Bootstrapping UMRs from Universal Dependencies for Scalable Multilingual Annotation

Federica Gamba[1*] and **Alexis Palmer**[2] and **Daniel Zeman**[1]

[1]Charles University, Faculty of Mathematics and Physics
[2]University of Colorado Boulder
{gamba,zeman}@ufal.mff.cuni.cz alexis.palmer@colorado.edu

Abstract

Uniform Meaning Representation (UMR) is a semantic annotation framework designed to be applicable across typologically diverse languages. However, UMR annotation is a labor-intensive task, requiring significant effort and time especially when no prior annotations are available. In this paper, we present a method for bootstrapping UMR graphs by leveraging Universal Dependencies (UD), one of the most comprehensive multilingual resources, encompassing languages across a wide range of language families. Given UMR's strong typological and cross-linguistic orientation, UD serves as a particularly suitable starting point for the conversion. We describe and evaluate an approach that automatically derives partial UMR graphs from UD trees, providing annotators with an initial representation to build upon. While UD is not a semantic resource, our method extracts useful structural information that aligns with the UMR formalism, thereby facilitating the annotation process. By leveraging UD's broad typological coverage, this approach offers a scalable way to support UMR annotation across different languages.

1 Introduction

Uniform Meaning Representation (UMR) (Van Gysel et al., 2021) is a graph-based meaning representation framework primarily grounded in Abstract Meaning Representation (AMR) (Banarescu et al., 2013). Unlike AMR, which is mainly designed for English, UMR was specifically developed with a cross-linguistic scope, focusing particularly on morphologically complex and low-resource languages. UMR provides a sentence-level representation that captures core elements of meaning such as predicate-argument structure and word senses. Compared to AMR, it also introduces features to better handle tense, aspect, modality, and quantification in a way that generalizes across languages. Beyond the sentence level, UMR supports document-level annotation, which defines strategies to represent coreference among entities and events, temporal relations, and modal relations. All these features make UMR a rich, flexible framework for modeling meaning in cross-lingual contexts. UMR graphs are directed graphs, mostly acyclic, with each concept represented as a node and edges encoding semantic relations. Through the use of re-entrancies, a single node can participate in multiple relations, supporting the expression of shared arguments and anaphoric reference.

As is often the case with deep semantic annotations, annotating data according to the UMR formalism has proven to be extremely time-consuming, highlighting the need for alternative solutions and partial automation of the annotation process. This issue is particularly relevant for languages which lack the same resources and annotators as widely spoken languages like English. In this paper, we present a method for converting Universal Dependencies (UD) (de Marneffe et al., 2021) trees into (partial) UMRs. UD is one of the most comprehensive multilingual resources, covering a wide range of typologically diverse languages – 179 in total as of version 2.16. In light of the typologically motivated nature of UMR, UD's broad typological coverage is particularly valuable for this task. At the same time, while UMR abstracts away from the morpho-syntactic representation of language properties, UD is primarily concerned with representing morpho-syntax. Since UD is not a semantic resource, a full UMR graph cannot be expected from this conversion. However, generating reasonably accurate partial graphs is already highly beneficial, as it provides annotators with a structured starting point, reducing the effort required for manual annotation.

Our contributions include: a) a language-independent UD-to-UMR converter; b) a manually

[*]Work partially done while visiting the University of Colorado Boulder.

annotated test set comprising 100 parallel sentences in three languages (Czech, English, and Italian), for a total of 300 sentences;[1] c) two-fold evaluation of the conversion, aimed at providing insights into the interaction between syntax and semantics.

The remainder of the paper is structured as follows. We first provide background on conversion strategies to UMR (Section 2), followed by the presentation (Section 3) and evaluation (Section 4) of the UD-to-UMR converter. Finally, we conclude with a discussion of future directions (Section 5).

2 Related Work

Like other forms of semantic representation, UMR annotation is a time-consuming and labor-intensive task, highlighting the need for automatization methods that could streamline the process. Converting AMR corpora to UMR (Bonn et al., 2023) is undoubtedly a promising and valid approach. However, due to UMR's inherent emphasis on multilinguality, restricting UMRs to languages with existing AMRs is not sufficient. Instead, it is crucial to develop strategies that leverage other available corpora to generate UMRs.

Buchholz et al. (2024) address this challenge by proposing a method to bootstrap UMRs from interlinear glossed text (IGT), providing annotators with a preliminary structure rather than requiring them to annotate from scratch – an objective that aligns with our UD-to-UMR conversion efforts. While their approach is applied exclusively to Arapaho, its potential for broader applicability is demonstrated with Quechua data. Their method generates subgraphs centered around individual verbs, leaving it to the annotator to integrate them into a cohesive structure for complex constructions, such as subordinate clauses. In contrast, our approach builds a single, comprehensive graph that directly incorporates subordination.

Another line of research involves converting the Prague Dependency Treebank (PDT) to UMR (Lopatková et al., 2024). The tectogrammatical layer in PDT (Hajič et al., 2020) captures deep syntactic-semantic properties of language; while maintaining the dependency structure used at the surface-syntactic level, it encodes semantic features such as argument (valency) structure, predicate senses, and semantic attributes of nodes. PDT trees share structural similarities with UD trees, but the presence of rich semantic annotations facilitates a more comprehensive conversion to UMR, including elements such as coreference. PDT is a Czech resource, so its conversion process remains language-specific. However, a similar PDT-style annotation exists for Latin,[2] and efforts are underway to convert it as well.

A prior attempt to generate meaning representations from dependency syntax was made by Han and Pavlova (2019), who focused on developing a system to convert UD trees into AMRs. This approach utilizes a rewriting system supported by a lexical resource containing predicates from the PropBank dataset. While this work serves as an important precedent, it differs from our approach in at least three key aspects: it converts to AMR rather than UMR, it is language-specific (English only), and it is highly lexicalized, relying on PropBank to disambiguate concepts.

In addition to efforts to generate complete or partial UMRs, there have also been attempts to automatically extract specific elements of the graph, such as verbal aspect (Chen et al., 2021) and word senses (Gamba, 2024).

3 UD-to-UMR Approach

In our work, we focus exclusively on generating the sentence-level UMR graph and alignments for each sentence, whereas a full UMR annotation typically includes a document-level block. Our approach involves iterating over all nodes in each UD tree and processing them sequentially. For each node, we determine its position in the sentence graph being generated and produce alignments by extracting token indices. To handle UMR graphs and UD trees, we use the Penman (Goodman, 2020) and Udapi (Popel et al., 2017) Python libraries, respectively.

Concept nodes are defined as lemmas. Since we do not rely on language-specific frame files, we extract UD lemmas to label concepts. This approach occasionally leads to a literal interpretation of the sentence, which may not always align perfectly with the intended UMR representation. However, in most cases, it provides a sufficient approximation for our purposes.

Participant roles are defined through a set of linguistically informed rules that map UD annotations to UMR structures. These mappings go beyond

[1]The converter and the annotated test set are openly available at `https://github.com/fjambe/UD2UMR`.

[2]The texts annotated in the PDT style are the Index Thomisticus Treebank (ITTB) (Passarotti, 2019) and a portion of the Latin Dependency Treebank (LDT) (Bamman and Crane, 2006).

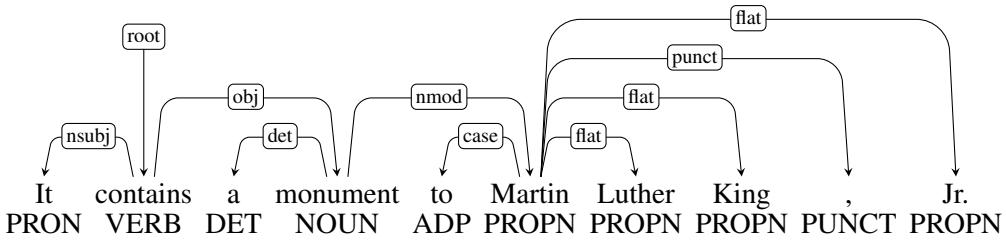


Figure 1: UD tree for the sentence *It contains a monument to Martin Luther King, Jr.* (English PUD, `w02005029`).

a simple one-to-one correspondence between UD syntactic relations and UMR semantic roles; they combine syntactic labels with morphological features (e.g., `Case`, `Polarity`) to infer appropriate semantic roles. For instance, `nsubj`, `csubj`, and `obl:agent` are mapped to the semantic role `actor`, while `obj`, `nsubj:pass`, and `csubj:pass` are interpreted as `undergoer`. Morphological cues play a key role in disambiguation: for example, a dependent labeled `obl` with `Case=Dat` is treated as a `recipient`. In some cases, the mapping introduces abstract predicates rather than roles. For instance, appositions (`appos`) are not merely mapped to a role label; instead, they are converted to the abstract predicate `identity-91`, following UMR conventions. Similarly, copular constructions (`cop`) are also converted to a set of of abstract predicate structures. Since UD relations are not as semantically fine-grained as UMR roles require, exact alignment is not always possible. Our goal is to approximate semantic roles in a principled way using available syntactic and morphological cues, rather than striving for exhaustive and exact coverage. The participant roles in our generated UMRs correspond to non-lexicalized semantic roles[3] typically used in what UMR guidelines call 'Stage 0 annotation', where no PropBank-style frame files are available. Incorporating frame files would introduce language-specific dependencies, and our goal is to develop a broadly applicable approach.

Hereafter, we use the English sentence "It contains a monument to Martin Luther King, Jr." as an example and present the corresponding human-annotated graph, the converted UMR graph, and its UD tree (Figure 1).

[3]For example, *actor*, *theme*, *recipient*, rather than frame-specific arguments like ARG0 or ARG1.

Gold UMR graph:

```
(s1c / contain
    :actor (s1t / thing
        :refer-number singular)
    :undergoer (s1m / monument
        :mod (s1p / person
            :name (s1n / name
                :op1 "Martin"
                :op2 "Luther"
                :op3 "King"
                :op4 "Jr."))
        :refer-number singular)
    :modal-strength full-affirmative
    :aspect state)
```

Generated UMR graph:

```
(s1c / contain
    :actor (s1t2 / thing
        :refer-number singular)
    :undergoer (s1m / monument
        :mod (s1t / type-NE
            :name (s1n / name
                :op1 "Martin"
                :op2 "Luther"
                :op3 "King"
                :op4 "Jr."))
        :refer-number singular)
    :modal-strength full-affirmative
    :aspect ASP)
```

In this example, the graphs diverge in the `aspect` attribute and `type-NE` element present in the converted graph. The `aspect` attribute is generated during conversion whenever a predicate is identified, even if no specific value can be assigned. In such cases, it is represented by the placeholder string `ASP`, ready for annotators to fill in. This approach is necessary because UD morphological features do not consistently provide aspect information, and can prove helpful as the objective is not to automatically produce perfect UMRs, but rather to streamline the annotation process. Similarly, for Named Entities, UD does not provide sufficient information to determine the correct type (e.g., `person`, `place`, or other values from the provided UMR hierarchy). Therefore, we assign a default placeholder (`type-NE`) to be refined during annotation. The same approach is applied to handle several relations that cannot be extracted from a syntactic

tree, but where we can at least identify the broader category (e.g., the placeholder `OBLIQUE`, encompassing various UMR relations such as `temporal`, `place`, `goal`, `source`, and others).

3.1 Syntax-Semantics Mismatches

Mapping syntax to semantics becomes particularly challenging when linguistic structure does not directly align with conceptual meaning. Szubert et al. (2018) observed that, while much of the semantics in English AMRs can be mapped to the lexical and syntactic structure of a sentence, substantial structural differences between AMR and dependency syntax often lead to non-isomorphic mappings between syntactic and semantic representations.

One key issue involves eventive concepts, which do not always correspond to verbal predicates. While verbs are prototypical carriers of event meaning, many events are expressed through nominal constructions (so-called event nominals) that lack explicit grammatical markers of aspect (e.g., *his arrival* vs. *he arrived*). Since UD relies on syntactic categories, such nominal events are difficult to identify automatically.[4]

Syntax and semantics also diverge in the case of abstract concepts, defined as concepts that are identified and annotated even though they do not consistently correspond to any overt word in the sentence. Among those, UMR introduces a set of abstract predicates to account for core non-verbal clause functions, such as `identity-91` (equational) and `have-mod-91` (property predication). In copula-using languages, these often align with copular constructions. While some heuristics can help disambiguate such structures, assigning these predicates automatically based on syntax alone remains highly challenging.

Another problematic phenomenon is re-entrancies, where the same participant appears multiple times in a sentence. Since UD trees do not encode repeated participants, extracting this information is not trivial.[5] Moreover, re-entrancies represent a form of coreference, which is typically handled at the discourse level rather than within sentence-level annotation, and is outside our current scope.

Finally, aspectual categories in UMR introduce additional complexity. UMR provides fine-grained aspectual distinctions, but these often rely more on lexical semantics and human interpretation than on overt morphosyntactic markers. For instance, in languages like Czech or Italian, the distinction between states and activities (in UMR annotated as aspect) relies primarily on lexical meaning rather than explicit grammatical cues. As a result, UD-based approaches struggle to capture such differences effectively.

3.2 Lexical Resources

Syntactic information alone is often inadequate for capturing semantic distinctions. In certain cases, lexical information can provide valuable insights, though it tends to be language-specific. To account for this, we adopt a modular approach, designing our converter to allow for the integration of language-specific lexical resources while ensuring that the code operates independently of them.

As of the current implementation, we have created lexical resources to cover interpersonal terms (used to assign the abstract predicate `have-rel-role-92`), conjunctions, verbs associated with specific `modal-strength` values, and subordinate conjunctions that help disambiguate adverbial clauses to assign the appropriate UMR relation. This set of lexical phenomena could be further expanded —- for example, by incorporating adverbials that signal specific `modal-strength` values —- but we leave this for future work. Lexical resources are available for Czech, English, French, Italian, and Latin, and it is straightforward to extend this to additional languages.

3.3 Impact of UD Annotation on Conversion

We have observed that variations in the consistency of the UD annotation have a significant impact on conversion. As in parsing (Gamba and Zeman, 2023a,b), a lack of harmonization in treebanks leads to error propagation, affecting the overall quality of the conversion.

The granularity of UD annotation also influences conversion outcomes. For example, when converting from the Italian Parallel UD Treebank (PUD) (Zeman et al., 2017), unwanted articles appear in the UMR graphs because the feature

[4] One possible approach is leveraging derivational lexicons, but this is only feasible for high-resource languages where such lexicons exist.

[5] Enhanced UD (Nivre et al., 2020) could be leveraged to extract this type of information; however, full annotation across all enhancement types is available for only 19 treebanks to date. Some of the missing enhancements can be extracted heuristically from basic UD trees, though the heuristics are partially language-specific.

PronType=Art is not annotated in the treebank.[6] Without this feature, distinguishing articles from other determiners (tagged as DET)—which do belong in UMR[7]—is not possible.

Similarly, the UD subtypes tmod and lmod, which mark temporal and locative obl and advmod modifiers, are not widely used across treebanks. If consistently available, they could help disambiguate UMR relations such as temporal and place.[8] However, their usefulness is limited, as these labels may also correspond to roles like start[9] or goal.[10] This highlights a structural limitation of UD, where syntactic distinctions are often less fine-grained than those required by UMR.

Additionally, some specific phenomena vary too much across languages to be handled uniformly in conversion. A notable example is date and time expressions, which differ widely in format, preventing a systematic conversion to the standardized UMR date-entity structure. This challenge is reflected by the difficulty of establishing a language-agnostic UD annotation strategy for these expressions, as noted by Zeman (2021). Even when semantically equivalent, their syntactic structures are not always compatible across languages, making it difficult to establish universal annotation rules.

4 Evaluation

Evaluating the performance of our UD-to-UMR conversion system is crucial for understanding its strengths and limitations. To this end, we propose a two-fold evaluation aimed at addressing two key questions: (a) How accurate is the conversion? That is, to what extent are the partial graphs constructed from UD syntactic information correct? and (b) How useful is the conversion for annotators? Specifically, does providing converted graphs as a starting point help streamline annotation?

To answer the first question, we design a quantitative evaluation to assess the converter's performance. However, evaluating converted UMR graphs poses challenges, as these graphs are often incomplete due to the inherent difficulty of capturing certain semantic phenomena solely from syntax. While tools like AnCast (Sun and Xue, 2024) and metrics like Smatch (Cai and Knight, 2013; Opitz, 2023) exist for evaluating graph-based meaning representations, relying solely on the metrics they provide would be insufficient. A more insightful approach involves focusing on specific challenging phenomena rather than just general scores. For example, examining how well the converter handles abstract predicates offers a clearer understanding of its performance with complex structures. Our approach is inspired by Groschwitz et al. (2023), who developed the GrAPES evaluation suite to assess not only the overall performance of AMR parsers but also their ability to handle specific linguistic and structural phenomena. Similarly, we aim to complement overall F_1 scores with targeted evaluations of key challenges in UMR conversion.

Another factor affecting evaluation is graph connectivity. To prevent the generation of disconnected subgraphs, some converted triples[11] are discarded before finalizing the graph. This happens when the parent node cannot be converted, leaving the subgraph unattached to the main structure. Such trade-off ensures structural integrity, while slightly affecting overall conversion scores and adding complexity to interpretation of the evaluation results.

In addition to the quantitative evaluation, we address the second question by conducting a time-based evaluation. Our goal is to measure whether, and to what extent, providing annotators with a graph backbone (the converted UD graph) helps them complete their annotations more efficiently.

4.1 Test Set

Our test set consists of 100 sentences per language,[12] covering Italian, English, and Czech. Each set is composed of 30 sentences annotated manually from scratch, and 70 automatically converted graphs that were then manually corrected. The decision to include more converted sentences than fully manual ones stems from the fact that

[6] As of UD v2.16.

[7] Some determiners, like *some* and *all* in English, are included in UMR graphs because they contribute meaning – for example, by indicating quantity. In contrast, articles are left out, since they typically do not add any semantic content.

[8] Defined in the UMR guidelines as the location at which the action takes place.

[9] Location at which a motion event begins.

[10] Location at which the action ends.

[11] A UMR graph is essentially a collection of triples, where triples can be of three types: 1) instances (g, instance, 'graph'), 2) edges (r, actor, g), and 3) attributes (g, refer-number, plural).

[12] However, for one sentence in Czech and English our approach did not output any graph; therefore only 99 sentences are actually evaluated for these languages. This occurred because the conversion process discards certain triples to prevent disconnected subgraphs. In these cases, the issue stemmed from the top node, i.e. the root of the syntactic tree, being a copular construction, which typically requires mapping to an abstract predicate and is often challenging to convert. Consequently, all triples became disconnected and were discarded, preventing the generation of a graph for these sentences.

annotation from scratch is highly time-consuming and labor-intensive. Additionally, starting from a converted backbone ensures greater comparability across UMRs, as multiple UMR structures can be equally valid.

The Italian and English test sets were each annotated by one annotator, whereas the 100 Czech sentences were evenly split among three annotators, both for manually annotated and converted sets. The sentences are sourced from PUD treebanks (Zeman et al., 2017), containing texts from two genres (Wikipedia and news) and five original languages, from which translations were made.[13] We randomly select our test set from the complete PUD treebank, in order to sample across both genres and original languages.

4.2 Quantitative Evaluation

The evaluation proposed here aims to measure the extent to which UD-converted UMRs align with their manually annotated counterparts, providing a measure of the conversion process's effectiveness. To structure our evaluation, we use AnCast (Sun and Xue, 2024) to process graphs. While its built-in metrics are insufficient for our specific needs (Section 4), its evaluation framework remains valuable and can be partially leveraged.

A key challenge in the evaluation is identifying which nodes to compare between the converted and gold-standard graphs. Typically, this task is handled by the alignment block, which maps UMR nodes to surface tokens. However, since the UMR guidelines do not formally regulate alignment annotation, inconsistencies arise in the data, making the parsing process more complex than expected. Specifically, a major limitation we encounter is that AnCast does not support discontinuous alignment ranges, which are common in UMR annotations. For instance, in a sentence like *He had already arrived*, the alignment for the predicate *arrive* would be discontinuous (aligning to *had* at position 2 and *arrived* at position 4, i.e. `2-2, 4-4`). Due to this limitation, we are unable to use manually provided alignment blocks and instead adopt AnCast's automated anchor extraction method. This method identifies a subset of highly similar node pairs between the two graphs and iteratively refines the anchor matrix through the anchor broadcast process. For a detailed explanation of this approach, see Sun and Xue (2024).

Table 1 presents evaluation results for Czech, English, and Italian across several linguistic categories. It includes both dependency-style evaluations and the phenomenon-specific evaluations described earlier. English generally has the highest performance, while Czech and Italian exhibit greater variability. Performance varies significantly across semantic categories. For example, relatively high scores are achieved for the assignment of `refer-person` and `refer-number` to newly generated entities,[14] or for annotation of operands (`op1`, `op2`, ...). It indicates that these categories are relatively straightforward to map to syntax, despite structural divergences between the annotation frameworks. In contrast, phenomena that tend not to be overtly encoded at the syntactic level, such as modal strength, or phenomena with very specific structures, such as inverted relations, present significant challenges for automatic extraction.

A consistent trend across all languages is the higher precision compared to recall; this is not surprising, particularly considering that, as mentioned in Section 4, some correct triples are discarded to prevent graph disconnection.

A key consideration is that we adopt a strict evaluation approach. Specifically, there are instances where we are unable to extract a UMR relation from the UD tree but can at least assign a placeholder indicating the broader category (e.g., `OBLIQUE`, Section 3). In the proposed evaluation, these cases have been counted as incorrect; however, there are instances where this annotation could be considered (partially) correct, as it corresponds to a group of UMR relations that we have defined as falling under the broader label. Another significant limitation stems from the alignment strategy, as only nodes that are successfully aligned following the anchor broadcast process are evaluated, meaning that a number of triples are excluded from assessment. As a result, the scores may be affected by the fact that not all nodes are compared.

[13] The first 750 sentences in PUD were originally written in English, while the remaining 250 sentences originated in German, French, Italian, or Spanish and were translated into other languages via English.

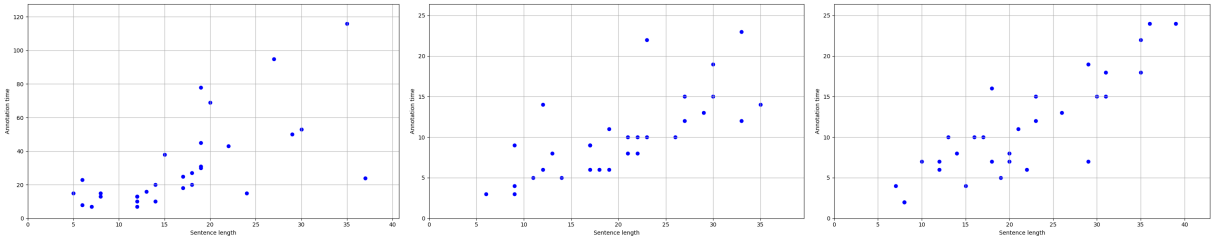
[14] The UMR representation of these attributes differs from their representation in morphosyntax. E.g., the English pronoun *he* is not represented as a lexicalized concept, but it is converted to an abstract concept `person` with `refer-number singular` and `refer-person 3rd`. Moreover, in pro-drop languages the equivalent pronoun (such as *on* 'he' in Czech) may be omitted at the syntactic level, while it is explicitly included in the corresponding UMR graph.

		Czech			English			Italian		
	Subtype	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
					Overall					
parent-label		0.666	0.622	0.643	0.718	0.668	0.692	0.712	0.704	0.708
					Edges					
LAS		0.276	0.234	0.253	0.366	0.331	0.347	0.311	0.317	0.314*
UAS		0.516	0.437	0.473	0.582	0.527	0.553	0.493	0.503	0.498*
child-label		0.374	0.317	0.343	0.449	0.407	0.427	0.401	0.409	0.405
LAS	manual**	0.234	0.257	0.245	0.168	0.219	0.190	0.237	0.260	0.248
					Participants					
LAS		0.222	0.203	0.212	0.362	0.303	0.330	0.304	0.269	0.285
UAS		0.380	0.348	0.364	0.502	0.420	0.457	0.432	0.383	0.406
					Non-participants					
LAS		0.240	0.443	0.311	0.351	0.447	0.393	0.256	0.535	0.346
UAS		0.309	0.571	0.401	0.427	0.543	0.478	0.306	0.641	0.346
					Arguments					
LAS		0.378	0.138	0.202	0.457	0.286	0.351	0.500	0.152	0.233
UAS		0.449	0.164	0.240	0.543	0.340	0.418	0.516	0.156	0.240
					Operands					
LAS		0.658	0.453	0.536	0.613	0.575	0.594	0.714	0.533	0.610
UAS		0.671	0.462	0.547	0.642	0.602	0.621	0.725	0.541	0.620
					Entities					
LAS	`refer-number`	0.862	0.403	0.549	0.952	0.385	0.548	0.875	0.167	0.280
LAS	`refer-person`	0.889	0.706	0.787	0.900	0.281	0.429	1.000	0.241	0.389
					Modal strength					
LAS	polarity	0.704	0.605	0.651	0.813	0.688	0.745	0.870	0.637	0.735
LAS	strength	0.180	0.155	0.166	0.224	0.189	0.205	0.235	0.172	0.199
					Inverted relations					
UAS		0.364	0.112	0.171	0.426	0.294	0.348	0.667	0.184	0.288
child-label		0.250	0.077	0.118	0.277	0.191	0.226	0.417	0.115	0.180
					Abstract predicates					
parent-label	predicate	0.410	0.211	0.278	0.581	0.340	0.429	0.548	0.274	0.366
UAS	dependents	0.487	0.447	0.466	0.565	0.565	0.565	0.500	0.500	0.500
LAS	ARG nodes	0.397	0.437	0.416	0.500	0.620	0.554	0.500	0.633	0.559

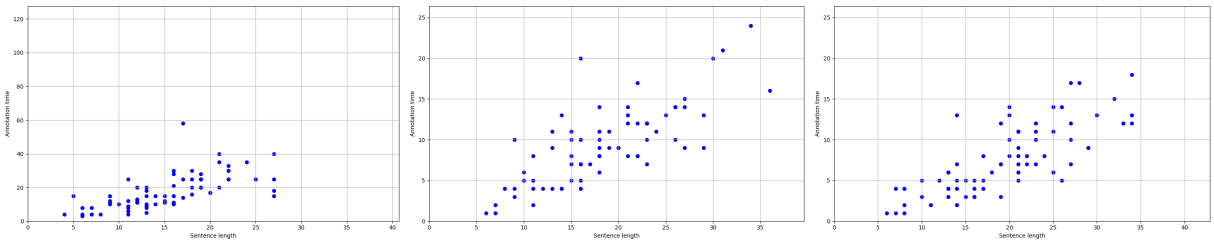
Table 1: **Evaluation results on the test set** for Czech, English, and Italian.
Inspired by dependency syntax (Buchholz and Marsi, 2006), LAS (Labeled Attachment Score) requires all three components of a dependency triple to be correct (`parent, edge, child`), whereas UAS (Unlabeled Attachment Score) evaluates the correctness of the child-parent relation, disregarding the edge label (`parent, child`). We extend these metrics by introducing *child-label* (`edge, child`) and *parent-label* (`parent, edge`). The *Overall* category includes all triples, since the *parent-label* metric is relevant for more than just edges. *Edges* considers only Edge triples, while the subsequent italicized lines correspond to particular subtasks. Specifically, for *Participants*, *Non-participants*, *Arguments*, and *Operands*, Edge triples are filtered based on whether the edge belongs to one of these four categories. More fine-grained phenomena are then evaluated, as described below.
Entities: we evaluate how correctly `refer-number` and `refer-person` are assigned to newly-generated abstract concepts representing entities (see 4.2).
Modal strength: we separately assess if the polarity (`positive`, `negative`) and strength (`full`, `partial`, `neutral`) values are correctly assigned.
Inverted relations: we evaluate the reported metrics exclusively for inverted triples (e.g., `actor-of`).
Abstract predicates (AP): the *predicate* subcategory measures how accurately predicate labels of APs representing core non-verbal clause functions (e.g., `identity-91`) are assigned, considering only Instance triples; *dependents* evaluates how correctly the child nodes of an AP are assigned to it; *ARG nodes* refers to the correct assignment of arguments to the parent, that is the AP.
* To assess the influence of automatic alignment on evaluation metrics, we manually aligned 10 Italian sentences. On this manually aligned sample, we achieved a LAS of 0.277 and a UAS of 0.569.
** LAS measured on the 30 fully manual sentences only.

	Manual		Converted		Time Reduction
	sentence length	time (min)	sentence length	time (min)	
Czech	17.13	31.57	15.29	17.62	44.24%
English	20.13	10.17	18.40	9.35	8.07%
English (2)	16.90	*20.20*	17.50	*10.48*	*48.12%*
Italian	21.23	11.07	19.51	7.66	30.78%

Table 2: Average annotation time (in minutes per sentence) and sentence length (in number of tokens, excluding punctuation) for each language and annotation approach, and observed time reduction from conversion. Italics indicate the less experienced annotator of the English subset.



(a) Czech (manual) (b) English (manual) (c) Italian (manual)



(d) Czech (converted) (e) English (converted) (f) Italian (converted)

Figure 2: Correlation between sentence length and annotation time for Czech, Italian, and English. The x-axis shows the sentence length (number of tokens, excluding punctuation); the y-axis represents the time taken to annotate each sentence in minutes. Each point corresponds to a specific sentence.

Language	Type	Score	
		Pearson	Spearman
Czech	manual	0.660	0.773
	converted	0.658	0.760
English	manual	0.728	0.797
	converted	0.754	0.737
Italian	manual	0.858	0.808
	converted	0.770	0.782

Table 3: Pearson's correlation and Spearman's rank for sentence length (in tokens) *vs.* annotation time.

4.3 Time-based Evaluation

The second evaluation assesses the impact of bootstrapping UMRs from UD on the efficiency of the annotation process, specifically measuring whether converted graphs help annotators work faster. To this end, we compare the annotation time required under two conditions (see Subsection 4.1): (1) 30 sentences are manually annotated from scratch and (2) for 70 sentences, annotators are given the conversion-generated graph and asked to make corrections. For each condition, the annotation time per sentence is recorded and the results are averaged within each group (Table 2). These average times are then analyzed in relation to the sentence length, measured by the number of tokens (Table 3, Figure 2). This approach allows us to assess the effectiveness of the conversion in streamlining the annotation process, particularly as it scales with sentence complexity.

The results confirm that automatic conversion substantially reduces annotation time, though the extent of improvement varies across languages. As shown in Table 2, Czech benefits the most from conversion, with a 44.24% reduction in an-

notation time, followed by Italian (30.78%) and English (8.07%). These differences suggest that language-specific factors may influence conversion efficiency; some languages might inherently benefit more from pre-annotated structures, while others appear to gain less. A key factor is annotator expertise: since the English annotator is the most experienced, the conversion process may have provided limited time savings. In contrast, less experienced annotators may benefit more from pre-converted graphs, as they reduce the need for extensive manual work; this is likely part of the explanation of the longer times and greater time reduction in Czech. To test the role of experience, a less experienced annotator annotated a subset of English sentences.[15] The observed reduction in annotation time (48.12%) supports our hypothesis that experience plays a crucial role in benefiting from converted graphs.

Table 3 investigates the correlation between sentence length and annotation time for both manual and converted approaches. The results confirm that sentence length is a strong predictor of annotation time, with generally high correlations observed across all languages. In most cases, manual annotation exhibits slightly stronger correlations than converted annotation. This suggests that sentence length influences manual annotation time more directly, whereas the conversion approach introduces additional variability, possibly due to errors that require corrections. Despite these differences, the correlations for the converted method remain relatively close to those for the manual method, implying that conversion does not fundamentally alter the relationship between sentence length and annotation time. Instead, it mainly accelerates the process while maintaining a similar complexity pattern.

5 Conclusion and Future Work

In this paper, we introduced an approach to bootstrap UMR graphs from UD trees. The approach was evaluated from two angles: the accuracy (LAS) of generated graphs, and the relative speedup of manual work. Multiple UD-related factors were discussed as possible obstacles for better results (but we cannot measure the impact of each such factor separately). And even if some semantic relations cannot be accurately extracted from syntax, the proposed conversion method has proven to be a valuable tool for annotation. By automating part of the process, it helps to make the annotation workflow faster, reducing the time and effort needed for annotators to complete their tasks. Given the broad availability of syntactic parsers, the potential of this approach is significant. In principle, a dependency parser can be applied to any dataset to generate the syntactic tree, which can then be converted to UMR. This makes the method highly accessible and scalable for a wide range of linguistic datasets.

Future work includes extending evaluation to a broader range of typologically diverse languages to further assess the robustness of the proposed approach. While the current results already demonstrate cross-linguistic applicability, additional testing on languages with different syntactic structures and morphologies will provide deeper insight into the generalizability and limitations of the conversion process. Additionally, refining specific conversion choices—such as improving aspect annotation and integrating named entity recognition (via dedicated NER tools or the Universal NER project (Mayhew et al., 2024)) could enhance semantic accuracy. To maximize the scalability of this approach, we also plan to develop a comprehensive guide to complement the existing technical documentation, making it easier for new users to apply the converter to additional languages.

Acknowledgments

The work described herein has been supported by the grant *Language Understanding: from Syntax to Discourse* of the Czech Science Foundation (Project No. 20-16819X) and by the grants *LINDAT/CLARIAH-CZ* (Project No. LM2023062) and *UMR* (Project No. LUAUS23283) of the Ministry of Education, Youth, and Sports of the Czech Republic. It has also been partially supported by the Charles University, GAUK project No. 104924, UNCE24/SSH/009, and SVV project No. 260 821. Finally, the second author is supported in part by a grant from the CNS Division of National Science Foundation (Award no: NSF_2213805) entitled *Building a Broad Infrastructure for Uniform Meaning Representations*.

References

David Bamman and Gregory Crane. 2006. The design and use of a Latin dependency treebank. In *Proceedings of the Fifth Workshop on Treebanks and*

[15] 10 sentences were annotated manually from scratch, while for 20 sentences the annotator had to correct generated graphs.

Linguistic Theories (TLT2006), pages 67–78. Citeseer.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

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. 2023. 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.

Matthew J. Buchholz, Julia Bonn, Claire Benet Post, Andrew Cowell, and Alexis Palmer. 2024. Bootstrapping UMR annotations for Arapaho from language documentation resources. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2447–2457, Torino, Italia. ELRA and ICCL.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. 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, Sofia, Bulgaria. Association for Computational Linguistics.

Daniel Chen, Martha Palmer, and Meagan Vigus. 2021. AutoAspect: Automatic annotation of tense and aspect for uniform meaning representations. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 36–45, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Federica Gamba. 2024. Predicate sense disambiguation for UMR annotation of Latin: Challenges and insights. In *Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024)*, pages 19–29, Hybrid in Bangkok, Thailand and online. Association for Computational Linguistics.

Federica Gamba and Daniel Zeman. 2023a. Latin morphology through the centuries: Ensuring consistency for better language processing. In *Proceedings of the Ancient Language Processing Workshop*, pages 59–67, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Federica Gamba and Daniel Zeman. 2023b. Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 7–16, Washington, D.C. Association for Computational Linguistics.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for AMR graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319, Online. Association for Computational Linguistics.

Jonas Groschwitz, Shay Cohen, Lucia Donatelli, and Meaghan Fowlie. 2023. AMR parsing is far from solved: GrAPES, the granular AMR parsing evaluation suite. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10728–10752, Singapore. Association for Computational Linguistics.

Jan Hajič, Eduard Bejček, Jaroslava Hlavacova, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague dependency treebank - consolidated 1.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5208–5218, Marseille, France. European Language Resources Association.

Kelvin Han and Siyana Pavlova. 2019. Going from UD towards AMR.

Markéta Lopatková, Eva Fučíková, Federica Gamba, Jan Štěpánek, Daniel Zeman, and Šárka Zikánová. 2024. Towards a conversion of the Prague Dependency Treebank data to the Uniform Meaning Representation. In *Proceedings of the 24th Conference Information Technologies–Applications and Theory (ITAT 2024)*, pages 62–76, Košice, Slovakia. CEUR-WS.org.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Suppa, Hila Gonen, Joseph Marvin Imperial, Börje Karlsson, Peiqin Lin, Nikola Ljubešić, Lester James Miranda, Barbara Plank, Arij Riabi, and Yuval Pinter. 2024. Universal NER: A gold-standard multilingual named entity recognition benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4322–4337, Mexico City, Mexico. Association for Computational Linguistics.

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

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. *Digital Classical Philology*, 10:299–320.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. 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 (LREC-COLING 2024)*, pages 1052–1062, Torino, Italia. ELRA and ICCL.

Ida Szubert, Adam Lopez, and Nathan Schneider. 2018. A structured syntax-semantics interface for English-AMR alignment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1169–1180, New Orleans, Louisiana. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, and 1 others. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Daniel Zeman. 2021. Date and time in Universal Dependencies. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 173–193, Sofia, Bulgaria. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, and 43 others. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.