1/D15-1075. `https://aclanthology.org/D15-1075`.

Curry, Haskell B. & Robert Feys. 1958. *Combinatory logic: volume I*. Amsterdam: North Holland.

Dalrymple, Mary, Vineet Gupta, John Lamping & Vijay Saraswat. 1999. Relating resource-based semantics to categorial semantics. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar*, 261–280. Cambridge, MA: MIT Press.

References III

Dalrymple, Mary, John Lamping & Vijay Saraswat. 1993. LFG semantics via constraints. In Steven Krauwer, Michael Moortgat & Louis des Tombe (eds.), *EACL 1993*, 97–105.

Dalrymple, Mary, John J. Lowe & Louise Mycock. 2019. *The Oxford reference guide to Lexical Functional Grammar*. Oxford: Oxford University Press.

Findlay, Jamie Y. 2021. Meaning in LFG. In I. Wayan Arka, Ash Asudeh & Tracy Holloway King (eds.), *Modular design of grammar: linguistics on the edge*, 340–374. Oxford University Press.

Findlay, Jamie Y. & Dag T. T. Haug. 2021. How useful are Enhanced Universal Dependencies for semantic interpretation? In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, 22–34. Sofia: Association for Computational Linguistics. `https://aclanthology.org/2021.depling-1.3`.

References IV

Frank, Anette & Josef van Genabith. 2001. GlueTag: Linear logic-based semantics for LTAG—and what it teaches us about LFG and LTAG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG01 conference*, Stanford, CA: CSLI Publications.

Girard, Jean-Yves. 1987. Linear logic. *Theoretical Computer Science* 50(1). 1–101. doi:10.1016/0304-3975(87)90045-4.

Gotham, Matthew. 2018. Making Logical Form type-logical. *Linguistics and Philosophy* 41(5). 511–556. doi:10.1007/s10988-018-9229-z.

Hepple, Mark. 1996. A compilation-chart method for linear categorial deduction. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, .

Howard, W.A. 1980. The formulae-as-types notion of construction. In J.P. Seldin & J.R. Hindley (eds.), *To H.B. Curry*, 479–490. New York: Academic Press.

References V

Jiang, Nanjiang & Marie-Catherine de Marneffe. 2019. Evaluating BERT for natural language inference: A case study on the CommitmentBank. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6086–6091. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1630. https://aclanthology.org/D19-1630.

Lev, Iddo. 2007. *Packed computation of exact meaning representations*: Stanford University dissertation.

McCoy, R. Thomas, Ellie Pavlick & Tal Linzen. 2019. Right for the wrong reasons: diagnosing syntactic heuristics in natural language inference. In *proceedings of the 57th annual meeting of the association for computational linguistics*, .

References VI

Messmer, Moritz & Mark-Matthias Zymla. 2018. The glue semantics workbench: A modular toolkit for exploring linear logic and glue semantics. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the lfg'18 conference, university of vienna*, 249–263. Stanford, CA: CSLI Publications. `http://cslipublications.stanford.edu/LFG/2018/lfg2018-messmer-zymla.pdf`.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty & Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 1659–1666. Portorož,

References VII

SI: European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2016/pdf/348_Paper.pdf.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers & Daniel Zeman. 2020. Universal Dependencies v2: an evergrowing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, 4034–4043. Marseille: European Language Resources Association. https://aclanthology.org/2020.lrec-1.497.

van Noord, Rik, Lasha Abzianidze, Antonio Toral & Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics* 6. 619–633.