# AnCast++: Document-Level Evaluation of Graph-based Meaning Representations

**Haibo Sun and Jayeol Chun and Nianwen Xue**
hsun,jchun,xuen@brandeis.edu
Computer Science Department, Brandeis University

## Abstract

Uniform Meaning Representation (UMR) is a cross-lingual document-level graph-based representation that is based on Abstract Meaning Representation (AMR) but is extended to include document-level semantic relations such as coreference as well as modal and temporal dependencies. With recent advancements in UMR annotation efforts, a reliable evaluation metric is essential for assessing annotation consistency and tracking progress in automatic parsing. In this paper, we present *AnCast++*, an aggregated metric that unifies the evaluation of four distinct sub-structures of UMR: (1) sentence-level graphs that represent word senses, named entities, semantic relations between events and their participants, aspectual attributes of events as well as person and number attributes of entities, (2) modal dependencies that represent the level of certainty that a source holds with respect to an event, (3) temporal dependencies between events and their reference times, and (4) coreference relations between entities and between events. In particular, we describe a unified method $TC^2$ for evaluating temporal and coreference relations that captures their shared transitive properties, and present experimental results on English and Chinese UMR parsing based on UMR v1.0 corpus to demonstrate the reliability of our metric. The tool has been made publicly available on Github [1].

## 1 Introduction

Uniform Meaning Representation (Van Gysel et al., 2021) is the latest iteration in a series of semantic representation frameworks beginning with PropBank (Palmer et al., 2005), whose focus on the predicate-argument structure has been inherited by Abstract Meaning Representation (AMR) (Banarescu et al., 2013). The current unified schema of UMR features a sentence-level representation similar to AMR, albeit with several modifications to promote cross-lingual applicability (Flanigan et al., 2022; Bonn et al., 2023b). In addition, UMR introduces a document-level structure that extends beyond sentence boundaries to capture inter-sentence semantic connections such as coreference as well as modal and temporal dependencies. These include entity and event coreference, temporal relations among events and between events and time expressions, as well as modal dependencies between events and their sources, referred to as conceivers (Van Gysel et al., 2021; Vigus et al., 2019; Yao et al., 2021).

Figure 1 shows a sample UMR annotation for a short document of 2 sentences as an illustration:

1. Activists alleged that the prisoners were likely being mistreated yesterday.
2. Their allegation went unnoticed by the public.

In the gold UMR graph on the left of Figure 1, the 3 light blue nodes at the top represent ubiquitous abstract concepts, which includes the ROOT node that ensures the graph is single-rooted. AUTH (author of the text) and DCT (document creation time) serve as sub-roots of modal and temporal dependencies respectively.

A modal dependency graph (MDG) captures the epistemic certainty and polarity with which the conceivers view events or cite other conceivers (Yao et al., 2021; Van Gysel et al., 2021). In the example, the author is certain that the allegation by the activists has already been made (':full-affirmative' edge from AUTH to 's1a:allege-01'). Then activists are partially certain that prisoners are mistreated (':partial-affirmative' edge from 's1a2:activist' to 's1m:mistreat-01'). MDG can be traced by following the red edges in the figure.

A temporal dependency graph (TDG) annotates the temporal relations between events and time expressions such as DCT (Zhang and Xue, 2018). Since all of the events in this example apparently took place in the past, the blue edges originating

[1] https://github.com/umr4nlp/ancast



*Findings of the Association for Computational Linguistics: ACL 2025*, pages 19642–19654

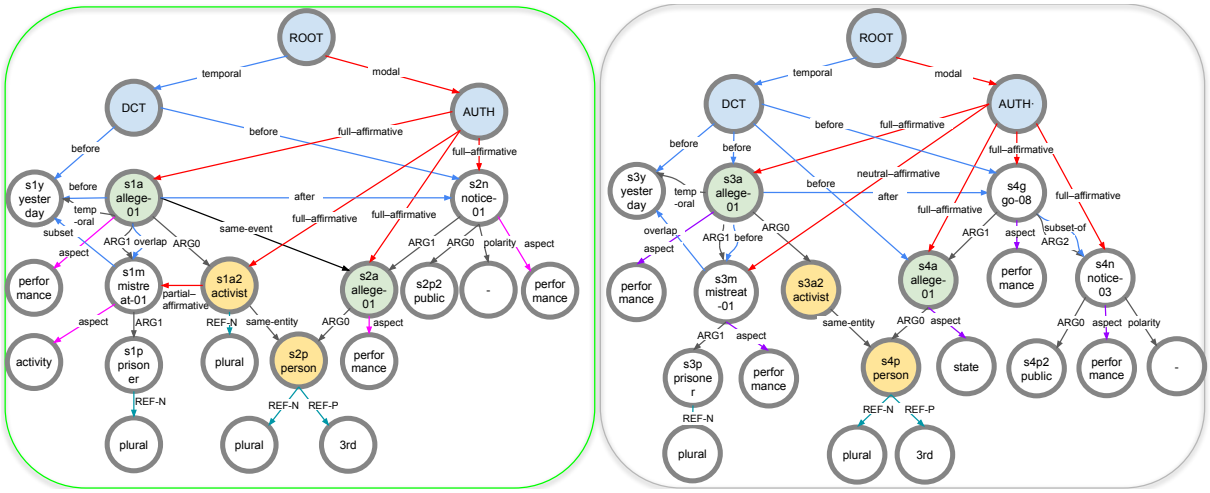



Figure 1: On the left is the gold UMR graph for *"Activists alleged that the prisoners were likely being mistreated yesterday. Their allegation went unnoticed by the public."* Light blue nodes indicate special nodes `ROOT`, `AUTHOR` and `DCT` (Document Creation Time). Modal relations are shown in red edges, temporal relations in blue edges, and the clusters of co-referent events and entities are highlighted in the same color such as green and yellow respectively. `REF-N` stands for `refer-number`, and `REF-P` for `refer-person`. On the right is the test UMR graph with artificial errors for the same sentence for illustration purposes.

from DCT are labeled ':before' which indicates that they happened *before* this document was created by the author.

Coreference chain can be found by identifying nodes linked by edge labels such as ':same-event' and ':same-entity.' In the sample document, 'alleged' ('s1a:allege-01') and 'allegation' ('s2a:allege-01') from sentence 1 and 2 both refer to the same event. Furthermore, the pronoun 'their' ('s2p:person') in the second sentence refers to 'the activists' ('s1a2:activist') in the first sentence.

Finally, the two sentence sub-graphs rooted at 's1a:allege-01' and 's2g:go-08' respectively show some deviations from their corresponding UMR annotations. The purple '`:aspect`' edges in the figure represent aspect which annotates the internal state of an event regarding whether it is an on-going, finished or habitual event, or simply a state with no changes over the course of action, or possibly something else (Donatelli et al., 2018, 2019). In the example, the mistreatment is described as on-going (thus 'activity'), whereas the allegation and going un-noticed events have both been completed (hence the 'performance' value). It is also worth pointing out that the explicit support for plurality of nominals as well as person for pronouns visualized with cyan-colored edges.

In the era of Large Language Models (LLMs), the role of symbolic meaning representations such as UMR and its precursor, AMR, has changed dramatically. Prior to LLMs, datasets annotated with symbolic meaning representations were routinely used to train supervised models as components in pipeline NLP systems such as information extraction (e.g., Pan et al., 2015; Garg et al., 2016; Rao et al., 2017), summarization (e.g., Liu et al., 2015; Liao et al., 2018), machine translation (e.g., Song et al., 2019; Nguyen et al., 2021), question answering (e.g., Sachan and Xing, 2016; Mitra and Baral, 2016; Kapanipathi et al., 2021), and dialog systems (e.g., Bonial et al., 2020; Bai et al., 2021). Increasingly, symbolic representations are now used to complement LLMs by providing structure, consistency, and interpretability (Liang et al., 2024; Edge et al., 2024). While LLMs excel at generating fluent and contextually appropriate responses, they often lack the explicit, systematic framework needed for tasks that require precise semantic representation, cross-linguistic consistency, or detailed reasoning. UMR fills these gaps by offering a clear, interpretable framework to represent meaning, which can enhance applications such as multilingual NLP, fact verification, and explainable AI. Thus, UMR and other symbolic systems work alongside LLMs to ensure that language understanding is not only fluent and adaptable, but also accurate, transparent, and reliable in scenarios that demand high levels of interpretability and rigor.

Thanks to the recent release of UMR v1.0 (Bonn et al., 2023a, 2024), it is now possible to explore automatic UMR parsing (Chun and Xue, 2024). How-

ever, while there are evaluation methods for each of the sub-structures of UMR, a unified framework for assessing the quality of the overall UMR structure is still lacking. This paper introduces AnCast++, a novel tool that implements a metric to measure the accuracy of UMR graphs. AnCast++ starts by mapping vertices between two graphs using the node alignment approach described in (Sun and Xue, 2024). Based on this alignment, the scores of sentence-level graph as well as the modal, temporal dependencies, and coreference relations are computed before finally producing a micro-average score that aggregates the scores of the four components making up the UMR graph. Since temporal dependencies and coreference relations both give rise to graph structures that can be further completed via transitive closure, we develop a unified algorithm, $TC^2$, to leverage this structural commonality and compute scores for both components.

The remainder of this article is organized as follows. In Section 2, we briefly describe the node alignment method used in AnCast. This is followed by a discussion of the evaluation metric for modal dependencies. We then present $TC^2$, a method for computing transitive closures of temporal and coreference relations of UMR and calculating the similarity between temporal and coreference subgraphs using the LEA-inspired method (Moosavi and Strube, 2016). After that, we describe the method for allocating weights to the four components. Section 3 provides experimental results on the use of this metric to measure the output of a UMR parser. Section 4 discusses related work, and Section 5 concludes the paper.

## 2 Methodology

Developing an evaluation metric for comparing UMR graphs presents two core challenges: (i) comparing transitive and expandable relation sets, specifically temporal and coreference relations within the UMR graph, and (ii) assigning appropriate weights to different components of the output to generate a comprehensive score for the graph of the entire document. The first step in comparing two UMR graphs with nodes that may potentially have different labels is to establish a node mapping using an alignment algorithm. To tackle this problem, AnCast++ adopts the anchor and broadcast algorithm proposed in AnCast (Sun and Xue, 2024) to identify node alignment and produces a score by comparing sentence-level semantic graphsin UMR. In this section, we briefly summarize the anchor-and-broadcast algorithm used to obtain node mappings (Section 2.1). We then present the sub-metrics for each component of the UMR graph: the metric for evaluating sentence-level graphs (Section 2.2), the metric for evaluating modal dependencies (Section 2.3), and the metric for temporal dependencies and coreference relations (Section 2.4). Finally, we present how the sub-metrics are aggregated into the full UMR metric in AnCast++ (Section 2.5).

### 2.1 Obtaining Node Mapping

The core strategy of AnCast is to identify the best possible match for each node individually, rather than to seek a global optimum as Smatch (Cai and Knight, 2013). This approach makes the matching process more interpretable and transparent, and avoids assigns similarity scores to graphs that may have similar structure but with nodes that carry entirely different meanings. In addition, the anchor and broadcast algorithm used is an $O(n^3)$ algorithm and it is considerably more efficient than the hill-climbing approaches adopted in Smatch.

The central idea of the anchor-and-broadcast algorithm is to leverage the local and neighboring information of each node to improve alignment accuracy. The local information of a node encompasses its intrinsic properties, such as its lemma—the base form of a word—-and its specific word sense, which defines its meaning in context. By calculating the intrinsic similarities between each pair of nodes from the reference graph and the response graph, an intrinsic similarity matrix $S$ is generated. Each cell of this matrix represents the similarity score of a pair of nodes based on their intrinsic properties and associated attributes (e.g., aspectual values of an event, person and number attributes of an entity). Based on this intrinsic similarity matrix $S$, node pairs with a high similarity score are set as the initial *anchor* for the alignment algorithm.

However, it is unlikely that a node from the reference graph will always have the same intrinsic properties as some node in the response graph. To make the alignment algorithm work, the intrinsic properties of the nodes need to be supplemented with neighboring information, based on the observation that a pair of aligned nodes tend to have same aligned neighbors. The similarity of a node pair can thus be recalculated based on anchors in their neighborhood, and in this sense the similarity of the anchors is *broadcast* to its neighbors. Node pairs with high similarities will again be designated as

| Gold | | Test | |
|---|---|---|---|
| Variable | Label | Variable | Label |
| s1a | allege-01 | s3a | allege-01 |
| s1a2 | activist | s3a2 | activist |
| s1m | mistreat-01 | s3m | mistreat-01 |
| s1p | prisoner | s3p | prisoner |
| s1y | yesterday | s3y | yesterday |
| s2n | notice-01 | s4n | notice-03 |
| s2p2 | public | s4p2 | public |
| s2p | person | s4p | person |
| s2a | allege-01 | s4a | allege-01 |
| NULL | | s4g | go-08 |

Table 1: Node mapping between the reference and test UMR graphs in Figure 1, where each row represents an alignment. Because AnCast prioritizes the intrinsic information of the node over its position in the graph, it correctly aligns *s2n* to *s4n*, which are in different positions relative to the root.

anchors for the next round of recalculation and the process repeats itself till it converges. The reader is referred to (Sun and Xue, 2024) for details on how the anchor matrices are computed.

Given the gold and test graphs in Figure 1, the output of the anchor and broadcast algorithm of AnCast is shown in Table 1. It should be noted that a pre-defined null node serves as the placeholder for the unaligned nodes, such as the one aligned with s4g:go-08, as it does not a corresponding node in the gold graph.

### 2.2 Sentence Level Evaluation

The AnCast score for a pair of sentence-level graphs is based on *labeled relation* scores (Sun and Xue, 2024), which calculates the weighted average of the similarity score of pairs of triples from the reference graph and the response graph, where a triple consists of a parent node and a child node as well as the relation between them. This score is calculated based on pairs of nodes using the gold graph as the reference graph and test graph as the response graph to calculate the precision or recall.

For a pair of nodes $(v_1, v_2)$ in one graph, considering its aligned counterpart $(w_1, w_2)$ in another graph based on the node mappings, the pairwise LR score for $(v_1, v_2)$ $s_p$ is calculated as

$$s_p = s_c \cdot s_{ol} \tag{1}$$

where $s_c$ is the concept similarity, the average of the concept overlap score between $S_{v_1w_1}$ and $S_{v_2w_2}$ within the two pairs and $s_{ol}$ is the number of overlaps between the set of relation labels $L_p$ between $(v_1, v_2)$ and the corresponding set $L'_p$ between $(w_1, w_2)$. The concept similarity $s_c$ is calculated as:

$$s_c = \frac{S_{v_1w_1} + S_{v_2w_2}}{2} \tag{2}$$

The number of overlapping labels is computed (assuming it is possible to have multiple labels between a pair of nodes) as

$$s_{ol} = \sum_{r \in L_p} 1(r \in L'_p) \tag{3}$$

The overall labeled relation precision / recall score is the weighted average of all pairwise scores. Since for an overwhelming majority of cases, the number of labels between a parent node and a child node is 1, it is essentially a plain average of all pairwise scores.

$$\Psi = \frac{1}{\sum_p |L_p|} \sum_p s_p \tag{4}$$

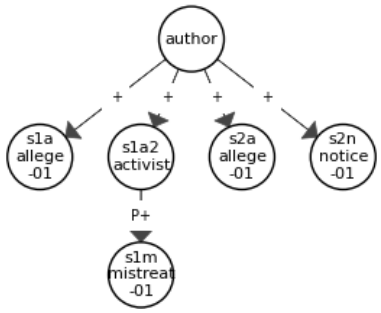
Using the gold graph as the reference graph and the test graph as the response graph, we will get the overall recall $r(s)$. Using the test graph as the reference and the gold as the response, we will get the overall precision $p(s)$.

Based on the gold and test UMR graphs in Figure 1 and the node mappings in Table 1, we identify 12 matched edges for the sentence-level graphs. Dividing the number of matched edges by the total number of sentence-level edges in either the test graph or the gold graph, we can get a precision $p(s)$ of 0.71 and a recall $r(s)$ of 0.75.

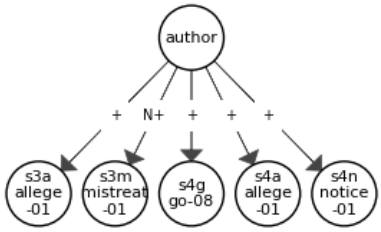
### 2.3 Modal Dependency Evaluation

Unlike temporal or coreference dependencies which are graphs (Yao et al., 2021), a Modal Dependency Graph generally takes the form of a tree structure with directed edges from a parent to a child. Calculating the similarity score for the modal dependency graphs is straightforward and amounts to counting the number of overlapping modal dependency triples between the gold graph and the test graph.

Given the two modal dependency graphs in Figure 2 (which are modal dependency subgraphs from the UMR gold and test graphs in Figure 1) and the node mappings in Table 1, we observe 3 matching triples: (author `:full-affirmative` allege-01 `[s1a|s3a]`), (author `:full-affirmative` allege-01 `[s2a|s4a]`) and (author `:full-affirmative`




(a) Reference Modal Dependency Graph.




(b) Incorrect Modal Dependency Graph.

Figure 2: Gold and test modal dependency sub-graphs from UMR graphs in Figure 1. '+,' 'P+' and 'N+' stand for `full-affirmative`, `partial-affirmative` and `neutral-affirmative` respectively.

notice-01 [`s2n|s4n`]). Dividing this by the total number of edges in the reference graph (5) and the response graph (5), we get a recall $r(m)$ of 0.6 and a precision $p(m)$ of 0.6.

### 2.4 $TC^2$: A unified metric for evaluating temporal and coreferential relations

In this subsection, we introduce $TC^2$ (Transitive Closure of Temporal and Coreferential relations), a novel unified metric we design to evaluate temporal and coreference relations as both exhibit transitive properties.

Following (Setzer et al., 2005), the first work to apply transitive closure to evaluate temporal relations, we use closed graphs rather than unclosed graphs to evaluate temporal and coreference relations. This helps reduce the impact of redundant annotations by ensuring that two graphs containing the same information are evaluated the same way even if the annotated relations in them before closure are not.

We adopt a link-based approach to evaluate the temporal and coreference subgraphs of UMR, extending the Link-based Entity-Aware (LEA) (Moosavi and Strube, 2016) metric—originally designed for coreference evaluation—to assess temporal relations as well.

The following sections detail the conversion of temporal and coreference relations in UMR annotations into computable clusters incorporating transitive closure. This process involves three steps: (i) transforming the temporal and coreference subgraphs in original UMR annotations into a unified graph structure; (ii) computing the transitive closure over the unified graph to capture all inferred relations; (iii) Applying a link-based metric to evaluate the resulting graph, and quantifying the accuracy of the represented relations.

#### 2.4.1 Unified Graph Representation for Temporal and Coreference Relations

**Temporal Dependencies** UMR includes five types of temporal relations, as shown in the top half of Table 2. "before", "after", and "contained" demonstrate transitive properties while "depends-on" (which means that the temporal interpretation of a time expression depends on that of another) and "overlap" (which means that two events or time expressions overlap each other in their duration) do not.

When converting the UMR temporal dependencies into unified graph representations, we observe that temporal composition can only be performed in the same direction for the "before" and "after" relations, and for "contained" relations. That is, we can infer a new relation with two "before" relations (If 'a' is before 'b', 'b' is before 'c', then 'a' is before 'c') or two "after relations" (If 'a' is after 'b', and 'b' is after 'c', then 'a' is after 'c'), but not with one "before" relation and one "after" relation. Similarly, for 'contained' relations, we observe that if 'a' contains 'b' and 'b' contains 'c', then we can infer 'a' contains 'c'. In the opposite direction, we observe that if 'a' is contained by 'b' and 'b' is contained by 'c', then we infer 'a' is contained by 'c'. However, the "before" (or "after") relation interacts with "contains" and "contained-by" relations in different ways: if 'a' is before 'b' and 'b' contains 'c', we can infer that 'a' is before 'c', but if 'b' is contained by 'c', then we cannot make such an inference. The full set of temporal composition rules are presented in Table 3.

Given that, in the unified graph representation, we merge "before" and "after" into a single relation $a \rightarrow b, r$, where $r$ is a shorthand for that rule. However, we split 'contained' into two relations, $up$ for upward containment and $dn$ for downward containment. The full set of temporal relations can be found in the top half of Table 2.

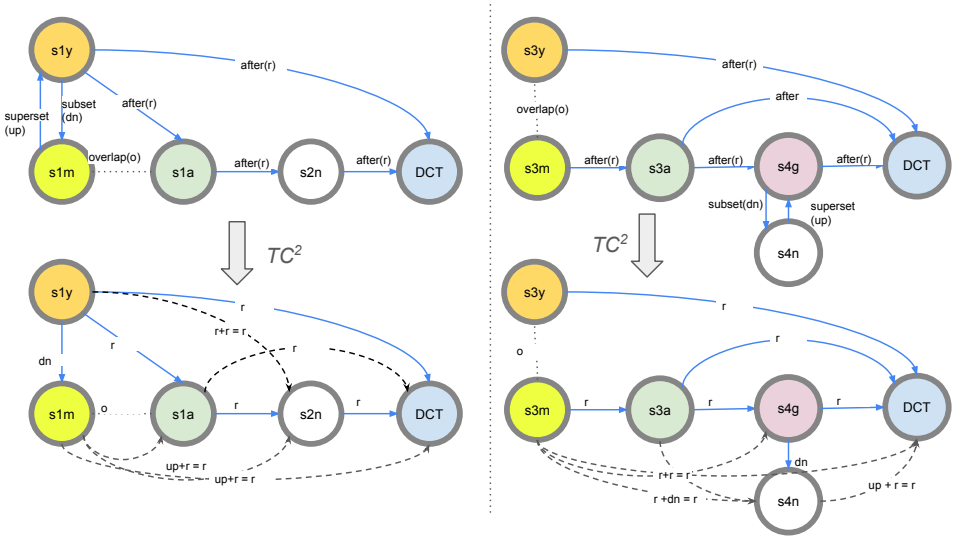



Figure 3: Extracted graphs from both annotations on the top half, and the graphs after transitive closure. Note that after the computation of transitive closure, two links exist between *s1m* and *s1a* which indicates there is an error in the original graph.

| Relationship | Representation |
|---|---|
| a :depends-on b | (a → b, d) |
| a :after b | (b → a, r) |
| a :before b | (a → b, r) |
| a :overlaps b | (a ↔ b, o) |
| a :contained b | (a → b, dn) / (b → a, up) |
| a :same-entity b | (a → b, sn) / (b → a, sn) |
| a :same-event b | (a → b, sv) / (b → a, sv) |
| a :subset-of b | (a → b, up) / (b → a, dn) |

Table 2: Conversion table for temporal and coreference relations, where $a \leftrightarrow b$ represents a symmetric relation.

| ○ | ref | o | d | r | up | dn | sn | sv |
|---|---|---|---|---|---|---|---|---|
| **ref** | - | o | d | r | up | dn | sn | sv |
| **o** | - | - | - | - | - | - | - | - |
| **d** | - | - | - | - | - | - | - | - |
| **r** | - | - | - | r | - | r | - | - |
| **up** | - | - | - | r | up | - | up | up |
| **dn** | - | - | - | - | - | dn | dn | dn |
| **sn** | - | - | - | - | up | dn | sn | sn* |
| **sv** | - | - | - | - | up | dn | sv* | sv |

Table 3: Composition Table for Relations

**Coreference Relations** UMR distinguishes between entities and events when representing coreference relations. Coreferent entities are linked by a "same-entity" relation, while coreferent events are connected by a "same-event" relation. By definition, these two types of coreference never belong to the same cluster, as they represent different conceptual categories. In addition, UMR entities and events can also have a subset ('subset-of') relation. Because all coreference relations are transitive and commutable, both directions are added to the conversion table (Bottom half of Table 2). In particular, the 'subset-of' relation has the same transitive properties as the temporal containment relation and is translated into the same two rules as temporal containment: 'up' and 'dn', as shown in Table 3.

### 2.4.2 Transitive Closure through Graph Traversal

Transitive closure is performed by conducting a depth-first[2] traversal of each node to see which other node this current node can connect to. The traversal is performed in an iterative process until no viable composition can be performed based on the rules in Table 3 and there is no more unvisited node.

Figure 3 illustrates the computation process of the converted temporal graph extracted from Figure 1. In the left (gold graph), `s1m` does not have a transitive connection to `s1a`, so we cannot infer its connection to `DCT` and `s2n` directly. However, since it is contained by `yesterday` which is before `s1a`, `s2n` and `DCT`, we can infer that `s1a` is before both `s2n` and `DCT`.

Similarly, in the test subgraph, we can infer the temporal relations between `s4n` and `s3m`, `s3a` and DCT through its containment relation with `s4g`. These temporal relations do not have a matching relation in the gold graph as `s4g` does not have a matching node in the gold graph. Nevertheless it

[2]The particular order of the traversal is of no significance, and depth-first traversal is chosen for its efficiency.

illustrates the pivotal role of s4g as a "bridge" in the temporal graph.

The coreference annotations extracted from Figure 1 are sparse as one might expect for such a short document. For both the gold graph and the test graph, they are already transitively closed since no additional co-reference relations can be inferred.

To better illustrate the commonality and difference in transitive closure for temporal and coreference in the unified graph representation, we design a more elaborate example in Figure 5, which can be found in the appendix.

During traversal, nodes encountered on the same search path are also recorded as in the same cluster. We then adopted the union find algorithm to merge these clusters after the search process is concluded.

At the conclusion of the transitive closure, we will get a list of node clusters $K = \{k_i\}$ (in the reference graph) and $R = \{r_j\}$ (in the response graph), each with an augmented set of relations that include the original annotated links and inferred links $\{rel(k_i)\}$ and $\{rel(r_j)\}$ from the transitive closure. Note that all links in $\{rel(k_i)\}$ and $\{rel(r_j)\}$ are deduplicated: all relations with "up", "dn", "sn", "sv" are recorded only once in the final set of links.

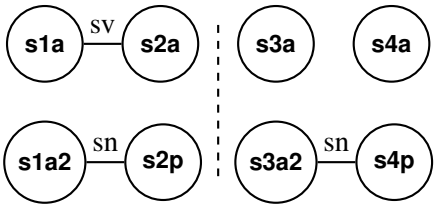



Figure 4: The extracted coreference annotations. The gold graph has two clusters, while the test graph has 3 clusters.

#### 2.4.3 Scoring the closed graph

The recall and precision are subsequently computed as in Equations 5 and 6. The formula are actually very simple: for each cluster in the gold graph, count the number of the links in its augmented set that overlap with those in the test graph, divide it by the total number of links in this cluster, and calculate the weighted average of the resulting ratios where the weights are the number of nodes in each cluster. Note that multiple relations between the same pair of nodes are not prohibited.

$$p = \frac{\sum_{r_i \in R}(|r_i| \times \sum_{k_j \in K} \frac{rel(r_i \cap k_j)}{rel(r_i)})}{\sum_{r_z \in R} |r_z|} \quad (5)$$

$$r = \frac{\sum_{k_i \in K}(|k_i| \times \sum_{r_j \in R} \frac{rel(k_i \cap r_j)}{rel(k_i)})}{\sum_{k_z \in K} |k_z|} \quad (6)$$

According to the equations above, we can compute precisions and recalls for Figure 3 and 4 as follows:

$$p(t) = \frac{6 * 7/12}{6} = 0.58 \quad (7)$$

$$r(t) = \frac{5 * 7/11}{5} = 0.64 \quad (8)$$

$$p(c) = \frac{1 * 0 + 1 * 0 + 2 * 1/1}{4} = 0.5 \quad (9)$$

$$r(c) = \frac{2 * 0 + 2 * 1/1}{4} = 0.5 \quad (10)$$

### 2.5 Weight Allocation

The sentence-level graphs are tree-like, closely resembling tree structures but allow for a few reentrancies. Along with the modality graphs, both are similarly sparse and intuitively weighted by the number of annotated relations. In contrast, the temporal and coreference graphs become significantly denser after transitive closure. To bring these components onto a comparable scale, we assign the weights of the sparse and dense graphs accordingly.

We use the number of nodes in the temporal and coreference annotations, denoted $V_{g/t}(t)$ and $V_{g/t}(c)$ respectively, as normalization factors. The sets of relations in the sentence-level and modality graphs are denoted by $R_{g/t}(s)$ and $R_{g/t}(m)$, where the subscript $g/t$ indicates whether the values are from the gold annotation or the test output.

The final comprehensive precision $p(A)$ and recall $r(A)$ are weighted average of precisions and recalls from all four components of a UMR graph, and a comprehensive F1 score is the harmonic average of $p(A)$ and $r(A)$.

$$W_{g/t} = \{|R_{g/t}(s)|, |R_{g/t}(m)|, |V_{g/t}(t)|, |V_{g/t}(c)|\} \quad (11)$$

$$\begin{aligned} p(A) &= \sum_{w_i \in W_t} w_i p(i) \\ &= \frac{.71 \times 17 + .6 \times 5 + .58 \times 6 + .5 \times 4}{17 + 5 + 6 + 4} \\ &= 0.64 \end{aligned} \quad (12)$$

$$
\begin{aligned}
r(A) &= \sum_{w_i \in W_g} w_i r(i) \\
&= \frac{.75 \times 16 + .6 \times 5 + .64 \times 5 + .5 \times 4}{16 + 5 + 5 + 4} \\
&= 0.67
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
F1 &= \frac{2p(A)r(A)}{p(A) + r(A)} = \frac{2 \times 0.64 \times 0.67}{0.64 + 0.67} \\
&= 0.65
\end{aligned} \tag{14}
$$

### 2.6 Properties of the AnCast++ metric

We analyze the properties of the proposed AnCast++ metrics based on the desiderata outlined in (Opitz et al., 2020) for evaluating graph-based meaning representations. (Opitz et al., 2020) proposed seven principles, some mathematically driven, and some based on linguistic or engineering principles. AnCast++ fulfills all of them except the last one, which is fulfilled partially.

- Continuity, Non-negativity, and Upper Bound: This requires that the metric provides a score in the [0,1] range and the Ancast++ metric does.
- Identity of Indiscernibles: This principle requires that a score of 1 if the two graphs match, and a score less than 1 if they do not. This is fulfilled by AnCast++.
- Symmetry: AnCast++ simply swaps precision and recall when changing the order of graph A and graph B, and the F1 will thus stay the same.
- Determinacy: Ancast++ utilizes a deterministic algorithm for node alignment based on AnCast, and this means that repeated calculation over the same inputs should yield the same score.
- No bias/Transparency: Ancast++ allows configurable weighting for different types of triples, and scores can be traced down to individual triples. Any biases towards a particular component of the graph will be explicitly and transparently indicated in the evaluation process.
- Symbolic semantic match: As an overlap-counting metric, Ancast++ is naturally compatible with the graph-based Jaccard index, which means that Graph A and Graph B are considered more similar to each other than A and Graph C iff A and B exhibit a greater relative agreement in their (symbolic) conditions
- Graded semantic match: Ancast++ partially satisfies this criterion by calculating surface string similarity between concepts. While using dense representations for concepts may offer better granularity, they introduce computational overhead and are unavailable for many sense-tagged predicates or abstract concepts, making full compliance infeasible.

## 3 Parsing Experiments

Although UMR v1.0 corpus contains annotations in 6 languages[3], current UMR parsing results are limited to English only (Chun and Xue, 2024). This is due to the modular and pipelined setup of the parser which consists of smaller sub-models trained independently on external annotations that do not exist in low-resource languages. In this work, we adapt this framework to present the first experimental results on Chinese UMR parsing, despite the absence of temporal dependency dataset for the language. In addition, although (Chun and Xue, 2024) reports a comprehensive macro F1 score of 60.0, this does not account for the disparity in the number of sentences across documents. Consequently, strong performance on a short document makes a disproportionately high contribution to the overall F1 score. We therefore report the aggregate macro F1 score weighted by the number of sentences per document, with the comprehensive score now at 51.9 for English. Table 4 shows the parsing results on the English UMR annotations and Table 5 show the parsing results on the Chinese UMR annotations. These results are primarily meant to demonstrate the reliability of our UMR evaluation metric, although they can also serve as baselines for UMR parsing research. Experimental details can be found in Appendix B.

## 4 Related Work

Research on document-level semantic graph representations remains limited, as do metrics for evaluating their quality. The metrics for evaluating document-level meaning representation graphs include Multi-sentence AMRs (MS-AMR) (O'Gorman et al., 2018) and DocAMR (Naseem

[3] Arapaho, Chinese, Cocama-Cocamilla, English, Navajo, and Sanapaná.

| Doc. ID | English Ancast++ F1 | | | | |
|---|---|---|---|---|---|
| | Sent. | Modal | Temp. | Coref. | Aggr. |
| 0001 | 66.2 | 40.2 | 16.2 | 8.2 | 55.5 |
| 0002 | 90.0 | 60.0 | 100.0 | 0.0* | 86.2 |
| 0003 | 71.8 | 53.9 | 18.2 | 40.0 | 63.4 |
| 0004 | 60.7 | 65.3 | 22.8 | 26.7 | 51.9 |
| 0005 | 55.0 | 12.3 | 7.3 | 20.4 | 42.9 |
| **Macro F1** | 61.3 | 54.0 | 20.3 | 23.6 | **51.9** |

Table 4: UMR Parsing results on English UMR v1.0 dataset. *english_umr-0002 contains no coreference.

| Doc. ID | Chinese Ancast++ F1 | | | | |
|---|---|---|---|---|---|
| | Sent. | Modal | Temp. | Coref. | Aggr. |
| 0001 | 36.6 | 37.3 | 0.0 | 2.7 | 33.9 |
| 0002 | 45.9 | 45.9 | 0.0 | 17.3 | 41.9 |
| 0003 | 37.8 | 52.9 | 0.0 | 9.7 | 36.1 |
| 0004 | 43.1 | 45.2 | 0.0 | 16.7 | 39.6 |
| 0005 | 51.4 | 55.8 | 0.0 | 16.8 | 48.0 |
| 0006 | 37.9 | 31.1 | 0.0 | 14.7 | 34.1 |
| 0007 | 44.7 | 50.0 | 0.0 | 20.0 | 41.5 |
| **Macro F1** | 43.0 | 45.8 | 0.0 | 14.8 | **39.7** |

Table 5: UMR Parsing results on Chinese UMR v1.0 dataset. Temporal dependency score remains zero due to the lack of temporal dependency annotations for Chinese.

et al., 2022) and both extend Smatch (Cai and Knight, 2013) to the document-level graph.

(O'Gorman et al., 2018) introduces a multi-sentence AMR corpus linking sentence-level AMRs into document-level graphs through coreference relations between entities and events. It proposes measuring agreement and parser accuracy by concatenating sentences under a new root and merging coreferent nodes, creating a single connected graph evaluated by Smatch. However, this approach can alter semantics when coreferent nodes are events or contain conflicting information. DocAMR (Naseem et al., 2022) addresses this by introducing a new *coref-entity* node for each identity chain, linking participating nodes via a :coref relation, except for named entities, which are merged, and pronouns, which are removed.

Both MS-AMR and DocAMR assume coreferent entities have identical referents and can be merged into clusters. However, UMR includes temporal relations, modal dependencies, and subset coreference relations that cannot be clustered similarly. To address this, AnCast++ takes a fundamentally different, link-based approach for measuring similarity in UMR graphs and handling temporal and coreference relations.

Multiple evaluation metrics for coreference as a standalone task exist, among which $B^3$(Bagga and Baldwin, 1998) and CEAF (Luo, 2005) consider the overlap between nodes (mentions), while MUC (Vilain et al., 1995) and BLANC (Recasens and Hovy, 2011) focus on the links between them. However, most still suffer from being uninterpretable or exploitable in extreme cases, leading to skewed evaluation results in certain scenarios as explained in (Moosavi and Strube, 2016). One of the best mitigation methods has been to get an average of multiple metrics outputs as is done in the commonly adopted CoNLL metric(Pradhan et al., 2014). Not all coreference metrics extend well to temporal relations. Although both coreference and temporal relations are transitive, temporal relations cannot form clusters as coreference relations typically do. Indeed, even some coreference relations resist plausible clustering. Therefore, we adopt the link-based LEA approach (Moosavi and Strube, 2016), as it readily generalizes to temporal relations.

Early temporal evaluation methods used transitive closures (Setzer et al., 2005), but differing opinions about the relevance of certain relations led later work to emphasize core relations or minimal graphs instead (Tannier et al., 2008). (UzZaman and Allen, 2011; UzZaman et al., 2012) applied temporal closure solely to verify explicit annotations rather than comparing two closed graphs. Other approaches simply counted annotated triples (Verhagen et al., 2010). In contrast, AnCast++ evaluates fully closed graphs, providing a uniform evaluation framework for both temporal and coreference relations.

## 5 Conclusion

We present AnCast++, an aggregated evaluation tool that implements intuitive metrics encompassing sentence-level annotation, modal dependencies, temporal and coreference relations in the UMR graph. AnCast++ also includes a novel $TC^2$ algorithm that unifies the evaluation of temporal and coreference relations using their transitive closures. This represents a significant improvement over the previous scattered metrics in terms of the multi-facet annotations on document level semantic content.

## Acknowledgment

This work is supported by grants from the CNS Division of National Science Foundation (Awards no: NSF_2213804) entitled "Building a Broad Infras-

tructure for Uniform Meaning Representations". Any opinions, findings, conclusions or recommendations expressed in this material do not necessarily reflect the views of NSF.

## Limitations

AnCast++ relies on AnCast's anchor-broadcast algorithm to establish node alignment, which requires that some anchor nodes be identified with high confidence as a starting point. Any node-alignment error could potentially cascade into the document-level evaluation of AnCast++.

Since the size of the UMR dataset remains small, the experimental results are not yet stable as it is possible that a short document may not have any document-level annotation such as coreference for evaluation.

AnCast++ is highly customized for UMR; it is unlikely to be compatible with other meaning representations.

## Ethical Statement

We do not anticipate that this work on semantic graph evaluation will present ethical issues.

## References


Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. 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 4430–4445, Online. 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.

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.

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.

Jayeol Chun and Nianwen Xue. 2024. A pipeline approach for parsing documents into uniform meaning representation graphs. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 40–52, Bangkok, Thailand. 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.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A

graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

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.

Sahil Garg, Aram Galstyan, Ulf Hermjakob, and Daniel Marcu. 2016. Extracting biomolecular interactions using semantic parsing of biomedical text. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, et al. 2024. Kag: Boosting llms in professional domains via knowledge augmented generation. *arXiv preprint arXiv:2409.13731*.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract meaning representation for multi-document summarization. *arXiv preprint arXiv:1806.05655*.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 25–32.

Arindam Mitra and Chitta Baral. 2016. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30(1).

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 Fernandez Astudillo, Radu Florian, Salim Roukos, and Nathan Schneider. 2022. Docamr: Multi-sentence amr representation and evaluation. pages 3496–3505.

Long HB Nguyen, Viet H Pham, and Dien Dinh. 2021. Improving neural machine translation with amr semantic graphs. *Mathematical Problems in Engineering*, 2021.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

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.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with Abstract Meaning Representation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1130–1139, Denver, Colorado. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2014, page 30. NIH Public Access.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using Abstract Meaning Representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural language engineering*, 17(4):485–510.

Mrinmaya Sachan and Eric Xing. 2016. Machine comprehension using rich semantic representations. In

*Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 486–492, Berlin, Germany. Association for Computational Linguistics.

Andrea Setzer, Robert J Gaizauskas, and Mark Hepple. 2005. Using semantic inferences for temporal annotation comparison.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

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 (LREC-COLING 2024)*, pages 1052–1062, Torino, Italia. ELRA and ICCL.

Xavier Tannier, Philippe Muller, et al. 2008. Evaluation metrics for automatic temporal annotation of texts. In *LREC*.

Naushad UzZaman and James Allen. 2011. Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 351–356.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3-4):343–360.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62.

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.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Zhixing Xu, Yixuan Zhang, Bin Li, Zhou Junsheng, and Weiguang Qu. 2023. Overview of CCL23-eval task 2: The third Chinese Abstract Meaning Representation parsing evaluation. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pages 70–83, Harbin, China. Chinese Information Processing Society of China.

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.

Yuchen Zhang and Nianwen Xue. 2018. 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).

| English Doc. ID | Sentences | Doc. Level | Tokens |
|---|---|---|---|
| english_umr-0001 | 28 | 28 | 700 |
| english_umr-0002 | 2 | 28 | 18 |
| english_umr-0003 | 9 | 9 | 140 |
| english_umr-0004 | 141 | 135 | 1,165 |
| english_umr-0005 | 29 | 29 | 566 |
| **Total** | 209 | 203 | 2,589 |
| Chinese Doc. ID | Sentences | Doc. Level | Tokens |
| chinese_umr-0001 | 51 | 51 | 1,466 |
| chinese_umr-0002 | 60 | 60 | 1,599 |
| chinese_umr-0003 | 31 | 31 | 1,021 |
| chinese_umr-0004 | 59 | 59 | 1,557 |
| chinese_umr-0005 | 40 | 40 | 1,128 |
| chinese_umr-0006 | 35 | 35 | 967 |
| chinese_umr-0007 | 82 | 82 | 1,505 |
| **Total** | 358 | 358 | 9,243 |

Table 6: UMR v1.0 dataset statistics for English and Chinese. *Doc. Level* refers to the number of non-empty document-level graphs.

## A Corpus Details

Table 6 provides a summary of UMR v1.0 dataset in English used for evaluation. The English corpus is sourced from newsire as well as weblog domain, whereas the Chinese dataset consists of newswire only.

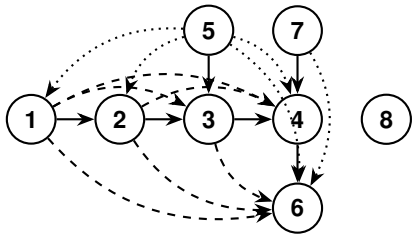
## B Experimental Details

Chun and Xue (2024) advocates for a divide-and-conquer approach for UMR parsing by building

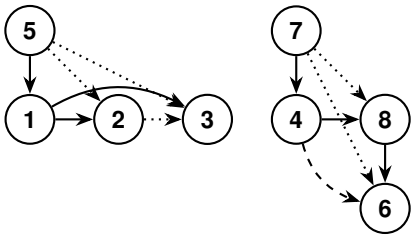
sub-structures of UMR individually, before merging them into the final structure. We adopt this setup and replicate the results on English to compute the weighted version of the Ancast++ metric.

In applying this framework to Chinese, the absence of Chinese temporal dependency parser makes it more challenging than for English. Our efforts of using the LLMs to (1) translate the English annotations to Chinese and then train the temporal dependency parser, and (2) predict the temporal relations directly does not yield fruitful results to be included in the pipeline. However, we observe some efficacy with coreference, where prompting the ChatGPT (`gpt-4o-2024-08-06`) leads to improved performance over traditional libraries. For AMR parsing, we train the SUDA's entry in the CCL23-Eval Task 2 (Xu et al., 2023). We use the modal dependency parser from Yao et al. (2022).

## C Illustrative examples for a complex temporal and coreference evaluation




(a) First diagram




(b) Second diagram

Figure 5: Combined diagrams. Dotted lines are inferred from co-reference only, and dashed lines are inferred from both scenarios.

Figure 5 serves a dual purpose for illustrating how transitive closure is performed on either temporal dependencies or coreference relations and how the similarity scores are computed. Horizontal lines between nodes represent either "after" relations (temporal) or "same-entity/event" relations (coreference). Vertical lines indicate "contained" or "subset-of" relationships. Only dashed lines denote inferred temporal relations, but both dashed and dotted lines indicate inferred coreference relations. Both graphs contain two clusters.

Suppose this is a temporal dependency graph used to compute transitive closure, and we start from 1. It can be inferred that 1 is connected to 2, 3, 4, and 6, but not 5 or 7, because 5 and 7 cover a wider time span than 3 and 4 so it is unclear whether 1 overlaps with 5 or precedes 5. Same for 7, as based on Table 3, $r + up$ is not computable.

However, if Figure 5 is a unifed graph for coreference, then 1 can travel to all nodes in the search process, because 1 is the same entity or event as 3 and 4, and 3 and 4 are a subset of 5 and 7 respectively, so 1 is also a subset of 5 and 7. Formally it is an application of the rule $sn + up = up$.

The temporal scores between Figure 5a and Figure 5b are computed in Equations 15 and 16. Only $1 \rightarrow 2$, $1 \rightarrow 3$ and $7 \rightarrow 4$ in Figure 5a are also found in Figure 5b. So:

$$p = \frac{7 \times \frac{3}{12} + 1 \times 0}{8} = 0.22 \quad (15)$$

$$r = \frac{4 \times \frac{2}{3} + 4 \times \frac{1}{4}}{8} = 0.46 \quad (16)$$

More coreference relations can be inferred than temporal relations. For example, In Figure 5b, $2 \rightarrow 3$ can be inferred from the coreference relations between $1 \rightarrow 3$ and $1 \rightarrow 2$ as `sn+sn=sn` based on Table 3. The coreference scores are computed in Equation 17 and 18.

$$p = \frac{7 \times \frac{6}{17} + 1 \times 0}{8} = 0.31 \quad (17)$$

$$r = \frac{4 \times \frac{6}{6} + 4 \times \frac{2}{6}}{8} = 0.67 \quad (18)$$