MRP 2020: The Second Shared Task on Cross-Framework and Cross-Lingual Meaning Representation Parsing

Stephan Oepen[♣], Omri Abend[♠], Lasha Abzianidze[♡], Johan Bos[♢], Jan Hajič[◦], Daniel Hershcovich[⋆], Bin Li[•], Tim O'Gorman[⋄], Nianwen Xue[∗], and Daniel Zeman[◦]

[♣] University of Oslo, Department of Informatics

[♠] The Hebrew University of Jerusalem, School of Computer Science and Engineering

[♡] Utrecht University, UiL OTS

[♢] University of Groningen, Center for Language and Cognition

[◦] Charles University, Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

[⋆] University of Copenhagen, Department of Computer Science

[•] Nanjing Normal University, School of Chinese Language and Literature

[⋄] University of Massachusetts at Amherst, College of Information and Computer Sciences

[∗] Brandeis University, Department of Computer Science

mrp-organizers@nlpl.eu

Abstract

The 2020 Shared Task at the Conference for Computational Language Learning (CoNLL) was devoted to Meaning Representation Parsing (MRP) across frameworks and languages. Extending a similar setup from the previous year, five distinct approaches to the representation of sentence meaning in the form of directed graphs were represented in the English training and evaluation data for the task, packaged in a uniform graph abstraction and serialization; for four of these representation frameworks, additional training and evaluation data was provided for one additional language per framework. The task received submissions from eight teams, of which two do not participate in the official ranking because they arrived after the closing deadline or made use of additional training data. All technical information regarding the task, including system submissions, official results, and links to supporting resources and software are available from the task web site at:

http://mrp.nlpl.eu

1 Background and Motivation

The 2020 Conference on Computational Language Learning (CoNLL) hosts a shared task (or 'system bake-off') on Cross-Framework Meaning Representation Parsing (MRP 2020), which is a revised and extended re-run of a similar CoNLL shared task in the preceding year. The goal of these tasks is to advance data-driven parsing into *graph-structured* representations of *sentence meaning*. For the first time, the MRP task series combines *formally* and *linguistically* different approaches to meaning representation in graph form in a uniform training and evaluation setup.

Key differences in the 2020 edition of the task include the addition of a graph-based encoding of Discourse Representation Structures (dubbed DRG); a generalization of Prague Tectogrammatical Graphs (to include more information from the original annotations); and a separate cross-lingual track, introducing one extra language (beyond English) for four of the frameworks involved.[1]

Participants were invited to develop parsing systems that support five distinct semantic graph frameworks in four languages (see §3 below)—all encoding core predicate–argument structure, among other things—in the same implementation. Ideally, these parsers predict sentence-level meaning representations in all frameworks in parallel. Architectures utilizing complementary knowledge sources (e.g. via parameter sharing) were encouraged, though not required. Learning from multiple flavors of meaning representation in tandem has hardly been explored (with notable exceptions, e.g. the parsers of Peng et al., 2017; Hershcovich et al., 2018; Stanovsky and Dagan, 2018; or Lindemann et al., 2019).

The task design aims to reduce framework-specific 'balkanization' in the field of meaning representation parsing. Its contributions include

[1] To reduce the threshold to participation, two of the target frameworks represented in MRP 2019 are not in focus this year, viz. the purely bi-lexical DELPH-IN MRS Bi-Lexical Dependencies and Prague Semantic Dependencies (PSD). These graphs largely overlap with the corresponding (but richer) frameworks in 2020, EDS and PTG, respectively, and the original bi-lexical semantic dependency graphs remain independently available (Oepen et al., 2015).

Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing, pages 1–22
Online, Nov. 19-20, 2020.

(a) a unifying formal model over different semantic graph banks (§2), (b) uniform representations and scoring (§4 and §6), (c) contrastive evaluation across frameworks (§5), and (d) increased cross-fertilization of parsing approaches (§7).

2 Definitions: Graphs and Flavors

Reflecting different traditions and communities, there is wide variation in how individual meaning representation frameworks think (and talk) about semantic graphs, down to the level of visual conventions used in rendering graph structures. Increased terminological uniformity and guidance in how to navigate this rich and diverse landscape are among the desirable side-effects of the MRP task series. The following paragraphs provide semi-formal definitions of core graph-theoretic concepts that can be meaningfully applied across the range of frameworks represented in the shared task.

Basic Terminology Semantic graphs (across different frameworks) can be viewed as directed graphs or *digraphs*. A semantic digraph is a triple (T, N, E) where N is a set of *nodes* and $E \subseteq N \times N$ is a set of *edges*. The *in-* and *out-degree* of a node count the number of edges arriving at or leaving from the node, respectively. In contrast to the unique *root* node in trees, graphs can have multiple (structural) roots, which we define as nodes with in-degree zero. The majority of semantic graphs are structurally multi-rooted. Thus, we distinguish one or several nodes in each graph as *top* nodes, $T \subseteq N$; the top(s) correspond(s) to the most central semantic entities in the graph, usually the main predication(s).

In a tree, every node except the root has in-degree one. In semantic graphs, nodes can have in-degree two or higher (indicating shared arguments), which constitutes a *reentrancy* in the graph. In contrast to trees, general digraphs may contain *cycles*, i.e. a directed path leading from a node to itself. Another central property of trees is that they are *connected*, meaning that there exists an undirected path between any pair of nodes. In contrast, semantic graphs need not generally be connected.

Finally, in some semantic graph frameworks there is a (total) linear order on the nodes, typically (though not necessarily) induced by the surface order of corresponding tokens. Such graphs are conventionally called *bi-lexical dependencies* (probably deriving from a notion of lexicalization articulated by Eisner, 1997) and formally constitute *ordered graphs*. A natural way to visualize a bi-lexical dependency graph is to draw its edges as semicircles in the halfplane above the sentence. An ordered graph is called *noncrossing* if in such a drawing, the semicircles intersect only at their endpoints (this property is a natural generalization of projectivity as it is known from dependency trees).

A natural generalization of the noncrossing property, where one is allowed to also use the halfplane below the sentence for drawing edges is a property called *pagenumber two*. Kuhlmann and Oepen (2016) provide additional definitions and a quantitative summary of various formal graph properties across frameworks.

Hierarchy of Formal Flavors In the context of the MRP shared task series, we have previously defined different *flavors* of semantic graphs based on the nature of the relationship they assume between the linguistic surface signal (typically a written sentence, i.e. a string) and the nodes of the graph (Oepen et al., 2019). We refer to this relation as *anchoring* (of nodes onto sub-strings); other commonly used terms include alignment, correspondence, or lexicalization.

Flavor (0) is characterized by the strongest form of anchoring, obtained in bi-lexical dependency graphs, where graph nodes injectively correspond to surface lexical units (i.e. tokens or 'words'). In such graphs, each node is directly linked to one specific token (conversely, there may be semantically empty tokens), and the nodes inherit the linear order of their corresponding tokens.

Flavor (1) includes a more general form of anchored semantic graphs, characterized by relaxing the correspondence between nodes and tokens, allowing arbitrary parts of the sentence (e.g. sub-token or multi-token sequences) as node anchors, as well as unanchored nodes, or multiple nodes anchored to overlapping sub-strings. These graphs afford greater flexibility in the representation of meaning contributed by, for example, (derivational) affixes or phrasal constructions and facilitate lexical decomposition (e.g. of causatives or comparatives).

Finally, Flavor (2) semantic graphs do not consider the correspondence between nodes and the surface string as part of the representation of meaning (thus backgrounding notions of derivation and compositionality). Such semantic graphs are simply unanchored.

While different flavors refer to formally defined

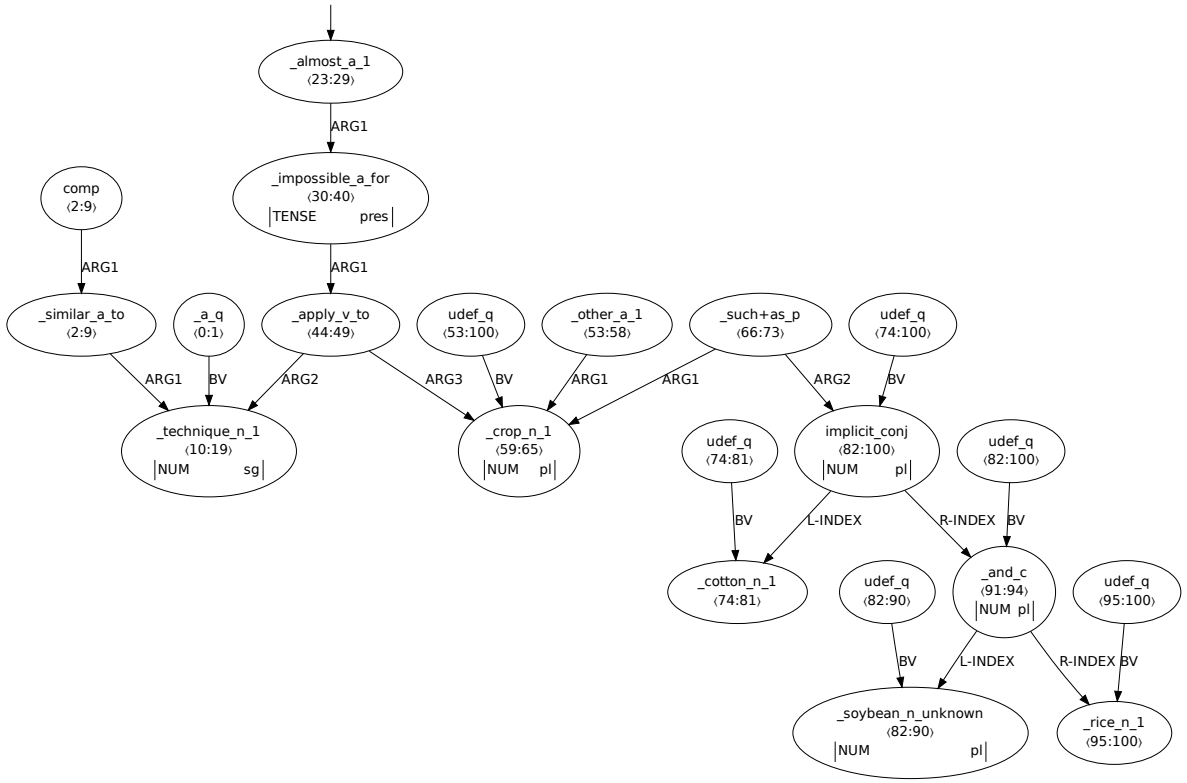

Figure 1: Semantic dependency graphs for the running example *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice*: Elementary Dependency Structures (EDS). Node properties are indicated as two-column records below the node labels.

sub-classes of semantic graphs, we reserve the term *framework* for specific linguistic approaches to graph-based meaning representation (typically encoded in a particular graph flavor, of course). However, the coarse classification into three distinct flavors does not fully account for the variability of anchoring relations observed across frameworks. For example, graphs can be partially anchored, meaning that only a subset of nodes are explicitly linked to the surface string; the anchoring relations that are present, can in turn stand in one-to-one correspondence to surface tokens, or allow overlapping and sub-token or phrasal relationships. At the same time, a framework may impose a total ordering of nodes independent (or possibly only partly dependent) on anchoring. We will interpret Flavors (0) and (2) strictly, as fully lexically anchored and wholly unanchored, respectively, leading to the categorization of mixed forms of anchoring as Flavor (1), and allow for the presence of ordered graphs, in principle at least, at all levels of the hierarchy.[2]

[2] Albeit in the realm of syntactic structure, the popular Universal Dependencies (UD; Nivre et al., 2020) initiative is currently exploring the introduction of 'enhanced' dependencies, where unanchored nodes for unexpressed material beyond the surface string can be postulated (Schuster and Manning, 2016). Whether or not these nodes occupy a well-defined position in the otherwise total order of basic UD nodes remains an open question, but either way the presence of unanchored nodes will take enhanced UD graphs beyond the bi-lexical Flavor (0) graphs in our terminology.

3 Meaning Representation Frameworks

The shared task combines five distinct frameworks for graph-based meaning representation, each with its specific formal and linguistic assumptions. This section reviews the frameworks and presents English example graphs for sentence #20209013 from the venerable Wall Street Journal (WSJ) Corpus from the Penn Treebank (PTB; Marcus et al., 1993):

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

The example exhibits some interesting linguistic complexity, including what is called a tough adjective (*impossible*), a scopal adverb (*almost*), a tripartite coordinate structure, and apposition. The example graphs in Figures 1 through 4 are pre-

sented in order of (arguably) increasing 'abstraction' from the surface string, i.e. ranging from fully anchored Flavor (1) to unanchored Flavor (2).

Elementary Dependency Structures The EDS graphs (Oepen and Lønning, 2006) originally derive from the underspecified logical forms computed by the English Resource Grammar (Flickinger et al., 2017; Copestake et al., 2005). These logical forms are not in and of themselves semantic graphs (in the sense of §2 above) and are often refered to as English Resource Semantics (ERS; Bender et al., 2015).[3] Elementary Dependency Structures (EDS; Oepen and Lønning, 2006) encode English Resource Semantics in a variable-free semantic dependency graph—not limited to bi-lexical dependencies—where graph nodes correspond to logical predications and edges to labeled argument positions. The EDS conversion from underspecified logical forms to directed graphs discards partial information on semantic scope from the full ERS, which makes these graphs abstractly—if not linguistically—similar to Abstract Meaning Representation (see below).

Nodes in EDS are in principle independent of surface lexical units, but for each node there is an explicit, many-to-many anchoring onto sub-strings of the underlying sentence. Thus, EDS instantiates Flavor (1) in our hierarchy of different formal types of semantic graphs and, more specfically, are fully anchored but unordered. Avoiding a one-to-one correspondence between graph nodes and surface lexical units enables EDS to adequately represent, among other things, lexical decomposition (e.g. of comparatives), sub-lexical or construction semantics (e.g. corresponding to morphological derivation or syntactic compounding, respectively), and covert (e.g. elided) meaning contributions. All nodes in the example EDS in Figure 1 make explicit their anchoring onto sub-strings of the underlying input, for example span $\langle 2 : 9 \rangle$ for *similar*.

In the EDS analysis for the running example, nodes representing covert quantifiers (e.g. on bare nominals, labeled udef_q[4]), the two-place such+as_p relation, as well as the implicit_conj(unction) relation (which reflects recursive decomposition of the coordinate structure into binary predications) do not correspond to individual surface tokens (but are anchored on larger spans, overlapping with anchors from other nodes). Conversely, the two nodes associated with *similar* indicate lexical decomposition as a comparative predicate, where the second argument of the comp relation (the 'point of reference') remains unexpressed in Example (1).

Prague Tectogrammatical Graphs These graphs present a conversion from the multi-layered (and somewhat richer) annotations in the tradition of Prague Functional Generative Description (FGD; Sgall et al., 1986), as adopted (among others) in the Prague Czech–English Dependency Treebank (PCEDT; Hajič et al., 2012) and Prague Dependency Treebank (PDT; Böhmová et al., 2003). For more details on how the graphs are obtained from the original annotations, see Zeman and Hajič (2020).

The PTG structures essentially recast core predicate–argument structure in the form of mostly anchored dependency graphs, albeit introducing 'empty' (or *generated*, in FGD terminology) nodes, for which there is no corresponding surface token. Thus, these partially anchored representations instantiate Flavor (1) in our hierarchy of different formal types of semantic graphs, where anchoring relations can be discontinuous: For example, the *technique* node in Figure 2 is anchored to both the noun and its indefinite determiner *a*. PTG structures assume a total order of nodes, which provides the foundation for an underlying theory of topic–focus articulation, as proposed by Hajičová et al. (1998).

The PTG structure for our running example has many of the same dependency edges as the EDS graph (albeit using a different labeling scheme and inverse directionality in a few cases), but it analyzes the predicative copula as semantically contentful and does not treat *almost* as 'scoping' over the entire graph. In the example graph, there are two generated nodes to represent the unexpressed BEN(efactive) of the *impossible* relation as well as the unexpressed ACT(or) argument of the three-place *apply* relation, respectively; these nodes are related by an edge indicating grammatical coreference. In this graph, the indefinite determiner, infinitival to, and the vacuous preposition marking

[3] The underlying grammar is rooted in the general linguistic theory of Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994).

[4] In the EDS example in Figure 1, all nodes corresponding to instances of bare 'nominal' meanings are bound by a covert quantificational predicate, including the group-forming implicit_conj and _and_c nodes that represent the nested, binary-branching coordinate structure. This practice of uniform quantifier introduction in ERS is acknowledged as "particularly exuberant" by Steedman (2011, p. 21).

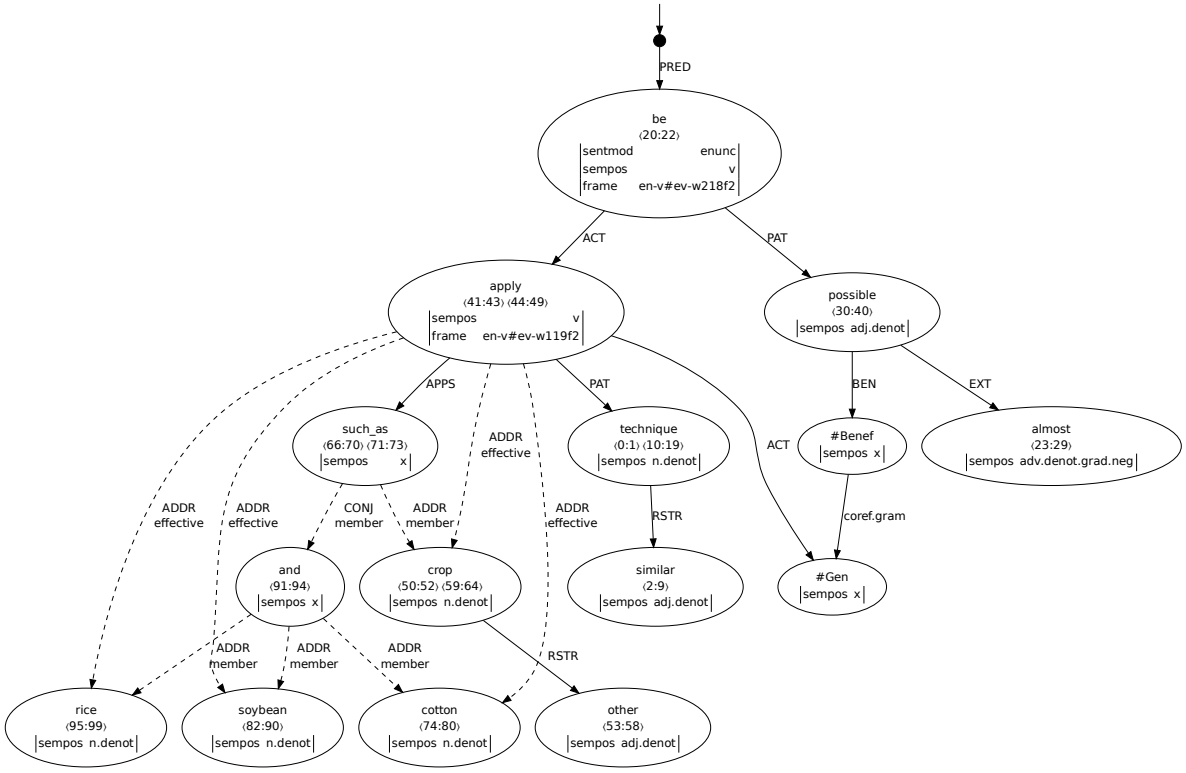

Figure 2: Semantic dependency graphs for the running example *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice*: Prague Tectogrammatical Graphs (PTG). In addition to node properties, visualized similarly to the EDS in Figure 1, boolean edge attributes are abbreviated below edge labels, for true values.

the deep object of apply can be argued to not have a semantic contribution of their own.

The ADDR argument relation to the *apply* predicate has been recursively propagated to both elements of the apposition and to all members of the coordinate structure. Accordingly, edge labels in PTG are not always functional, in the sense of allowing multiple outgoing edges from one node with the same label.

In FGD, role labels (called *functors*) ACT(or), PAT(ient), ADDR(essee), ORIG(in), and EFF(ect) indicate 'participant' positions in an underlying valency frame and, thus, correspond more closely to the numbered argument positions in other frameworks than their names might suggest.[5] The PTG annotations are grounded in a machine-readable valency lexicon (Urešová et al., 2016), and the `frame` values on verbal nodes in Figure 2 indicate specific verbal senses in the lexicon.

Universal Conceptual Cognitive Annotation Universal Cognitive Conceptual Annotation (UCCA; Abend and Rappoport, 2013) is based on cognitive linguistic and typological theories, primarily Basic Linguistic Theory (Dixon, 2010/2012). The shared task targets the UCCA foundational layer, which focuses on argument structure phenomena (where predicates may be verbal, nominal, adjectival, or otherwise). This coarse-grained level of semantics has been shown to be preserved well across translations (Sulem et al., 2015). It has also been successfully used for improving text simplification (Sulem et al., 2018c), as well as to the evaluation of a number of text-to-text generation tasks (Birch et al., 2016; Sulem et al., 2018a; Choshen and Abend, 2018).

The basic unit of annotation is the *scene*, denoting a situation mentioned in the sentence, typically involving a predicate, participants, and potentially modifiers. Linguistically, UCCA adopts a notion of semantic constituency that transcends pure dependency graphs, in the sense of introducing separate, unlabeled nodes, called *units*. One or more labels are assigned to each edge. Formally, UCCA has a

[5] Accordingly, multiple instances of the same core participant role—as ADDR:member in Figure 2—will only occur with propagation of dependencies into paratactic constructions.

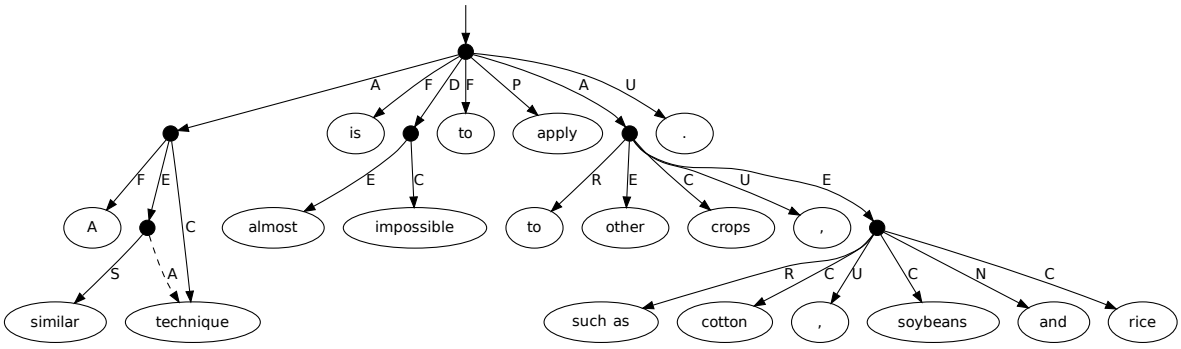

Figure 3: Universal Conceptual Cognitive Annotation (UCCA), foundational layer, for the running example *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice*. The dashed edge whose target is the node anchored to *technique* abbreviates a boolean remote edge attribute.

Type (1) flavor, where leaf (or terminal) nodes of the graph are anchored to possibly discontinuous sequences of surface sub-strings, while interior (or 'phrasal') graph nodes are formally unanchored.

The UCCA graph for the running example (see Figure 3) includes a single scene, whose main relation is the Process (P) evoked by *apply*. It also contains a secondary relation labeled Adverbial (D), *almost impossible*, which is broken down into its Center (C) and Elaborator (E); as well as two complex arguments, labeled as Participants (A). Unlike the other frameworks in the task, the UCCA foundational layer integrates all surface tokens into the graph, possibly as the targets of semantically bleached Function (F) and Punctuation (U) edges. UCCA graphs need not be rooted trees: Argument sharing across units will give rise to reentrant nodes much like in the other frameworks. For example, *technique* in Figure 3 is both a Participant in the scene evoked by *similar* and a Center in the parent unit. UCCA in principle also supports implicit (unexpressed) units which do not correspond to any tokens, but these are currently excluded from parsing evaluation and, thus, suppressed in the UCCA graphs distributed in the context of the shared task.

Abstract Meaning Representation The shared task includes Abstract Meaning Representation (AMR; Banarescu et al., 2013), which in the MRP hierarchy of different formal types of semantic graphs (see §2 above) is simply unanchored, i.e. represents Flavor (2). The AMR framework is independent of particular approaches to derivation and compositionality and, accordingly, does not make explicit how elements of the graph correspond to the surface utterance. Although most AMR parsing research presupposes a pre-processing step that 'aligns' graph nodes with (possibly discontinuous) sets of tokens in the underlying input, this anchoring is not part of the meaning representation proper.

At the same time, AMR frequently invokes lexical decomposition and normalization towards verbal senses, such that AMR graphs often appear to 'abstract' furthest from the surface signal. Since the first general release of an AMR graph bank in 2014, the framework has provided a popular target for data-driven meaning representation parsing and has been the subject of two consecutive tasks at SemEval 2016 and 2017 (May, 2016; May and Priyadarshi, 2017).

The AMR example graph in Figure 4 has a topo-

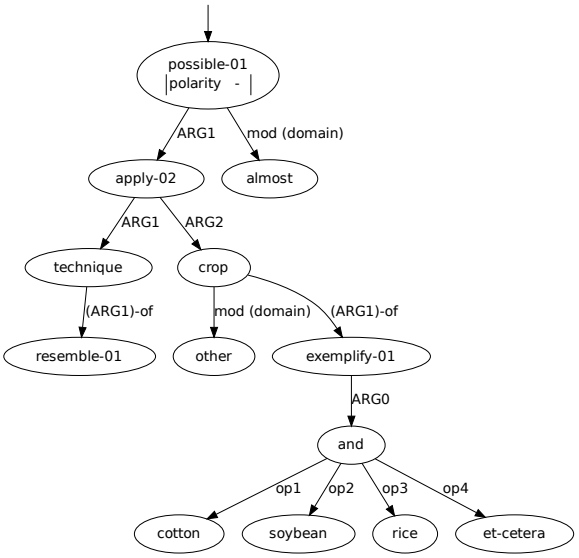

Figure 4: Abstract Meaning Representation (AMR) for the running example *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice*. Edge labels in parentheses indicate normalized (i.e. un-inverted) roles.

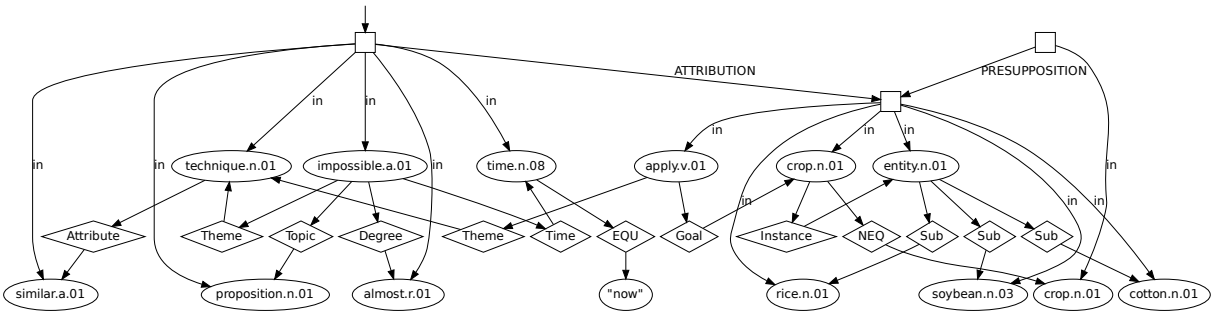

Figure 5: Discourse Representation Graph (DRG) for the running example *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice*. Different node shapes are not formally part of the graph but serve as a visual aid to distinguish different types of the underlying DRS elements.

logy broadly comparable to EDS, with some notable differences. Similar to the UCCA example graph (and unlike EDS), the AMR representation of the coordinate structure is flat. Although most lemmas are linked to derivationally related forms in the sense lexicon, this is not universal, as seen by the nodes corresponding to *similar* and *such as*, which are labeled as resemble-01 and exemplify-01, respectively. These sense distinctions (primarily for verbal predicates) are grounded in the inventory of predicates from the PropBank lexicon (Kingsbury and Palmer, 2002; Hovy et al., 2006).

Role labels in AMR encode semantic argument positions, with the particular roles defined according to each PropBank sense, though the counting in AMR is zero-based such that the ARG1 and ARG2 roles in Figure 4 often correspond to ARG2 and ARG3, respectively, in the EDS of Figure 1. PropBank distinguishes such numbered arguments from non-core roles labeled from a general semantic inventory, such as frequency, duration, or domain.

Figure 4 also shows the use of inverted edges in AMR, for example ARG1-of and mod. These serve to allow annotators (and in principle also parsing systems) to view the graph as a tree-like structure (with occasional reentrancies) but are formally merely considered notational variants. Therefore, the MRP rendering of the AMR example graph also provides an unambiguous indication of the underlying, normalized graph: Edges with a label component shown in parentheses are to be reversed in normalization, e.g. representing an actual ARG0 edge from resemble-01 to technique or a domain edge from other to crop.

Given the non-compositionality of AMR annotation, AMR allows the introduction of semantic concepts which have no explicit lexicalization in the text, for example the et-cetera element in the coordinate structure in Figure 4. Conversely, like in the other frameworks (except UCCA), some surface tokens are analyzed as semantically vacuous. For example, parallel to the PTG graph in Figure 2, there is no meaning contribution annotated for the determiner *a* (let alone for covert determiners in bare nominals, as made explicit in EDS).

Discourse Representation Graphs Finally, Discourse Representation Graphs (DRG) provide a graph encoding of Discourse Representation Structure (DRS), the meaning representations at the core of Discourse Representation Theory (DRT; Kamp and Reyle, 1993; Van der Sandt, 1992; Asher, 1993). DRSs can model many challenging semantic phenomena including quantifiers, negation, scope, pronoun resolution, presupposition accommodation, and discourse structure. Moreover, they are directly translatable into first-order logic formulas to account for logical inference.

DRG used in the shared task represents a type of graph encoding of DRS that makes the graphs structurally as close as possible to the structures found in other frameworks; Abzianidze et al. (2020) provide more details on the design choices in the DRG encoding. The source DRS annotations are taken from data release 3.0.0 of the Parallel Meaning Bank (PMB; Abzianidze et al., 2017; Bos et al., 2017).[6] Although the annotations in the PMB are compositionally derived from lexical semantics, anchoring information is not explicit in its DRSs; thus, (like AMR) the DRG framework formally instantiates Flavor (2) of meaning representations.

The DRG of the running example is given in Figure 5. The concepts (vissualized as oval shapes) are represented by WordNet 3.0 senses and semantic roles (in diamond shapes) by the adapted version

[6] https://pmb.let.rug.nl/data.php

		EDS	PTG	UCCA	AMR	DRG
	Flavor	1	1	1	2	2
TRAIN	Text Type	newspaper	newspaper	mixed	mixed	mixed
	Sentences	37,192	42,024	6,872	57,885	6,606
	Tokens	861,831	1,026,033	171,838	1,049,083	44,692
VALIDATE	Text Type	mixed	mixed	mixed	mixed	mixed
	Sentences	3,302	1,664	1,585	3,560	885
	Tokens	65,564	40,770	25,982	61,722	5,541
TEST	Text Type	mixed	newspaper	mixed	mixed	mixed
	Sentences	4,040	2,507	600	2,457	898
	Tokens	68,280	59,191	18,633	49,760	5,991

Table 1: Quantitative summary of English gold-standard training, validation, and evaluation data for the five frameworks in the cross-framework track; token counts reflect the morpho-syntactic companion parses, see §4.

of VerbNet roles. Nodes with quoted labels represent entities which semantically behave as constants. Such a node is used for the indexical "now", modelling the time of speech, which is part of the semantics of the present-tense copula *is*.

Explicit encoding of the scope is one of the main differences between DRG and the other frameworks. Scopes can be triggered by discourse segments, negation, universal quantification, clause embedding (e.g. *to apply* . . .), and presuppositions (e.g. *other crops*). The scopes are represented as unlabeled (square-shaped) nodes in DRG (UCCA also has unlabeled nodes, albeit for a different reason). The node for the first discourse segment is treated as a root, which is connected to the scope of the embedded clause by the ATTRIBUTION discourse relation. The latter scope presupposes the scope containing a *crop* which is different (with NEQ inequality) from the group of crops consisting of (with the Sub semantic role) *rice*, *soybeans*, and *cotton*. Each concept, represented by a WordNet synset, has explicitly assigned its scope via in edges.[7]

Compared to the other frameworks, DRG structures are larger in size due to the number of semantic relations, explicit nodes for scope, scope membership edges, role reification, and information about the time (which usually introduces at least four additional nodes).

[7]Since in principle the scope of a semantic role cannot be uniquely determined by the scopes of its arguments, semantic roles are reified as nodes and can have ingoing in edges. But whenever the scopes of a role and its arguments coincide, the scope membership edge for the role is omitted and hence recoverable. This decision decreases the number of edges in DRG.

4 Task Setup

The following paragraphs summarize the 'logistics' of the MRP 2020 shared task. Except for the addition of the new cross-lingual track, the overall task setup mirrored that of the 2019 predecessor; please see Oepen et al. (2019) for additional background.

Cross-Framework Track The English training, validation, and evaluation data are summarized in Table 1. For EDS, PTG, UCCA, and AMR the provenance of these gold-standard annotations is the same as in the MRP 2019 setup (Oepen et al., 2019).[8] The DRG target structures have been converted using the procedure sketched in §3 above. Unlike in the 2019 edition of the task, designated validation segments have been provided for all five frameworks in the cross-framework track; this data could be used during system development, e.g. for parameter tuning, but not for training the final system submission. For EDS, UCCA, and AMR, the 2020 validation data corresponds to the 2019 evaluation segments, thus allowing some comparability across the two editions of the MRP shared task.

As a common point of reference, the training data includes a sample of 89 WSJ sentences annotated in all five frameworks (twenty for DRG); for all frameworks but DRG, the evaluation data further includes parallel annotations over the same random selection of 100 sentences from the novel *The Little Prince* (by Antoine de Saint-Exupéry) as used in MRP 2019, dubbed LPPS. These parallel subsets of the gold-standard data are available for public download from the task site (see §9 below).

[8]There are slightly more EDS and PTG (compared to PSD in 2019) graphs this year, because the two underlying resources are no longer intersected; for UCCA, the 2020 release includes additional, recent gold-standard annotations.

			EDS	PTG	UCCA	AMR^{-1}	DRG
PROPORTIONS	(02)	Average Tokens per Graph	22.17	24.42	25.01	18.12	6.77
	(03)	Average Nodes per Token	1.26	0.74	1.33	0.64	2.09
	(04)	Distinct Edge Labels	10	72	15	101	16
	(05)	Percentage of top nodes	0.99	1.27	1.66	3.77	3.40
	(06)	Percentage of node labels	29.02	21.61	–	43.91	39.81
	(07)	Percentage of node properties	12.54	26.22	–	7.63	–
	(08)	Percentage of node anchors	29.02	19.63	38.80	–	–
	(09)	Percentage of (labeled) edges	28.43	26.10	56.88	44.69	56.79
	(10)	Percentage of edge attributes	–	5.17	2.66	–	–
TREENESS	(11)	$\%_g$ Rooted Trees	0.09	22.63	28.19	22.05	0.35
	(12)	$\%_g$ Treewidth One	68.60	22.67	34.17	49.91	0.35
	(13)	Average Treewidth	1.317	2.067	1.691	1.561	2.131
	(14)	Maximal Treewidth	3	7	4	5	5
	(15)	Average Edge Density	1.015	1.177	1.055	1.092	1.265
	(16)	$\%_n$ Reentrant	32.77	16.23	4.90	19.89	25.92
	(17)	$\%_g$ Cyclic	0.27	33.97	0.00	0.38	0.27
	(18)	$\%_g$ Not Connected	1.90	0.00	0.00	0.00	0.00
	(19)	$\%_g$ Multi-Rooted	99.93	0.00	0.00	71.64	32.32

Table 2: Contrastive graph statistics for the MRP 2020 English training data using a subset of the properties defined by Kuhlmann and Oepen (2016). Here, $\%_g$ and $\%_n$ indicate percentages of all graphs and nodes, respectively, in each framework; AMR^{-1} refers to the *normalized* form of the graphs, with inverted edges reversed, as discussed in §3. The second block of statistics indicates the proportional distribution of different formal types of information in the graphs, according to the categorization used in the MRP cross-framework evaluation metric (see §5).

Table 2 provides a quantitative side-by-side comparison of the training data, using some of the graph-theoretic properties discussed by Kuhlmann and Oepen (2016); see §2 for semi-formal definitions. The table indicates clear differences among the frameworks. The underlying input strings for AMR (where text selection is more varied), for example, are shorter, and much shorter in turn for DRG. EDS, UCCA, and DRG have many more nodes per token, on average, than the other frameworks—reflecting lexical decomposition, 'phrasal' grouping, and role reification, respectively, as evident in Figures 1, 3, and 5. In some respects, the PTG and UCCA graphs are more tree-like than graphs in the other frameworks, for example in their proportions of actual rooted trees, the frequencies of reentrant nodes, and the lack of multi-rooted structures. At the same time, PTG exhibits comparatively high average and maximal treewidth and is the only framework with a non-trivial percentage of cyclic graphs.

Cross-Lingual Track For four of the frameworks (excluding EDS), gold-standard training and evaluation data has been compiled in other languages than English: Mandarin Chinese for AMR, Czech for PTG, and German for UCCA and DRG. For UCCA and in particular DRG, however, available data is comparatively limited, as summarized in Table 3. These target representations constitute a separate *cross-lingual* track, which transcends the MRP 2019 task setup.

Additional Resources For reasons of comparability and fairness, the shared task constrained which additional data or pre-trained models (e.g. corpora, word embeddings, language models, lexica, or other annotations) can be legitimately used besides the resources distributed by the task organizers—such that all participants should in principle have access to the same range of data. However, to keep such constraints to the minimum required, a 'white-list' of legitimate resources was compiled from nominations by participants (with a cut-off date eight weeks before the end of the eval-

		PTG	UCCA	AMR	DRG
	Language	Czech	German	Chinese	German
	Flavor	1	1	1	2
TRAIN	Text Type	newspaper	mixed	mixed	mixed
	Sentences	43,955	4,125	18,365	1,575
	Tokens	740,466	95,634	428,054	9,088
TEST	Text Type	newspaper	mixed	mixed	mixed
	Sentences	5,476	444	1,713	403
	Tokens	92,643	10,585	39,228	2,384

Table 3: Quantitative summary of gold-standard data for the four frameworks in the cross-lingual track.

uation period).[9] Thus, the task design reflects what is at times called a *closed track*, where participants are constrained in which additional data and pre-trained models can be used in system development.

Companion Syntactic Parses At a technical level, training (and evaluation) data were distributed in two formats, (a) as sequences of 'raw' sentence strings and (b) in pre-tokenized, part-of-speech–tagged, lemmatized, and syntactically parsed form. For the latter, premium-quality morpho-syntactic dependency analyses were provided to participants, called the MRP 2020 companion parses. These parses were obtained using a pre-release of the 'future' UDPipe architecture (Straka, 2018; Straka and Straková, 2020), trained on available gold-standard UD 2.x treebanks, for English augmented with conversions from PTB-style annotations in the WSJ and OntoNotes corpora (Hovy et al., 2006), using the UD-style CoreNLP 4.0 tokenizer (Manning et al., 2014) and jack-knifing where appropriate (to avoid overlap with the texts underlying the MRP semantic graphs).

Rules of Participation While the various meaning representation frameworks and graph banks represented in the shared task inevitably present considerable linguistic variation, all MRP 2020 data was repackaged in a uniform and normalized abstract representation with a common serialization, the same JSON Lines format as used in the previous year (Oepen et al., 2019). Because some of the semantic graph banks involved in the shared task had originally been released by the Linguistic Data Consortium (LDC), the training data was made available to task participants by the LDC under no-cost evaluation licenses. All task data (including system submissions and evaluation results) is being prepared for general release through the LDC, while subsets that are copyright-free will also become available for direct, open-source download.

The shared task was first announced in March 2020, the initial release of the cross-framework training data became available in late April, and the evaluation period ran between July 27 and August 10, 2020; during this period, teams obtained the unannotated input strings for the evaluation data and had available a little more than two weeks to prepare and submit parser outputs. Submission of semantic graphs for evaluation was through the on-line CodaLab infrastructure. Teams were allowed to make repeated submissions, but only the most recent successful upload to CodaLab within the evaluation period was considered for the official, primary ranking of submissions. Task participants were encouraged to process all inputs using the same general parsing system, but—owing to inevitable fuzziness about what constitutes 'one' parser—this constraint was not formally enforced.

[9] See `http://svn.nlpl.eu/mrp/2020/public/resources.txt` for the list of legitimate extra resources.

	EDS	PTG	UCCA	AMR	DRG
Top Nodes	✓	✓	✓	✓	✓
Node Labels	✓	✓	✗	✓	✓
Node Properties	✓	✓	✗	✓	✗
Node Anchors	✓	✓	✓	✗	✗
Labeled Edges	✓	✓	✓	✓	✓
Edge Attributes	✗	✓	✓	✗	✗

Table 4: Different tuple types per framework.

5 Evaluation

Following the previous edition of the shared task, the official MRP metric for the task is the micro-average F_1 score across frameworks over all tuple types that encode 'atoms' of information in MRP graphs. The cross-framework metric uniformly evaluates graphs of different flavors, regardless of a specific framework exhibiting (a) labeled or unlabeled nodes or edges, (b) nodes with or without anchors, and (c) nodes and edges with optional properties and attributes, respectively (see Table 4).

The MRP metric generalizes earlier framework-specific metrics (Dridan and Oepen, 2011; Cai and Knight, 2013; Hershcovich et al., 2019a) in terms of decomposing each graph into sets of typed tuples, as indicated in Figure 6. To quantify graph similarity in terms of tuple overlap, a correspondence relation between the nodes of the gold-standard and system graphs must be determined. Adapting a search procedure for the NP-hard maximum common edge subgraph (MCES) isomorphism problem, the MRP scorer will search for the node-to-node correspondence that maximizes the intersection of tuples between two graphs, where node identifiers (m and n in Figure 6) act like variables that can be equated across the gold-standard and system graphs.[10] This means that during evaluation all information in the MRP graphs is con-

[10] Conceptually, the search expands both graphs into larger structures with 'lightly labeled' nodes and edges, e.g. treating node properties much like 'pseudo-edges' with globally unique constant-valued target nodes.

Teams	Cross-Framework					Cross-Lingual				Reference
	AMR	DRG	EDS	PTG	UCCA	AMR	DRG	PTG	UCCA	
Hitachi	✓	✓	✓	✓	✓	✓	✓	✓	✓	Ozaki et al. (2020)
ÚFAL	✓	✓	✓	✓	✓	✓	✓	✓	✓	Samuel and Straka (2020)
HIT-SCIR	✓	✓	✓	✓	✓	✓	✓	✓	✓	Dou et al. (2020)
HUJI-KU	✓	✓	✓	✓	✓	✓	✓	✓	✓	Arviv et al. (2020)
ISCAS	✓	✓	✓	✓	✓	✗	✗	✗	✗	
TJU-BLCU	✓	✓	✓	✓	✓	✓	✓	✓	✗	
JBNU	✓	✗	✗	✗	✗	✗	✗	✗	✗	Na and Min (2020)
ÚFAL	✓	✓	✓	✓	✓	✓	✓	✓	✓	Samuel and Straka (2020)
ERG	✗	✗	✓	✗	✗	✗	✗	✗	✗	Oepen and Flickinger (2019)

Table 5: Overview of participating teams and the tracks they participated in. Columns correspond to tracks and frameworks, and rows correspond to teams. The top block represents 'official' submissions, which participated in the competition. The middle block represents 'unofficial' submissions, which were submitted after the closing deadline. The bottom row represents the ERG baseline.

sidered with equal weight, i.e. tops, node and edge labels, properties and attributes, and anchors.

MRP scoring is carried out using the open-source `mtool` software—the Swiss Army Knife of Meaning Representation[11]—which implements a refinement of the MCES algorithm by McGregor (1982). Based on pre-computed per-node rewards and upper bounds on adjacent edge correspondences, candidate node-to-node mappings are initialized and scheduled in decreasing order of expected similarity. For increased efficiency (in principle tractability, in fact), `mtool` will return the best available solution when it exhausts its preset search space limits. This anytime behavior of the scores provides a distinction between *exact* vs. *approximate* solutions (which contrasts with the greedy hill-climbing search of e.g. `Smatch`; Cai and Knight, 2013). MRP scoring is robust with respect to equivalent variations of values, e.g. case and string vs. number type distinctions for all literals. Comparison of anchor values ignores whitespace character positions, internal segmentation of adjacent anchors, and basic punctuation marks in the left or right periphery of a normalized anchor. Assuming the string *Oh no!* as a hypothetical parser input, the following anchorings will all be considered equivalent: $\{\langle 0{:}6\rangle\}$, $\{\langle 0{:}2\rangle, \langle 3{:}6\rangle\}$, $\{\langle 0{:}1\rangle, \langle 1{:}6\rangle\}$, and $\{\langle 0{:}5\rangle\}$.

[11] https://github.com/cfmrp/mtool

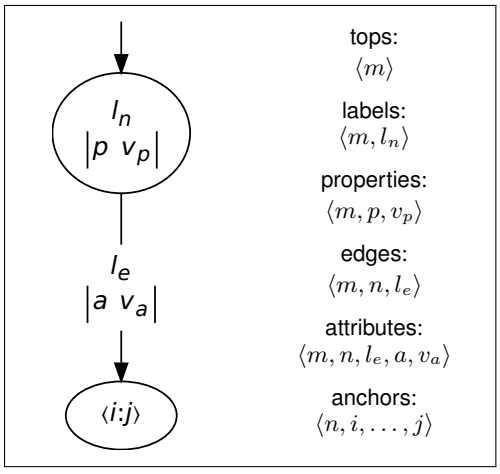

Figure 6: Representing an abstractMRP graph as a set of typed tuples, with m and n as node identifiers for the top and bottom node, respectively.

6 Submissions and Results

Six teams submitted parser outputs to the shared task within the official evaluation period. In addition, we received two submissions after the submission deadline, which we mark as 'unofficial'. We further include results from an additional 'reference' system by one of the task co-organizers, namely EDS outputs from the grammar-based ERG parser (Oepen and Flickinger, 2019).

Table 5 presents an overview of the participating systems and the tracks and frameworks they submitted results for. All official systems submitted results for the cross-framework track (across all frameworks), and additionally five of them submitted results to the cross-lingual track as well (where TJU-BLCU did not submit UCCA parser outputs in the cross-lingual track). We note that the shared task explicitly allowed partial submissions, in order to lower the bar for participation (which is no doubt substantial). Two of the teams—ISCAS and TJU-BLCU—declined the invitation to submit a system description paper to the shared task proceedings.

Team	Cross-Framework						Cross-Lingual				
	All	**EDS**	**PTG**	**UCCA**	**AMR**	**DRG**	**All**	**PTG**	**UCCA**	**AMR**	**DRG**
Hitachi	1	1	2	1	1	–	–	–	–	–	–
	1	1	1	2	1	2	1	2	3	1	1
ÚFAL	1	2	1	1	2	–	–	–	–	–	–
	1	2	2	1	1	1	1	1	1	2	2
HIT-SCIR	3	3	3	3	3	–	–	–	–	–	–
	3	3	3	2	3	3	3	3	2	3	3
HUJI-KU	4	5	4	4	5	–	–	–	–	–	–
	4	5	4	4	5	5	4	4	4	4	4
ISCAS	5	4	6	6	4	–	–	–	–	–	–
	5	4	6	6	4	4	–	–	–	–	–
TJU-BLCU	6	6	5	5	6	–	–	–	–	–	–
	6	6	5	5	6	6	5	5	–	5	5

Team	Tops			Labels			Properties			Anchors			Edges			Attributes			All		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Hitachi	.93	.93	.93	.65	.68	.66	.63	.62	.62	.71	.70	.70	.82	.80	.81	.39	.32	.34	.85	.85	.85
	.95	.95	.95	.72	.72	.72	.54	.54	.54	.57	.55	.56	.83	.80	.82	.24	.23	.24	.88	.85	.86
ÚFAL	.93	.93	.93	.68	.68	.68	.61	.60	.60	.69	.71	.70	.80	.79	.80	.42	.33	.36	.85	.85	.85
	.94	.94	.94	.74	.73	.74	.55	.54	.54	.56	.57	.56	.80	.80	.80	.23	.24	.24	.87	.86	.86
HIT-SCIR	.94	.94	.94	.63	.64	.64	.45	.41	.43	.71	.71	.71	.77	.76	.77	.37	.30	.33	.80	.80	.80
	.94	.94	.94	.70	.69	.69	.44	.37	.40	.57	.56	.57	.77	.75	.76	.22	.22	.22	.82	.80	.81
HUJI-KU	.87	.84	.85	.36	.36	.36	.29	.18	.20	.66	.67	.67	.67	.62	.64	.15	.07	.10	.73	.63	.67
	.88	.83	.85	.29	.29	.29	.40	.24	.28	.51	.51	.51	.65	.62	.64	.07	.08	.07	.73	.58	.64
ISCAS	.70	.70	.70	.50	.49	.48	.22	.26	.24	.35	.41	.37	.52	.35	.39	–	–	–	.53	.43	.43
	.75	.74	.74	.56	.55	.55	.22	.22	.21	.29	.31	.29	.57	.40	.44	–	–	–	.58	.46	.48
TJU-BLCU	.83	.82	.83	.41	.29	.34	–	–	–	.45	.30	.35	.53	.30	.37	–	–	–	.57	.30	.39
	.75	.74	.75	.54	.29	.38	–	–	–	.33	.14	.19	.44	.18	.24	–	–	–	.55	.22	.30
ÚFAL	.93	.93	.93	.68	.68	.68	.61	.60	.60	.71	.71	.71	.80	.80	.80	.43	.34	.37	.85	.85	.85
	.94	.94	.94	.74	.73	.74	.55	.54	.54	.57	.57	.57	.80	.80	.80	.23	.24	.24	.87	.86	.87

Team	Tops			Labels			Properties			Anchors			Edges			Attributes			All		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Hitachi	.96	.96	.96	.65	.65	.65	.44	.42	.43	.7	.68	.69	.8	.77	.78	.27	.27	.26	.86	.84	.85
ÚFAL	.95	.95	.95	.66	.66	.66	.43	.43	.43	.65	.72	.68	.78	.79	.79	.3	.33	.31	.84	.86	.85
HIT-SCIR	.95	.95	.95	.53	.52	.53	.21	.18	.20	.47	.47	.47	.66	.65	.66	.23	.24	.23	.72	.67	.69
HUJI-KU	.9	.84	.87	.15	.15	.15	.31	.32	.32	.42	.42	.42	.59	.58	.59	.08	.08	.08	.69	.54	.60
TJU-BLCU	.56	.55	.56	.41	.21	.27	–	–	–	.23	.12	.15	.28	.13	.18	–	–	–	.35	.15	.20
ÚFAL	.95	.95	.95	.66	.66	.66	.43	.43	.43	.71	.72	.72	.79	.79	.79	.3	.33	.31	.86	.86	.86

Table 6: Official rankings (top) for both tracks, and MRP scores for the cross-framework (middle) and cross-lingual (bottom) tracks. Each cross-framework submission is evaluated in two settings, where the top scores present results for the LPPS sub-corpus, and the bottom ones for the full English evaluation set. The rankings are presented both for the overall average scores (All), and separately per framework. Evaluation results are broken down by 'atomic' component pieces. For each component we report precision (P), recall (R), and F_1 score (F). Entries in the two MRP tables are split into the same blocks as in Table 5: official (top) vs. unofficial (bottom) submissions, omitting the two highly partial unofficial submissions by JBNU and ERG.

	EDS			PTG			UCCA			AMR			DRG		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Hitachi	0.97	0.97	0.97	0.80	0.84	0.82	0.86	0.80	0.83	0.78	0.79	0.79	–	–	–
	0.94	0.93	0.94	0.89	0.89	0.89	0.78	0.72	0.75	0.83	0.80	0.82	0.94	0.92	0.93
ÚFAL	0.96	0.95	0.95	0.81	0.84	0.83	0.84	0.82	0.83	0.77	0.79	0.78	–	–	–
	0.93	0.92	0.93	0.88	0.89	0.88	0.75	0.78	0.76	0.81	0.79	0.80	0.95	0.93	0.94
HIT-SCIR	0.90	0.89	0.89	0.78	0.78	0.78	0.84	0.80	0.82	0.68	0.71	0.70	–	–	–
	0.87	0.88	0.87	0.85	0.84	0.84	0.75	0.74	0.75	0.74	0.66	0.70	0.90	0.89	0.89
HUJI-KU	0.83	0.76	0.79	0.71	0.49	0.58	0.80	0.76	0.78	0.56	0.5	0.53	–	–	–
	0.83	0.76	0.80	0.69	0.44	0.54	0.73	0.73	0.73	0.57	0.49	0.52	0.84	0.5	0.63
ISCAS	0.86	0.90	0.88	0.12	0.25	0.16	0.45	0.08	0.13	0.68	0.47	0.56	–	–	–
	0.85	0.87	0.86	0.14	0.26	0.18	0.42	0.03	0.06	0.74	0.53	0.61	0.78	0.63	0.69
TJU-BLCU	0.83	0.51	0.64	0.41	0.24	0.30	0.52	0.13	0.21	0.50	0.34	0.4	–	–	–
	0.84	0.35	0.49	0.38	0.15	0.21	0.50	0.06	0.10	0.54	0.21	0.30	0.49	0.34	0.40
JBNU	–	–	–	–	–	–	–	–	–	0.74	0.73	0.74	–	–	–
	–	–	–	–	–	–	–	–	–	0.71	0.62	0.66	–	–	–
ÚFAL	0.96	0.95	0.95	0.83	0.84	0.84	0.84	0.81	0.83	0.77	0.79	0.78	–	–	–
	0.93	0.92	0.93	0.89	0.89	0.89	0.75	0.78	0.76	0.81	0.79	0.80	0.95	0.93	0.94
ERG	0.95	0.96	0.96	–	–	–	–	–	–	–	–	–	–	–	–
	0.94	0.91	0.93	–	–	–	–	–	–	–	–	–	–	–	–

Table 7: Per-framework results for the cross-framework track, using the same groupings as in Table 6.

Table 6 presents the official rankings for the official submissions (top), including an overall score for each track and per-framework rankings. Rankings are given over the LPPS dataset, a sample from the Little Prince annotated by all frameworks save for DRG, and over the entire test set. Results are consequently more readily comparable for the LPPS sub-corpus, but should be more robust on the entire test corpus, due to its larger size (see §4). That said, LPPS and overall test results are very similar, both in terms of ranking and in terms of bottom line scores.

The main task results are summarized in Table 6 for both the cross-framwork (middle) and cross-lingual (bottom) tracks. Results are broken down into component pieces. Edge attributes are only present in PTG and UCCA. While they are still predicted with fairly low results, this constitutes a notable improvement over the findings of MRP 2019 (the best score on the official track on UCCA edge attributes was 0.12 F_1 then, as opposed to 0.36 now). Anchors are predicted with substantially lower scores compared to MRP 2019, probably since we did not include in MRP 2020 the bi-lexical Flavor (0) frameworks. Edges and tops are slightly more accurate, while labels and properties slightly less, but these are not directly comparable since the frameworks and data are different. See §8 for an overall discussion of the state of the art, considering MRP 2019 and MRP 2020.

Results show that the Hitachi and ÚFAL submissions share the first place for both tracks, and together rank first or second for almost all the individual frameworks (save for UCCA parsing in the cross-lingual track, where Hitachi ranks third). HIT-SCIR further ranks second for UCCA parsing in both tracks. Interestingly, rankings in the per-framework track are similar across frameworks, which may indicate some similarity in the parsing problem exhibited by different linguistic schemes, despite differences in structure and content.

Per-framework scores using the official MRP metric are given in Table 7 for the cross-framework track and Table 8 for the cross-lingual track. Examining these results, we note that cross-framework and cross-lingual scores are quite similar, an encouraging sign of cross-linguistic applicability. Another trend to note is that precision and recall are surprisingly close to each other for many systems, often identical.

7 Overview of Approaches

Compared with systems from MRP 2019, there has been a fairly clear shift in approaches for participating systems this year, resulting in significant improvements in performance. The improvements for some of the frameworks are fairly substantial. For example, the Hitachi system, one of the two winning systems, achieves a score of 0.82 F_1 in AMR parsing, in comparison to 0.73 F_1 achieved by the top AMR parser in MRP 2019. This reflects an improvement of over eight points, reflecting a

	PTG			UCCA			AMR			DRG		
	P	R	F	P	R	F	P	R	F	P	R	F
Hitachi	.89	.86	.87	.79	.79	.79	.82	.79	.8	.93	.94	.93
ÚFAL	.91	.91	.91	.79	.83	.81	.75	.81	.78	.90	.89	.90
HIT-SCIR	.82	.75	.78	.78	.82	.80	.60	.42	.49	.68	.69	.68
HUJI-KU	.65	.53	.58	.74	.76	.75	.55	.38	.45	.82	.50	.62
TJU-BLCU	.51	.14	.22	–	–	–	.46	.17	.25	.42	.28	.34
ÚFAL	.93	.92	.92	.79	.83	.81	.81	.8	.81	.9	.89	.9

Table 8: Per-framework results for the cross-lingual track.

number of innovations from the participants this year, as well as contemporaneous developments outside the shared task (see §8).

Broadly speaking, top performers at MRP 2020 have all adopted a system architecture that is based on an encoder–decoder framework in which the input sentence is encoded into contextualized token embeddings that are used as input to the decoder. The system vary in the decoding strategies.

The Hitachi system adopts a transformer-based encoder–decoder architecture. The system uses the standard transformer encoder in which self-attention and position embeddings are used to compute the contextualized token embeddings. In its decoder, this system has a number of innovations, however. First of all, the system rewrites the meaning representation graphs into a reversible Plain Graph Notation (PGN), and enhances PGN with a number of pseudo-nodes that indicate the end of node prediction, the end of label prediction, etc. These correspond well with parsing actions commonly found in transition-based systems. In this sense, the systems combines the strengths of graph-based parsing on the encoder side resulting from self attention with efficiency of transition-based parsing on the decoder side. Another innovation is the use of a 'hierarchical' decoding process in which the model first predicts a *mode*, and then predicts the next action conditioned on the mode. For example, if the mode is `G`(raph), the decoder predicts a meta node, and if the mode is `S`(urface), the decoder predicts the node label of a specific concept. This allows a fair competition among actions that are similar in nature.

The PERIN system computes contextualized token embeddings with XLM-R (Conneau et al., 2019) on the encoder side, and then on the decoder side, uses separate attention heads to predict the node labels, identify anchors for nodes, and predict edges between nodes, as well as edge labels. Because the label set for nodes is typically very large, rather than predicting the node labels directly, the PERIN system reduces the search space by predicting 'relative rules' that can be used to map surface token strings to node labels in meaning representation graphs, an idea that is similar to the use of Factored Concept Labels in Wang and Xue (2017). Another innovation of the PERIN system is that it is trained with a permutation-invariant loss function that returns the same value independently of how the nodes in the graph are ordered. This captures the unordered nature of nodes in (most of the MRP 2020) meaning representation graphs and prevents situations in which the model is penalized for generating the correct nodes in an order that is different from that in the training data.

The HIT-SCIR and JBNU systems adopt the iterative inference framework first proposed by Cai and Lam (2020) for Flavor (2) meaning representation graphs that do not enforce strict correspondences between tokens in the input sentence and the concepts in meaning representation graphs. The iterative inference framework is also based on an encoder–decoder architecture. The encoder takes the sentence as input and computes contextualized token embeddings that are used as text memory by a decoder that iteratively predicts the next node given the text memory and a predicted parent node in the partially constructed graph memory at the previous time step, and then identifies the parent node for the newly predicted node from the partially constructed graph. While the HIT-SCIR system essentially uses the Cai and Lam (2020) architecture with little modification, the JBNU system attempts to extend the work of Cai and Lam (2020) by using a shared state to make both predictions but did not observe substantial improvements.

Transition-based systems, which had achieved strong performance in the 2019 shared task, are also represented in the competition this year. The HIT-SCIR team uses a transition-based system to parse Flavor (1) meaning representations where

there is a stricter correspondence between tokens in the input sentence and concepts in the meaning representation graph. The HIT-SCIR transition-based system is essentially the overall top performing system they developed for MRP 2019. It uses Stack LSTM to compute transition states in the parsing process, and the parsing actions are tailored to specific meaning representation frameworks. In the training process, the system fine-tunes BERT contextualized encodings.

The HUJI-KU system also extends an entry in the 2019 MRP shared task (originally called TUPA) to parse additional frameworks and handle meaning representation parsing in a multilingual setting. TUPA is a transition-based system that supports general DAG parsing. TUPA applies separate constraints tailored to each meaning representation framework. When parsing cross-framework meaning representations for English, the system is trained with a *BERT-large-cased* pretrained encoder, and when parsing cross-lingual meaning representations, it is trained with multilingual BERT.

8 On the State of the Art

MRP 2019 (Oepen et al., 2019) yielded parsers for five frameworks in a uniform format, of which EDS, UCCA, and AMR are represented in MRP 2020 again. Submissions included transition-, factorization-, and composition-based systems, and gold-standard target structures in 2019 were solely for English. Comparability is limited by the fact that two of the 2020 frameworks (PTG and DRG) are new, training and (in particular) evaluation sets for the others have been updated since MRP 2019, and additional validation sets was introduced. However, the LPPS evaluation sub-corpus (*Le Petite Prince*) is identical between the two years for EDS, UCCA, and AMR. This allows a comparison on nearly equal grounds: as Table 9 shows, in terms of LPPS F_1, the state-of-the-art has substantially improved for EDS and AMR parsing, but stayed the same for UCCA. However, as mentioned in §6, remote edge detection for UCCA improved substantially, though it carries only a small weight in terms of overall scores due to the scarcity of remote edges.

	EDS			**UCCA**			**AMR**		
	P	R	F	P	R	F	P	R	F
2019	.92	.93	.93	.84	.82	.83	.74	.72	.73
2020	.97	.97	.97	.86	.80	.83	.78	.79	.79

Table 9: Per-framework cross-task comparison of top MRP metric scores on LPPS between the 2019 and 2020 editions of the MRP task, on the three frameworks represented in both year, for English. The top systems in MRP 2019 for EDS, UCCA, and AMR were Peking (Chen et al., 2019), HIT-SCIR (Che et al., 2019), and Saarland (Donatelli et al., 2019), respectively; in MRP 2020 the Hitachi system (Ozaki et al., 2020) was at the top for all three frameworks, sharing the UCCA first rank with ÚFAL (Samuel and Straka, 2020).

For EDS, the strongest results were obtained in the MRP 2019 official competition by SUDA–Alibaba (Zhang et al., 2019c). However, in the post-evaluation stage, they were outperformed by the Peking system (Chen et al., 2019). Both used factorization-based parsing with pre-trained contextualized language model embeddings (which has consistently proved to be very effective for other frameworks too). These parsers even approached the performance of the carefully designed grammar-based ERG parser (Oepen and Flickinger, 2019).

English PTG has not been comprehensively addressed by parsers prior to MRP 2020, but a bilexical framework called PSD is a subset of PTG. It was included in the SDP shared tasks (Oepen et al., 2014, 2015) as well as in MRP 2019, and has been addressed by numerous parsers since (Kurita and Søgaard, 2019; Kurtz et al., 2019; Jia et al., 2020, among others). Wang et al. (2019) established the state of the art in supervised PSD using a second-order factorization-based parser, and Fernández-González and Gómez-Rodríguez (2020) matched it using a stack-pointer parser.

Czech PTG, in its original form as published in the Prague Dependency Treebank (Hajič et al., 2018), has been used in several version of the TectoMT machine translation system (Rosa et al., 2016); however, parsing results have not been published separately. A (lossy) conversion has been included in the CoNLL 2009 Shared Task on Semantic Role Labeling (Hajič et al., 2009), but the differences in task design are and conversion make empirical comparison impossible.

UCCA parsing has been dominated by transition-based methods (Hershcovich et al., 2017, 2018; Che et al., 2019). However, both English and German UCCA parsing featured in a SemEval shared task (Hershcovich et al., 2019b), where the best system, a composition-based parser (Jiang et al., 2019), treated the task as constituency tree parsing with the recovery of remote edges as a postprocess-

ing task.

Prior to MRP 2019, Lyu and Titov (2018) parsed AMR using a joint probabilistic model with latent alignments, avoiding cascading errors due to alignment inaccuracies and outperforming previous approaches. Lyu et al. (2020) recently improved the latent alignment parser using stochastic softmax. Lindemann et al. (2019) trained a composition-based parser on five frameworks including AMR and EDS, using the Apply–Modify algebra, on which the third-ranked Saarland submission to MRP 2019 was based (Donatelli et al., 2019). They employed multi-task training with all tackled semantic frameworks and UD, establishing the state of the art on all graph banks but AMR 2017. Since then, a new state-of-the-art has been established for English AMR, using sequence-to-sequence transduction (Zhang et al., 2019a,b) and iterative inference with graph encoding (Cai and Lam, 2019, 2020). Xu et al. (2020a) improved sequence-to-sequence parsing for AMR by using pre-trained encoders, reaching similar performance to Cai and Lam (2020). Astudillo et al. (2020) introduced a stack-transformer to enhance transition-based AMR parsing (Ballesteros and Al-Onaizan, 2017), and Lee et al. (2020) improved it further, using a trained parser for mining oracle actions and combining it with AMR-to-text generation to outperform the state of the-art.

Wang et al. (2018) parsed Chinese AMR with a transition-based system. For cross-lingual AMR parsing, Blloshmi et al. (2020) trained an AMR parser similar to the approach of Zhang et al. (2019b), using cross-lingual transfer learning, outperforming the transition-based cross-lingual AMR parser of Damonte and Cohen (2018) on German, Spanish, Italian, and Chinese.

DRG is a novel graph representation format for DRS that was specially designed for MRP 2020 to make it structurally as close as possible to other frameworks (Abzianidze et al., 2020). However, several semantic parsers exist for DRS, which employ different encodings. Liu et al. (2018) used a DRG format that dominantly labels edges compared to nodes. van Noord et al. (2018) process DRSs in a clausal form, sets of triples and quadruples. The latter format is more common among DRS parsers, as it was officially used by the shared task on DRS parsing (Abzianidze et al., 2019). The shared task gave rise to several DRS parsers: Evang (2019); Liu et al. (2019); van Noord (2019); Fancellu et al. (2019), among which the best results ($F_1 = 0.85$) were achieved by the word-level sequence-to-sequence model with Tranformer (Liu et al., 2019). Note that the DRS shared task used F_1 calculated based on the DRS clausal forms, which is not comparable to MRP F_1 over DRGs.

Similarly to English DRG, German DRG has not been used for semantic parsing prior to the shared task due to the new DRG format. Moreover, semantic parsing with German DRG is novel in the sense that its DRS counterpart is also new. In German DRG, concepts are grounded in English WordNet 3.0 (Fellbaum, 2012) senses assuming that synsets are language-neutral. The mismatch between German tokens and English lemmas of senses must be expected to add additional complexity to German DRG parsing.

Direct comparison to non-MRP results is impossible: we are using a new version of AMRbank. Gold-standard tokenization is not provided for any of the frameworks. We use the MRP scorer. However, general trends appear consistent with recent developments. Pretrained embeddings and cross-lingual transfer help; but multi-task learning less so. There is yet progress to be made in sharing information between parsers for different frameworks and making better use of their overlap.

9 Reflections and Outlook

The MRP series of shared tasks has contributed to general availability of accurate data-driven parsers for a broad range of different frameworks, with performance levels ranging between 0.76 MRP F_1 (English UCCA) and 0.94 F_1 (English EDS). Parsing accuracies in the cross-lingual track present comparable levels of performance, despite limited training data in the case of UCCA and DRG. Furthermore, the evaluation sets for most of the frameworks comprise different text types and subject matters—offering some hope of robustness to domain variation. We expect that these parsers will enable follow-up experimentation on the utility of explicit meaning representation in downstream tasks like, for example, relation extraction, argumentation mining, summarization, or text generation.

Maybe equally importantly, the MRP task design capitalizes on uniformity of representations and evaluation, enabling resource creators and parser developers to more closely (inter)relate representations and parsing approaches across a diverse range of semantic graph frameworks. This facilitates

both quantitative contrastive studies (e.g. the 'post-mortem' analysis by Buljan et al. (2020), which observes that top-performing MRP 2019 parsers have complementary strengths and weaknesses) but also more linguistic, qualitative comparison. General availability of parallel gold-standard annotations over the same text samples—drawing from the WSJ and LPPS corpora—enables side-by-side comparison of linguistic design choices in the different frameworks. This is an area of investigation that we hope will see increased interest in the aftermath of the MRP task series, to go well beyond the impressionistic observations from §3 and ideally lead to contrastive refinement across linguistic schools and traditions.

Despite uniformity in packaging and evaluation, cumulative overall complexity and inherent diversity of the frameworks deemed participation in the shared task a formidable challenge. Of the sixteen teams who participated in MRP 2019, only four teams (predominantly strong performers from before) decided to submit parser outputs in 2020. The two 'newcomer' teams, by comparison, only made partial submissions in the cross-lingual track and ended up not competing for top ranks overall. Similar trends of 'competitive self-selection' and declining participant groups for consecutive instances have been observed with earlier CoNLL shared task and similar benchmarking series. On the upside, with the possible exception of English AMR (where there has been much contemporaneous progress recently), the MRP 2020 empirical results present a strong state-of-the-art benchmark for meaning representation parsing.

On the more foundational question of the relevance of explicit, discrete representations of sentence meaning, the past several years of breakthrough neural advances have been comparatively insensitive to syntactico-semantic structure. In our view, these developments have at least in part been reflective of the stark lack of general techniques for the encoding of hierarchical structure in end-to-end neural architectures. Increased adoption of Graph Convolutional Networks (Kipf and Welling, 2017) and other hierarchical modeling techniques suggest new opportunities for the exploration of both structurally informed end-to-end archictures or e.g. multi-task learning setups. Beyond such ultimately performance-driven research, explicit encoding of syntactico-semantic structure in our view further bears promise in terms of model interpretability and safe-guarding against 'neural meltdown' (e.g. discarding something as foundational as negation or inadvertently altering a date expression in summarization or translation). In a similar vain, meaning representations are being successfully applied in evaluation, e.g. to quantify system output vs. gold standard similarity beyond surface n-grams (Sulem et al., 2018b; Xu et al., 2020b, inter alios).

All technical information regarding the MRP 2019 shared task, including system submissions, detailed official results, and links to supporting resources and software are available from the task web site at:

http://mrp.nlpl.eu

Acknowledgments

Several colleagues have assisted in designing the task and preparing its data and software resources. We thank Dotan Dvir (Hebrew University of Jerusalem) for leading the annotation efforts on UCCA. Dan Flickinger (Stanford University) created fresh gold-standard annotations of some 1,000 WSJ strings, which form part of the EDS evaluation graphs in 2020. Sebastian Schuster (Stanford University) advised on how to convert the gold-standard syntactic annotations from the venerable PTB and OntoNotes treebanks to Universal Dependencies, version 2.x, using 'modern' tokenization. Anna Nedoluzhko and Jiří Mírovský (Charles University in Prague) enhanced the PTG annotation of LPPS data with previously missing items, most notably coreference. Milan Straka (Charles University in Prague) made available an enhanced version of his UDPipe parser and assisted in training Czech, English, and German morpho-syntacic parsing models (for the MRP companion trees). Jayeol Chun (Brandeis University) provided invaluable assistance in conversion of the Chinese AMR annotations, preparation of the Chinese morpho-syntactic companion trees, and provisioning of companion alignments for the English AMR graphs.

We are grateful to the Nordic Language Processing Laboratory (NLPL) and Uninett Sigma2, which provided technical infrastructure for the MRP 2020 task. Also, we warmly acknowledge the assistance of the Linguistic Data Consortium (LDC) in distributing the training data for the task to participants at no cost to anyone.

The work on UCCA and the HUJI-KU submission was partially supported by the Israel Science Foundation (grant No. 929/17). The work

on PTG has been partially supported by the Ministry of Education, Youth and Sports of the Czech Republic (project LINDAT/CLARIAH-CZ, grant No. LM2018101) and partially by the Grant Agency of the Czech Republic (project LUSyD, grant No. GX20-16819X). The work on DRG was supported by the NWO-VICI grant (288-89-003) and the European Union Horizon 2020 research and innovation programme (under grant agreement No. 742204). The work on Chinese AMR data is partially supported by project 18BYY127 under the National Social Science Foundation of China and project 61772278 under the National Science Foundation of China.

References

Omri Abend and Ari Rappoport. 2013. UCCA. A semantics-based grammatical annotation scheme. In *Proceedings of the 10th International Conference on Computational Semantics*, pages 1 – 12, Potsdam, Germany.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Johan Bos, and Stephan Oepen. 2020. DRS at MRP 2020: Dressing up Discourse Representation Structures as graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 23 – 32, Online.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019. The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Ofir Arviv, Ruixiang Cui, and Daniel Hershcovich. 2020. HUJI-KU at MRP 2020: Two transition-based neural parsers. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 73 – 82, Online.

Nicholas Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers.

Ramon Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of EMNLP*.

Miguel Ballesteros and Yaser Al-Onaizan. 2017. AMR parsing using stack-LSTMs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1269–1275, Copenhagen, Denmark. Association for Computational Linguistics.

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, Sofia, Bulgaria.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation. On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239 – 249, London, UK.

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. 2016. HUME. Human UCCA-based evaluation of machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1264 – 1274, Austin, TX, USA.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks. Building and Using Parsed Corpora*, pages 103 – 127. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*. Springer Netherlands.

Maja Buljan, Joakim Nivre, Stephan Oepen, and Lilja Øvrelid. 2020. A tale of three parsers: Towards diagnostic evaluation for meaning representation parsing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1902–1909, Marseille, France. European Language Resources Association.

Deng Cai and Wai Lam. 2019. Core semantic first: A top-down approach for AMR parsing. 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)*, pages 3799–3809, Hong Kong, China. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 1290–1301, Online. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch. An evaluation metric for semantic feature structures. In *Proceedings of the 51th Meeting of the Association for Computational Linguistics*, pages 748 – 752, Sofia, Bulgaria.

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 76 – 85, Hong Kong, China.

Yufei Chen, Yajie Ye, and Weiwei Sun. 2019. Peking at MRP 2019: Factorization- and composition-based parsing for Elementary Dependency Structures. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 166 – 176, Hong Kong, China.

Leshem Choshen and Omri Abend. 2018. Reference-less measure of faithfulness for grammatical error correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, LA, USA.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, 3(4):281 – 332.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual Abstract Meaning Representation parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1146–1155, New Orleans, Louisiana. Association for Computational Linguistics.

Robert M. W. Dixon. 2010/2012. *Basic Linguistic Theory*. Oxford University Press.

Lucia Donatelli, Meaghan Fowlie, Jonas Groschwitz, Alexander Koller, Matthias Lindemann, Mario Mina, and Pia Weißenhorn. 2019. Saarland at MRP 2019: Compositional parsing across all graphbanks. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 66 – 75, Hong Kong, China.

Longxu Dou, Yunlong Feng, Yuqiu Ji, Wanxiang Che, and Ting Liu. 2020. HIT-SCIR at MRP 2020: Transition-based parser and iterative inference parser. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 65 – 72, Online.

Rebecca Dridan and Stephan Oepen. 2011. Parser evaluation using elementary dependency matching. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 225 – 230, Dublin, Ireland.

Jason Eisner. 1997. Bilexical grammars and a cubic-time probabilistic parser. In *Proceedings of the Fifth International Workshop on Parsing Technologies*, pages 54–65, Boston/Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Federico Fancellu, Sorcha Gilroy, Adam Lopez, and Mirella Lapata. 2019. Semantic graph parsing with recurrent neural network DAG grammars. 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)*, pages 2769–2778, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum. 2012. Wordnet. *The Encyclopedia of Applied Linguistics*.

Daniel Fernández-González and Carlos Gómez-Rodríguez. 2020. Transition-based semantic dependency parsing with pointer networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7035–7046, Online. Association for Computational Linguistics.

Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. Sustainable development and refinement of complex linguistic annotations at scale. In Nacy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 353 – 377. Springer, Dordrecht, The Netherlands.

Jan Hajič, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Hajičová, Jiří Havelka, Petr Homola, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Petr Pajas, Jarmila Panevová, Lucie Poláková, Magdaléna Rysová, Petr Sgall, Johanka Spoustová, Pavel Straňák, Pavlína Synková, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Barbora Vidová Hladká, Daniel Zeman, Šárka Zikánová, and Zdeněk Žabokrtský. 2018. Prague dependency treebank 3.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1 – 18, Boulder, Colorado. Association for Computational Linguistics.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3153 – 3160, Istanbul, Turkey.

Eva Hajičová, Barbara Partee, and Petr Sgall. 1998. *Topic–Focus Articulation, Tripartite Structures, and Semantic Content*. Kluwer, Dordrecht, The Netherlands.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver, Canada. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics*, pages 373 – 385, Melbourne, Australia.

Daniel Hershcovich, Zohar Aizenbud, Leshem Choshen, Elior Sulem, Ari Rappoport, and Omri Abend. 2019a. SemEval-2019 task 1. Cross-lingual semantic parsing with UCCA. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1 – 10, Minneapolis, MN, USA.

Daniel Hershcovich, Zohar Aizenbud, Leshem Choshen, Elior Sulem, Ari Rappoport, and Omri Abend. 2019b. SemEval-2019 task 1: Cross-lingual semantic parsing with UCCA. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1–10, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes. The 90% solution. In *Proceedings of Human Language Technologies: The 2006 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 57 – 60, New York City, USA.

Zixia Jia, Youmi Ma, Jiong Cai, and Kewei Tu. 2020. Semi-supervised semantic dependency parsing using CRF autoencoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6795–6805, Online. Association for Computational Linguistics.

Wei Jiang, Zhenghua Li, Yu Zhang, and Min Zhang. 2019. HLT@SUDA at SemEval-2019 task 1: UCCA graph parsing as constituent tree parsing. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 11–15, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1989 – 1993, Las Palmas, Spain.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, Toulon, France.

Marco Kuhlmann and Stephan Oepen. 2016. Towards a catalogue of linguistic graph banks. *Computational Linguistics*, 42(4):819 – 827.

Shuhei Kurita and Anders Søgaard. 2019. Multi-task semantic dependency parsing with policy gradient for learning easy-first strategies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2420–2430, Florence, Italy. Association for Computational Linguistics.

Robin Kurtz, Daniel Roxbo, and Marco Kuhlmann. 2019. Improving semantic dependency parsing with syntactic features. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 12–21, Turku, Finland. Linköping University Electronic Press.

Young-Suk Lee, Ramon Fernandez Astudillo, Tahira Naseem, Revanth Gangi Reddy, Radu Florian, and Salim Roukos. 2020. Pushing the limits of AMR parsing with self-learning. In *Findings of EMNLP*.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2018. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439, Melbourne, Australia. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Chunchuan Lyu, Shay B. Cohen, and Ivan Titov. 2020. A differentiable relaxation of graph segmentation and alignment for AMR parsing.

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407, Melbourne, Australia. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics*, 19:313 – 330.

Jonathan May. 2016. SemEval-2016 Task 8. Meaning representation parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pages 1063 – 1073, San Diego, CA, USA.

Jonathan May and Jay Priyadarshi. 2017. SemEval-2017 Task 9. Abstract Meaning Representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 536 – 545.

James J. McGregor. 1982. Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience*, 12(1):23 – 34.

Seung-Hoon Na and Jinwoo Min. 2020. JBNU at MRP 2020: AMR parsing using a joint state model for graph-sequence iterative inference. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 83 – 87, Online.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Rik van Noord. 2019. Neural Boxer at the IWCS shared task on DRS parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

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

Stephan Oepen, Omri Abend, Jan Hajič, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdeňka Urešová. 2019. MRP 2019: Cross-framework Meaning Representation Parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 1 – 27, Hong Kong, China.

Stephan Oepen and Dan Flickinger. 2019. The ERG at MRP 2019: Radically compositional semantic dependencies. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 40 – 44, Hong Kong, China.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 Task 18. Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 915 – 926, Denver, CO, USA.

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*, pages 63 – 72, Dublin, Ireland.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 1250 – 1255, Genoa, Italy.

Hiroaki Ozaki, Gaku Morio, Yuta Koreeda, Terufumi Morishita, and Toshinori Miyoshi. 2020. Hitachi at MRP 2020: Text-to-graph-notation transducer. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 40 – 52, Online.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics*, pages 2037 – 2048, Vancouver, Canada.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press, Chicago, USA.

Rudolf Rosa, Martin Popel, Ondřej Bojar, David Mareček, and Ondřej Dušek. 2016. Moses & treex hybrid MT systems bestiary. In *Proceedings of the 2nd Deep Machine Translation Workshop*, pages 1 – 10, Lisbon, Portugal. ÚFAL MFF UK.

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53 – 64, Online.

Rob A. Van der Sandt. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4):333 – 377.

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, Portorož, Slovenia. European Language Resources Association (ELRA).

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht, The Netherlands.

Gabriel Stanovsky and Ido Dagan. 2018. Semantics as a foreign language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2412 – 2421, Brussels, Belgium.

Mark Steedman. 2011. *Taking Scope*. MIT Press, Cambridge, MA, USA.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197 – 207.

Milan Straka and Jana Straková. 2020. UDPipe at EvaLatin 2020: Contextualized embeddings and treebank embeddings. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 124–129, Marseille, France. European Language Resources Association (ELRA).

Elior Sulem, Omri Abend, and Ari Rappoport. 2015. Conceptual annotations preserve structure across translations. A French–English case study. In *Proceedings of the 1st Workshop on Semantics-Driven Statistical Machine Translation*, pages 11 – 22.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. Semantic structural annotation for text simplification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, LA, USA.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018c. Simple and effective text simplification using semantic and neural methods. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics*, Melbourne, Australia.

Zdeňka Urešová, Eva Fučíková, and Jana Šindlerová. 2016. CzEngVallex. A bilingual Czech–English valency lexicon. *The Prague Bulletin of Mathematical Linguistics*, 105:17 – 50.

Chuan Wang, Bin Li, and Nianwen Xue. 2018. Transition-based Chinese AMR parsing. 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 247–252, New Orleans, Louisiana. Association for Computational Linguistics.

Chuan Wang and Nianwen Xue. 2017. Getting the most out of AMR parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1257 – 1268, Copenhagen, Denmark.

Xinyu Wang, Jingxian Huang, and Kewei Tu. 2019. Second-order semantic dependency parsing with end-to-end neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4618, Florence, Italy. Association for Computational Linguistics.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020a. Improving AMR parsing with sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Jin Xu, Yinuo Guo, and Junfeng Hu. 2020b. Incorporate semantic structures into machine translation evaluation via UCCA. In *Proceedings of the International Conference on Machine Translation*, Online.

Daniel Zeman and Jan Hajič. 2020. FGD at MRP 2020: Prague Tectogrammatical Graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 33 – 39, Online.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. 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)*, pages 3786–3798, Hong Kong, China. Association for Computational Linguistics.

Yue Zhang, Wei Jiang, Qingrong Xia, Junjie Cao, Rui Wang, Zhenghua Li, and Min Zhang. 2019c. SUDA–Alibaba at MRP 2019: Graph-based models with BERT. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Computational Natural Language Learning*, pages 149 – 157, Hong Kong, China.