Uniform Meaning Representation Parsing as a Pipelined Approach

Jayeol Chun
Brandeis University
415 South Street, Waltham, MA 02453
jchun@brandeis.edu

Nianwen Xue
Brandeis University
415 South Street, Waltham, MA 02453
xuen@brandeis.edu

Abstract

Uniform Meaning Representation (UMR) is the next phase of semantic formalism following Abstract Meaning Representation (AMR), with added focus on inter-sentential relations allowing the representational scope of UMR to cover a full document. This, in turn, greatly increases the complexity of its parsing task with the additional requirement of capturing document-level linguistic phenomena such as coreference, modal and temporal dependencies. In order to establish a strong baseline despite the small size of recently released UMR v1.0 corpus, we introduce a pipeline model that does not require any training. At the core of our method is a two-track strategy of obtaining UMR's sentence and document graphs separately, with the document-level triples being compiled at the token level and the sentence graph being converted from AMR graphs. By leveraging alignment between AMR and its sentence, we are able to generate the first automatic English UMR parses.

1 Introduction

While the end-to-end deep learning methods based on transformers (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019) helped usher in an era of Large Language Models (LLM) with outstanding results especially in the practical domains of Natural Language Processing (NLP), they also brought about significant advances in the performance of Abstract Meaning Representation (AMR) parsing. Once thought extremely challenging due to its inherently multi-tasking nature, AMR parsing with its adoption of transformer architecture (Bevilacqua et al., 2021; Lee et al., 2022a; Vasylenko et al., 2023) has since matured to a point where its automatic parses feature in various downstream applications (Bonial et al., 2020; Mansouri et al., 2023; Wang et al., 2023), often as a meaningful companion to the Pre-trained Language Models (PLM) llke T5 (Raffel et al., 2019) or BART (Lewis et al., 2020). This trend serves to highlight the enduring interest of the community in leveraging symbolic meaning representations not only for the computational benefit in boosting the model performance but also as a way to better understand how a model seems to 'understand' language.

However, AMR by design is limited to the representational scope of a single sentence. Although efforts have been made to bring together multiple AMRs into a single unified structure (O'Gorman et al., 2018; Naseem et al., 2022), additional annotations across different sentences remain largely confined to coreference and implicit role labeling.

Uniform Meaning Representation

In contrast, Uniform Meaning Representation (UMR) (Van Gysel et al., 2021) begins by inheriting AMR's focus on predicate-argument structure in its sentence-level representation and further adds semantic coverage for aspect, scope, person and number for cross-lingual compatibility (Flanigan et al., 2022; Bonn et al., 2023b). In addition, UMR introduces new document-level triples which cover linguistic phenomena such as coreference, modal and temporal dependencies (Vigus et al., 2019; Zhang and Xue, 2018a; Yao et al., 2022) that potentially go beyond sentence boundaries.

Figure 1 provides an example of UMR annotation for a sample document of two sentences:

1. Kim left to join the others.
2. "They are probably eating," she said.

At the top is an abstract `ROOT` node, whose immediate children `AUTHOR` (author of the text) and `DCT` (document creation time) serve as sub-roots of modal and temporal dependencies respectively. These abstract nodes are highlighted in lightblue.

A modal dependency graph (MDG), shown as a series of red edges in the figure, captures the epistemic certainty and polarity with which

Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing, pages 40–52

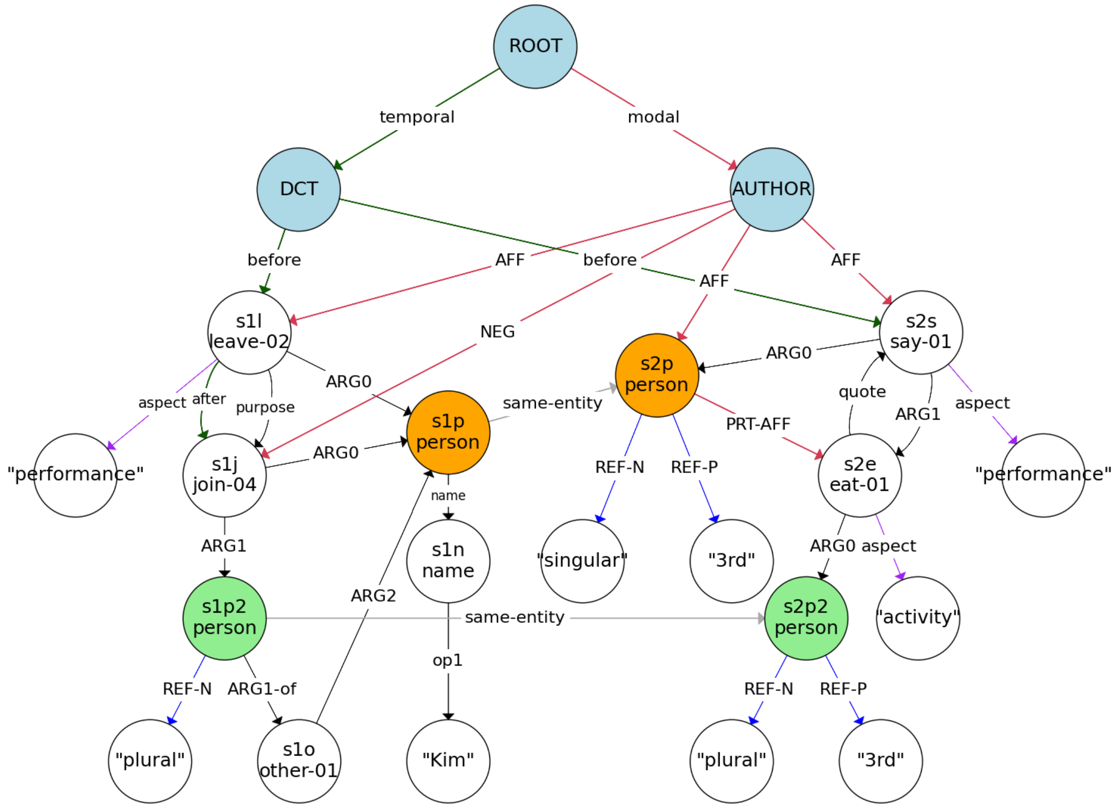

Figure 1: Example of UMR for "Kim left to join the others. 'They are probably eating,' she said." Lightblue nodes indicate special semantic nodes ROOT, AUTHOR and DCT (Document Creation Time) that are implied in every document. Modal relations are shown in red edges, temporal relations in green edges, and the clusters of coreferent entities are highlighted in the same color such as orange and green. AFF stands for full-affirmative, NEG for full-negative, PRT-AFF for partial-affirmative, REF-N for refer-number, and REF-P for refer-person.

the sources (formally known as *conceivers*) view another conceivers or events (Yao et al., 2021; Van Gysel et al., 2021). In the example, the Author knows with full certainty that *Kim* already *left* (:full-affirmative edge from the Author to s1l:leave-02), while *Kim* expresses uncertainty in her belief that *They* are *eating* at the moment (:partial-affirmative edge from s2p:person to s2e:eat-01). Since the Author presumably knew about *Kim*'s state of mind, a :full-affirmative modal relation between these two sources is finally established.

On the other hand, a temporal dependency graph (TDG) represents the temporal relations between events and time expressions such as DCT (Zhang and Xue, 2018b). The past tense of the main predicates *left* and *said* in the above example provides a strong indication of the actions having taken place before DCT, hence its two :before outgoing edges to s1l:leave-02 and s2e:eat-01. Furthermore, the chain of events dictates that *Kim* could not have possibly *joined* the others without having first *left*. This is annotated with the :after edge from s1l:leave-2 to s1j:join-04, which adds the temporal aspect to the :purpose relation that already exists between the two. Following the green edges in the figure reveals the temporal graph in its entirety.

Finally, the two sentences are further linked via the participation of same entities: *Kim* and *the others*. Their presence in the second sentence solely as pronouns, *she* and *they*, requires the context from the first sentence for anaphora resolution. These clusters of coreferent entities are highlighted as the same colored nodes in the figure connected by :same-entity edges between (1) s1p:person and s2p:person, and (2) s1p2:person and s2p2:person.

It is also worth noting the core differences between UMR sentence graphs and AMRs despite

Document ID	Sentences	Doc. Level	Tokens	AMR R3 Overlaps
english_umr-0001	28	28	700	NW_AFP_ENG_0024_2006_0217.[1~28]
english_umr-0002	2	28	18	-
english_umr-0003	9	9	140	NW_PRI_ENG_0153_2000_1214.[1~9]
english_umr-0004	141	135	1,165	-
english_umr-0005	29	29	566	NW_PRI_ENG_0152_2000_1208.[1~29]
Total	209	203	2,589	66

Table 1: UMR v1.0 English dataset statistics. *Doc. Level* refers to the number of non-empty document-level graphs. *R3 Overlaps*, if any, displays the AMR ids from AMR R3 corpus that share the same source sentence with UMRs.

their striking similarities. One of the notable discrepancies is the addition of `:aspect` annotations in UMR, visualized as purple edges in Figure 1, representing the internal state of an eventive concept as it relates to its status as an on-going, finished or habitual event, or simply a state with no changes over the course of action, or something else[1] (Donatelli et al., 2018, 2019). In the figure, *Kim* having *left* and *said* had already come to an end ("`performance`"), whereas *eating* is presumably still an on-going process at the time of writing ("`activity`").

Finally, pronouns in AMRs are replaced with generic `person` nodes with `:refer-person` edges denoting first, second or third point of view. Generic, non-named entities, including pronouns, are further annotated for their plurality with `:refer-number` relations, as seen with the blue outgoing edges from variables `s1p2:person`, `s2p:person` and `s2p2:person` in Figure 1.

UMR Parsing

While these new features help expand the representational scope of UMR to include a full document, they come at a great cost to the parsing complexity. In addition to the sentence graph generation, a parser would have to produce an additional document-level structure whose scope generally encompasses multiple sentences. Since the triples in the document graph need to be grounded in the context of the sentence graphs (Figure 1), the parsing task effectively revolves around a series of pairwise relation classifications between sentence graph nodes that have been abstracted away from their source tokens, much like AMRs. This is further complicated by the limited number of publicly available annotations in the recently released UMR v1.0 corpus[2] (Bonn et al., 2023a, 2024) .

In light of these challenges, we propose to settle for a more tractable version of the problem that does not require any training. Our approach adopts a two-track strategy of obtaining sentence and document graphs separately. This is possible if we obtain the document-level triples at the token level, i.e., between the source tokens, *not* between the sentence graph nodes. By leveraging models individually trained for each of the document-level parsing tasks, we can set up a pipeline that compiles a list of document-level triples without any training on the limited UMR corpus. At the same time, we rely on off-the-shelf AMR parsers to first generate AMR, which is then subsequently converted into the UMR sentence graph using linguistically motivated heuristics. The final step involves the alignment of source tokens in the document-level triples to their corresponding nodes in the sentence graph, resulting in the final UMR structure.

The performance of our pipelined model is evaluated against the entire English section of the UMR v1.0 corpus, using a recently introduced AnCast++[3] whose details are provided in Section 5. We report the highest comprehensive macro F1 score at **61.5**, establishing a strong baseline for future improvement. The code is available at `https://github.com/umr4nlp/umrlib`.

2 UMR-v1.0 Corpus

UMR v1.0 corpus consists of documents annotated in 6 languages: Arapaho, Chinese, Cocama-Cocamilla, English, Navajo, and Sanapaná[4]. This work focuses only on 5 English documents, whose summary statistics are

[1]See umr-guidelines for the full lattice of aspectual values.

[2]`https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-5198`

[3]`https://github.com/sxndqc/ancast`

[4]`https://umr4nlp.github.io/web/data.html` shows the number of annotations for all 6 languages.

given in Table 1. The entire newsire domain (`english_umr-0001`, `english_umr-0003`, and `english_umr-0005`) overlaps with the LDC's latest release of AMR R3 corpus LDC2020T02[5] (Knight et al., 2021). Each sentence receives 2 core layers of annotations: (1) sentence graph and (2) document-level triples involving at least one local variable from its sentence graph.

Corpus Preprocessing

The corpus exhibits a few labeling inconsistencies. For instance, there are 12 occurrences of `:AFF` abbreviated modal relation label in addition to the more established `:full-affirmative` at 324. We attribute these and other similar occurrences to be simple errors and apply a cleanup to ensure labeling consistency across all of the annotations, e.g., `:AFF` replaced with `:full-affirmative`.

In addition, the `:modal-strength` relation (sometimes abbreviated as `:modstr`) is used as a shorthand to annotate a modal triple *within* a UMR sentence graph, although modal triples typically belong to a document-level annotation. In order to facilitate correct evaluation in our parsing experiments as required by AnCast++, these embedded modal triples are relocated from the sentence graph to its document-level annotation. It should be noted that this operation does not modify the content of the original annotation. We report parsing performance results with and without these procedures.

3 Model Description

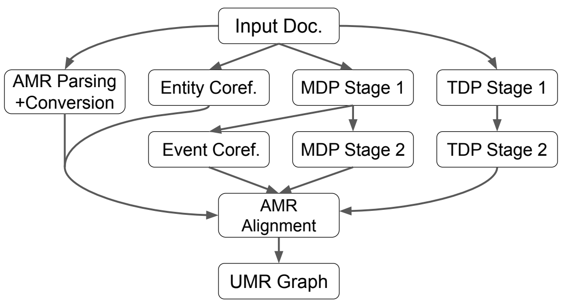

Figure 2: Flowchart for the proposed pipelined parser. `MDP` stands for Modal Dependency Parsing and `TDP` stands for Temporal Dependency Parsing.

In this section, we provide a detailed description of each of the models that makes up our pipeline. The entire flowchart is depicted in Figure 2.

[5] `https://catalog.ldc.upenn.edu/LDC2020T02`

3.1 AMR Parsing

AMR parsing aims to transform text into AMR where the meaning of a sentence is encoded in a single-rooted, directed and acyclic graph, as partially seen with the two sentence graphs in Figure 1 rooted by variables `s1l` and `s2s` whose black edges reveal the predicate-argument structure of each sentence. Due to its graphical nature, previous parsing methods often adopted graph methods such as finding the maximum spanning AMR graph (Flanigan et al., 2014, 2016), while others exploited the structural similarity between AMR and a dependency graph by applying a series of actions to transform the dependency graph into AMR in a transition-based framework (Wang et al., 2015, 2016; Wang and Xue, 2017). These approaches were largely superseded by larger models that began to pivot around various deep learning-based approaches (Foland and Martin, 2017; Lyu and Titov, 2018; Cai and Lam, 2020), culminating in the adoption of transformers (Bevilacqua et al., 2021). The subsequent advancements in AMR parsing relied on pretrained language models to consume and predict linearized AMRs (Chen et al., 2022; Bai et al., 2022; Yu and Gildea, 2022; Vasylenko et al., 2023), and the linearized representation of AMRs further opened up the possibility of a transition-based approach where a sequence of transductions are interpreted graphically to incrementally build towards the final AMR graph (Zhou et al., 2021b,a; Drozdov et al., 2022).

Given the efficacy of transformers-based AMR parsers, along with the unmistakable similarity of AMR to the UMR sentence graph, it is only natural to choose AMR parsing as a starting point of the pipeline. We experiment with four AMR parsers: LeakDistill (Vasylenko et al., 2023), SPRING (Bevilacqua et al., 2021), AMRBART (Bai et al., 2022) and IBM Transition Parser (Zhou et al., 2021b,a; Lee et al., 2022b; Drozdov et al., 2022). Maximum Bayes Smatch Ensemble (MBSE) (Lee et al., 2022b) is additionally used to ensemble best performing parsers for further improvement. Experiments using BLINK (Ledell Wu, 2020) entity linker for Wikification did not improve the model performance and is thus omitted in our experimental setup. Finally, we run LEAMR (Blodgett and Schneider, 2021) to produce sentence-AMR alignment for subsequent use in AMR-to-UMR conversion. Appendix A provides more details on the setup used in our experiments.

AMR Parser	Before Conversion		After Conversion	
	AnCast	Smatch	AnCast	Smatch
LeakDistill (Vasylenko et al., 2023)	51.3	56.7	63.2	71.3
SPRING (Bevilacqua et al., 2021)	51.1	56.4	62.9	71.2
Struct-BART (Zhou et al., 2021b)	49.3	56.0	60.9	70.6
AMRBART (Bai et al., 2022)	51.4	57.0	63.0	71.7
3-way MBSE* (Lee et al., 2022b)	51.3	57.2	63.1	71.8
4-way MBSE†	**52.6**	**57.5**	**64.2**	**72.2**
5-way MBSE‡	52.1	57.4	64.1	71.9

Table 2: Results on AMR-to-UMR Sentence Graph Conversion. *3-way MBSE includes LeakDistill + SPRING + AMRBART. †4-way MBSE includes LeakDistill + SPRING + AMRBART + Struct-BART. ‡5-way MBSE includes LeakDistill + SPRING + AMRBART (2 checkpoints) + Struct-BART.

3.2 AMR-to-UMR Conversion

Once an AMR parse is obtained, we apply heuristics for in-place conversion to the UMR sentence graph based on the mapping methodology described in Bonn et al. (2023b) and UMR guidelines[6]. We notice a few minor discrepancies between the methodology and some of the annotations in UMR v1.0; for instance, the guidelines advocates for `:ref-person` label whereas the corpus prefers `:refer-person`. In cases like this, we choose to follow the corpus for consistent parsing evaluation. A more recent work on AMR-to-UMR conversion provides fine-grained, nondeterministic mapping strategies based on the graph context (Post et al., 2024) but was not consulted for this work.

One of the practical challenges in AMR-to-UMR conversion is the `:aspect` edge creation task for events. Its heavily context-dependent nature makes it difficult to reliably determine its child node label—i.e., aspectual value—via heuristics. For this reason, we seek the help of Universal Dependency-style syntactic analysis from UDPipe v2 (Straka, 2018) whose UD features such as `Tense` and `Verbform` provide limited but helpful insights. The distribution of the aspect labels from the corpus is shown in Table 3.

Aspect	Count
Performance	184
State	146
Activity	55
Endeavor	17
Process	16
Habitual	8
Total	426

Table 3: Distribution of the aspectual values in UMR v1.0 English dataset.

Another important aspect of conversion is handling of the non-named entities including the pronouns. Their ubiquitous presence makes it a high-priority sub-task, and here again we rely on UD features from which we are able to infer the plurality of any generic entity.

Table 2 provides the overall results with AMR parsers and subsequent in-place conversion to UMR sentence graph, using Smatch (Cai and Knight, 2013) and AnCast[7] (Sun and Xue, 2024). AnCast is a recently introduced metric for evaluating graph-based meaning representations whose alignment strategy differs from the hill-climbing heuristics of Smatch by first identifying anchor nodes based on content similarity, and then iteratively propagating alignment throughout the neighborhood. It finally computes the labeled relation F1 score which measures the degree of matching for concepts and relations. This value represents the overall metric of AnCast and is reported in Table 2.

3.3 Modal Dependency Parsing

Modal dependency parsing (MDP) aims to construct a hierarchical structure representing the epistemic strength (`full`, `neutral` and `partial`) and polarity (`affirmative` and `negative`) of conceivers as related to other conceivers or events (Yao et al., 2021; Van Gysel et al., 2019). Largely based on FactBank (Saurí and Pustejovsky, 2009), UMR modal dependency sub-structure combines 3 modal strengths with 2 polarities as shown in Table 4. As

[6] https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

[7] not to be confused with AnCast++ whose details are provided in Section 5.

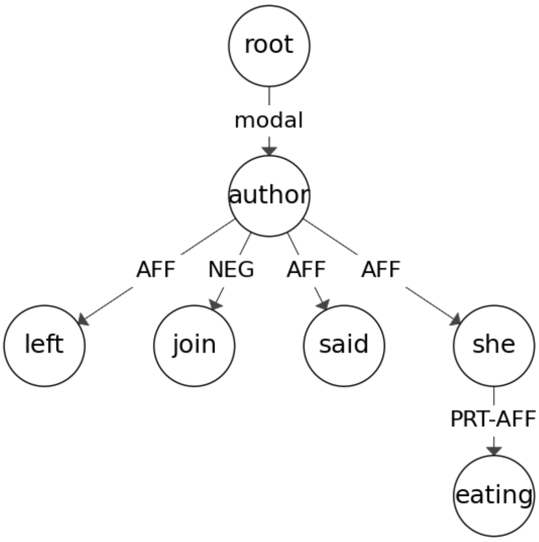

Figure 3: Example of Modal Dependency Graph for "Kim **left** to **join** the others. 'They are *probably* **eating**,' she **said**." `AFF` stands for `full-affirmative`, `NEG` for `full-negative`, and `PRT-AFF` for `partial-affirmative`.

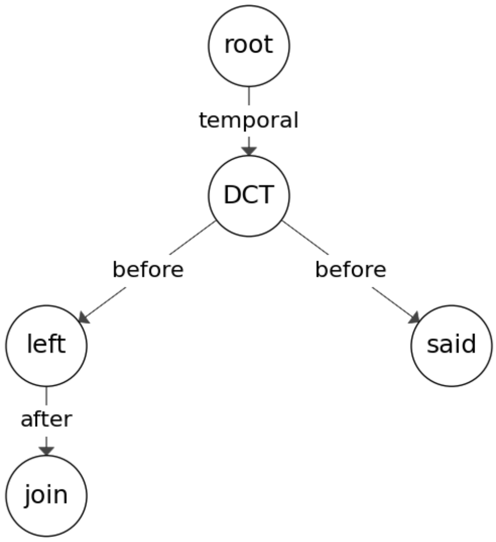

Figure 4: Example of Temporal Dependency Graph for "Kim **left** to **join** the others. 'They are probably eating,' she **said**."

seen with Figure 3, the resulting graph typically involves heavy traffic through the `Author` who displays confidence or doubt in various statements s/he commits to in writing.

Modal Label	Count
`:full-affirmative`	408
`:neutral-affirmative`	24
`:partial-affirmative`	14
`:full-negative`	23
`:neutral-negative`	3
`:partial-negative`	3
`:unspecified*`	10
Total	486

Table 4: Distribution of modal labels in UMR v1.0 English dataset. *UMR v1.0 corpus contains `:unspecified` which is not part of the target modal labels in MDP.

In practice, MDP consists of two different stages. First, the conceivers and events must be identified; then, each event or conceiver must be paired with the most appropriate parent in the text in a newly-created modal triple whose label needs to be predicted. In our experiments, we use a prompt-based model described in Yao et al. (2022), where the two tasks are trained end-to-end in a joint manner based on language model priming. Table 4 shows the distribution of modal labels in the UMR English corpus.

3.4 Temporal Dependency Parsing

In a similar vein to MDP, temporal dependency parsing (TDP) is the task of identifying a document-level graph whose nodes are time expressions (timex) and events, and edges represent the temporal relations between them. Specifically, an event first searches for its referent timex that is the most specific (i.e., closest) temporal anchor (Pustejovsky and Stubbs, 2011) in whose absence it settles for another event that can provide the most specific temporal context (Zhang and Xue, 2018b; Yao et al., 2020). Figure 4 depicts a temporal dependency graph for the sample document of two sentences.

TDP also consists of 2 stages. The timex and event identification is performed first, followed by edge generation between the identified nodes. For stage 1 we use the neural ranking model described in Yao et al. (2020) based on Zhang and Xue (2018a) and Ross et al. (2020) that extracts timex and events by labeling the appropriate span in the text[8]. Then we turn to the parser from Yao et al. (2023) which interprets TDP as a textual entailment (NLI) task in which the temporal relation is verbalized into text, requiring the model to infer entailment probability. Table 5 shows the distribu-

[8]We observed higher performance when TDP stage 1 output is augmented with events from MDP stage 1.

tion of temporal labels in the corpus.

Temporal Label	Count
overlap	143
after	106
before	54
contained	24
depends-on	7
Total	334

Table 5: Distribution of temporal labels in UMR v1.0 English dataset.

3.5 Coreference

UMR supports two types of coreference—event and entity—which form disjoint clusters. Both may additionally participate in the `:subset-of` relationship. Table 6 provides the number of coreference labels in the corpus.

Event Coreference

For cross-sentence event clustering, our pipeline relies on Cross-Document Coreference Resolution (CDLM) described in Cattan et al. (2021), which is pre-trained to include multiple documents by leveraging global attention. Although the model is designed with cross-document context in mind, we limit the global range to a single document. Since it requires event candidates be provided as input, we re-use the events identified in MDP stage 1.

Entity Coreference

For entities we use wl-coref (Dobrovolskii, 2021) and caw-coref (D'Oosterlinck et al., 2023) which attempt to build a coreference link between individual words.

Coref. Label	Count
same-entity	317
same-event	62
subset-of	55
Total	434

Table 6: Distribution of coreference labels in UMR v1.0 English dataset.

3.6 Context Grounding via Alignment

So far, the pipeline has produced two distinct structures—a sentence graph as a result of AMR-to-UMR conversion, and document-level triples from MDP, TDP and coreference—that are seemingly independent from each other. This is because the sentence graph is generated by transforming an AMR parse whose nodes have been abstracted away from their source tokens, whereas the document-level triples obtained from MDP, TDP and coreference are expressed as between these source tokens.

In order to bring these structures together, the final step of our pipelined approach involves the use of the alignment between the sentence graph and the source sentence provided by LEAMR[9] to map the tokens in document-level triples to the corresponding nodes in the UMR sentence graph. This effectively means transferring the context of the document-level triples from the source sentence to the UMR sentence graph, and only after this stage do these structures demonstrate cohesion as required for UMR.

4 Experiments

We follow the flowchart in Figure 2 to generate UMR parses. Appendix A provides details on the experimental setup. Our model is evaluated against all of the English section of UMR v1.0 corpus. In order to cope with the input length limitation of some of the pipeline models, `english_umr-0004` is split into smaller fragments each of which is treated as a separate document. The intermediate results for the split data are pieced together at the end into a single document for evaluation. Table 7 shows the experimental results using AnCast++ evaluation which we introduce in the next section.

5 Evaluation

Currently, there is no published work that can evaluate the performance of a UMR parser. To this end, we first provide Smatch scores for the sentence graphs evaluation in Table 2. Since the UMR sentence graphs resemble AMRs, Smatch can continue to provide a meaningful and comparable evaluation score during the transition towards UMR.

For the full UMR evaluation we adopt AnCast++[10], a recently introduced open-source evaluation toolkit for UMR that provides an aggregated metric of Sentence, Modal, Temporal and Coreference scores. The Sentence graph evaluation is based on AnCast and is claimed to be highly correlated with Smatch despite differences in the align-

[9] LEAMR provides AMR-to-sentence alignment, which is preserved during the in-place conversion.

[10] `https://github.com/sxndqc/ancast`

Document ID	AnCast++ F1 Scores				
	Sentence Graph	Modal	Temporal	Coref.	Comprehensive
`english_umr-0001`	69.2 (66.2)	51.4 (40.2)	15.6 (16.2)	8.2 (8.2)	57.9 (55.5)
`english_umr-0002`	90.0 (90.0)	60.0 (60.0)	100.0 (100.0)	0.0* (0.0)	86.2 (86.2)
`english_umr-0003`	75.3 (71.8)	70.0 (53.9)	16.9 (18.2)	58.3 (40.0)	68.6 (63.4)
`english_umr-0004`	61.2 (60.7)	64.5 (65.3)	22.8 (22.8)	26.7 (26.7)	52.1 (51.9)
`english_umr-0005`	55.3 (55.0)	13.8 (12.3)	6.3 (7.3)	19.5 (20.4)	42.8 (42.9)
Macro F1	70.2 (68.8)	52.0 (46.3)	32.3 (32.9)	22.5 (19.1)	**61.5** (60.0)

Table 7: Parsing Evaluation Results on UMR v1.0 English corpus using AnCast++. Scores within the parenthesis are from evaluating against the UMR corpus without any preprocessing. *`english_umr-0002` contains no coreference.

ment strategy (Sun and Xue, 2024). While the Modal score is based on the number of overlaps in the modal triples owing to its inherently tree structure, Temporal and Coreference scores require finding the transitive closures via Depth-First Search (DFS) in order to identify clusters of nodes and relations, from which precision and recall measures are computed in terms of closed sets as follows:

$$p = \frac{\sum_{r_i \in R}(|r_i| \times \sum_{k_j \in K} \frac{rel(r_i \cap k_j)}{rel(r_i)})}{\sum_{r_z \in R} |r_z|}$$

$$r = \frac{\sum_{k_i \in K}(|k_i| \times \sum_{r_j \in R} \frac{rel(k_i \cap r_j)}{rel(k_i)})}{\sum_{k_z \in K} |k_z|}$$

where k_i and r_i are node clusters in key (gold) and response (prediction) graphs, and $rel(k_i)$ and $rel(r_i)$ are the reference and deducted links respectively. This approach builds on Setzer et al. (2005) and Link-based Entity-Aware (LEA) metric (Moosavi and Strube, 2016; Moosavi, 2020).

6 Error Analysis

As a pipeline model, our parser is prone to error propagation when generating document-level triples. This is especially true with the event identification phase in MDP and TDP stage 1, where the identified event candidates are subsequently considered for the modal and temporal dependency edge generation as well as cross-sentence event coreference. Naturally, any event that goes undetected is non-recoverable in the subsequent pipeline. This is further compounded by the fact that the generated triples ultimately need to be aligned to the appropriate UMR sentence sub-graph but may be un-aligned or mis-aligned, resulting in low performance on the document-level parsing tasks. Nevertheless, MDP appears to show comparatively stronger performance because MDG is inherently a tree unlike TDG and coreference clusters, with most of traffic consolidated around the `Author`.

The parser is also unable to guarantee 100% coverage of UMR as it is unable produce certain labels such as "Habitual" aspectual value and "`:partial-negative`" modal label. Another prominent example is "`:subset-of`" coreference label which makes up a sizable portion of coreference labels (Table 6), and its lack thereof carries significant repercussions for overall parsing performance. This is to be expected as none of the models are directly trained on the UMR-style of annotations, and it remains a major source of error in our experiments.

The corpus itself shows highly varied annotation styles across different documents. For instance, English UMR documents 1, 2 and 4 consistently annotate `:modal` relation from `ROOT` to `AUTHOR`, although its presence is implied in every document and is not strictly necessary—a view taken in documents 3 and 5. `english_umr-0005` further stands out as what initially appears to be a news article abruptly turns into a dialogue, leading to subsequent sentence graphs being wrapped under `(s / say-01 :ARG0 (p / person) :ARG1 ...)` 'phantom' outer sub-graph. This explains the comparatively low score for the document.

7 Conclusions

This paper presents the first published UMR parsing model evaluated against UMR v1.0 English corpus using AnCast++. We describe our pipelined approach to cope with the shortage of publicly available UMR data so that no training on the UMR corpus is necessary. Our experimental results at 61.5 macro F1 establishes a strong baseline for future improvement. The proposed parser is suitable for modular upgrade by optimizing individual models, which we plan to visit in future work.

Limitations

Due to the small number of UMR data available for evaluation, current parsing result is not yet stable. UMR English dataset further shows highly skewed distribution of number of sentences per document—as small as 2 for `english_umr-0002` and over 140 for `english_umr-0004` which is not taken into account by AnCast++. Increased number of UMR annotations will partially mitigate this issue.

The proposed UMR parser uses sub-models trained in English and is unable to parse any other languages. To apply this model in a cross-lingual setting depends on the availability of models such as temporal dependency parser being trained either multi-lingually or on non-English datasets.

Since the pipeline consists of multiple models each of which may require a different set of dependencies, the parser is difficult to set up for use in practice. We therefore provide a WebUI version of our parser which serves as a one-stop interface to interact with every component in the pipeline.

Acknowledgments

This work is supported by grants from the CNS Division of National Science Foundation (Awards no: NSF_2213804) entitled "Building a Broad Infrastructure for Uniform Meaning Representations". Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of NSF.

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of AMR alignments. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3310–3321, Online. Association for Computational Linguistics.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Julia Bonn, Chen Ching-wen, James Andrew Cowell, William Croft, Lukas Denk, Jan Hajič, Kenneth Lai, Martha Palmer, Alexis Palmer, James Pustejovsky, Haibo Sun, Rosa Vallejos Yopán, Jens Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2023a. Uniform meaning representation. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023b. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. *CoRR*, abs/2004.05572.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online. Association for Computational Linguistics.

Liang Chen, Peiyi Wang, Runxin Xu, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2022. ATP: AMRize then parse! enhancing AMR parsing with PseudoAMRs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2482–2496, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Agata Cybulska and P. Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *International Conference on Language Resources and Evaluation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lucia Donatelli, Nathan Schneider, William Croft, and Michael Regan. 2019. Tense and aspect semantics for sentential amr. pages 346–348.

Karel D'Oosterlinck, Semere Kiros Bitew, Brandon Papineau, Christopher Potts, Thomas Demeester, and Chris Develder. 2023. CAW-coref: Conjunction-aware word-level coreference resolution. In *Proceedings of The Sixth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2023)*, pages 8–14, Singapore. Association for Computational Linguistics.

Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramon Fernandez Astudillo. 2022. Inducing and using alignments for transition-based amr parsing. *Preprint*, arXiv:2205.01464.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.

Jeffrey Flanigan, Ishan Jindal, Yunyao Li, Tim O'Gorman, Martha Palmer, and Nianwen Xue. 2022. Meaning representations for natural languages: Design, models and applications. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 1–8, Abu Dubai, UAE. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

William Foland and James H. Martin. 2017. Abstract Meaning Representation parsing using LSTM recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–472, Vancouver, Canada. Association for Computational Linguistics.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2021. Abstract meaning representation (amr) annotation release 3.0.

Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Young-Suk Lee, Ramón Astudillo, Hoang Thanh Lam, Tahira Naseem, Radu Florian, and Salim Roukos. 2022a. Maximum Bayes Smatch ensemble distillation for AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5379–5392, Seattle, United States. Association for Computational Linguistics.

Young-Suk Lee, Ramon Fernandez Astudillo, Thanh Lam Hoang, Tahira Naseem, Radu Florian, and Salim Roukos. 2022b. Maximum bayes smatch ensemble distillation for amr parsing. *Preprint*, arXiv:2112.07790.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

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.

Behrooz Mansouri, Ricardo Campos, and Adam Jatowt. 2023. Towards timeline generation with abstract meaning representation. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 1204–1207, New York, NY, USA. Association for Computing Machinery.

Nafise Sadat Moosavi. 2020. Robustness in coreference resolution.

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany. Association for Computational Linguistics.

Tahira Naseem, Austin Blodgett, Sadhana Kumaravel, Tim O'Gorman, Young-Suk Lee, Jeffrey Flanigan, Ramón Astudillo, Radu Florian, Salim Roukos, and Nathan Schneider. 2022. DocAMR: Multi-sentence AMR representation and evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3496–3505, Seattle, United States. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Claire Benet Post, Marie C. McGregor, Maria Leonor Pacheco, and Alexis Palmer. 2024. Accelerating UMR adoption: Neuro-symbolic conversion from AMR-to-UMR with low supervision. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 140–150, Torino, Italia. ELRA and ICCL.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160, Portland, Oregon, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Hayley Ross, Jonathon Cai, and Bonan Min. 2020. Exploring Contextualized Neural Language Models for Temporal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8548–8553, Online. Association for Computational Linguistics.

Roser Saurí and James Pustejovsky. 2009. Factbank: A corpus annotated with event factuality. *Language Resources and Evaluation*, 43:227–268.

Andrea Setzer, Robert Gaizauskas, and Mark Hepple. 2005. Using semantic inferences for temporal annotation comparison. In *The Language of Time: A Reader*, pages 575–584.

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, Brussels, Belgium. Association for Computational Linguistics.

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*.

Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *KI - Künstliche Intelligenz*, 35(3):343–360.

Jens E. L. Van Gysel, Meagan Vigus, Pavlina Kalm, Sook-kyung Lee, Michael Regan, and William Croft. 2019. Cross-linguistic semantic annotation: Reconciling the language-specific and the universal. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 1–14, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Pavlo Vasylenko, Pere-Lluís Huguet Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli. 2023. Incorporating graph information in transformer-based amr parsing. *Preprint*, arXiv:2306.13467.

Meagan Vigus, Jens E. L. Van Gysel, and William Croft. 2019. A dependency structure annotation for modality. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 182–198, Florence, Italy. Association for Computational Linguistics.

Chuan Wang, Sameer Pradhan, Xiaoman Pan, Heng Ji, and Nianwen Xue. 2016. CAMR at SemEval-2016 task 8: An extended transition-based AMR parser. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1173–1178, San Diego, California. 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. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado. Association for Computational Linguistics.

Cunxiang Wang, Zhikun Xu, Qipeng Guo, Xiangkun Hu, Xuefeng Bai, Zheng Zhang, and Yue Zhang. 2023. Exploiting Abstract Meaning Representation for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2083–2096, Toronto, Canada. Association for Computational Linguistics.

Jiarui Yao, Steven Bethard, Kristin Wright-Bettner, Eli Goldner, David Harris, and Guergana Savova. 2023. Textual entailment for temporal dependency graph parsing. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 191–199, Toronto, Canada. Association for Computational Linguistics.

Jiarui Yao, Haoling Qiu, Bonan Min, and Nianwen Xue. 2020. Annotating Temporal Dependency Graphs via Crowdsourcing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5368–5380, Online. Association for Computational Linguistics.

Jiarui Yao, Haoling Qiu, Jin Zhao, Bonan Min, and Nianwen Xue. 2021. Factuality assessment as modal dependency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1540–1550, Online. Association for Computational Linguistics.

Jiarui Yao, Nianwen Xue, and Bonan Min. 2022. Modal dependency parsing via language model priming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2913–2919, Seattle, United States. Association for Computational Linguistics.

Chen Yu and Daniel Gildea. 2022. Sequence-to-sequence AMR parsing with ancestor information. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 571–577, Dublin, Ireland. Association for Computational Linguistics.

Yuchen Zhang and Nianwen Xue. 2018a. Neural ranking models for temporal dependency structure parsing. *CoRR*, abs/1809.00370.

Yuchen Zhang and Nianwen Xue. 2018b. Structured interpretation of temporal relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, and Radu Florian. 2021a. AMR parsing with action-pointer transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5585–5598, Online. Association for Computational Linguistics.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021b. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Experimental Setup

Experiments were run on NVIDIA RTX 3090.

AMR Parser

We found 4-way and 5-way MBSE models to produce the highest Smatch and AnCast scores on UMR sentence graphs evaluation (Table 2). We were also able to obtain the highest AnCast++ scores on full UMR evaluation using 5-way MBSE (Table 7). These include:

1. LeakDistill trained on AMR R3[11].
2. SPRING trained on AMR R3.
3. Struct-BART trained on AMR R3 and parsed using ensemble of 3 seeds: 42, 43, and 44.

[11] Checkpoint 'best-smatch_checkpoint_12_0.8534' is available upon request to the authors

4. AMRBART 3.0 trained on AMR R3.
5. AMRBART 2.0 trained on AMR R2 (not part of 4-way MBSE).

We do not run the BLINK entity linking system in our pipeline.

Modal Dependency Parsing

mdp-prompt (Yao et al., 2022) is the prompt-based modal dependency parser trained on publicly available English modal dependency dataset[12] (Yao et al., 2021). We exactly follow the training configurations described in the paper for English.

Temporal Dependency Parsing

Unlike MDP where a single parser can perform stage 1 and stage 2 jointly, we train two separate models since the best stage 2 parser does not produce stage 1 outputs.

TDP Stage 1

To identify events and timex, we use the XLM-Roberta (Conneau et al., 2020) based ranking model (Yao et al., 2020) whose source code is not publicly available but is similar to that of mdp-prompt.

The model is trained on publicly available English temporal dependency dataset[13] for 30 epochs with learning rate of 2e-5 and max sequence length of 128. The model processes a long document by splitting it into smaller segments before encoding each with the language model. When doing so, we allow the model to apply segmentation by letting each overlap with one another. These procedures are in accordance with what is described in the paper.

In practice, the identified events are merged with those found by mdp-prompt, leading to better results. Finally, the merged events also serve as inputs to CDLM for event coreference.

TDP Stage 2

thyme-tdg (Yao et al., 2023) is trained following the model implementation details as specified for the general-domain experiments, but we allow training to last for 10 epochs rather than 3. We use seed 42 for data preparation as well as model training.

In practice, we find that the ranking model (Yao et al., 2020) should also be trained for stage 2 event-to-time and event-to-event edge generation task, whose outputs are then fed to thyme-tdg. In both scenarios the hyperparameters remain the same as described in the paper.

Coreference

CDLM for event coreference is trained on ECB+ corpus[14] (Cybulska and Vossen, 2014). For wl-coref and caw-coref, we use the Roberta (Liu et al., 2019) based pre-trained models publicly available at their respective Github repositories. In our experiments, using wl-coref led to higher AnCast++ scores.

[12] https://github.com/Jryao/modal_dependency/tree/main/data

[13] https://github.com/Jryao/temporal_dependency_graphs_crowdsourcing/tree/master/tdg_data

[14] https://www.newsreader-project.eu/results/data/the-ecb-corpus/