Accelerating UMR Adoption: Neuro-Symbolic Conversion from AMR-to-UMR with Low Supervision

Claire Benét Post*, **Marie McGregor***, **Maria Leonor Pacheco**, **Alexis Palmer**
University of Colorado Boulder
{benet.post, marie.mcgregor, maria.pacheco, alexis.palmer}@colorado.edu

Abstract

Despite Uniform Meaning Representation's (UMR) potential for cross-lingual semantics, limited annotated data has hindered its adoption. There are large datasets of English AMRs (Abstract Meaning Representations), but the process of converting AMR graphs to UMR graphs is non-trivial. In this paper we address a complex piece of that conversion process, namely cases where one AMR role can be mapped to multiple UMR roles through a non-deterministic process. We propose a neuro-symbolic method for role conversion, integrating animacy parsing and logic rules to guide a neural network, thus minimizing human intervention. On test data, the model achieves promising accuracy, highlighting its potential to accelerate AMR-to-UMR conversion. Future work includes expanding animacy parsing, incorporating human feedback, and applying the method to broader aspects of conversion. This research demonstrates the benefits of combining symbolic and neural approaches for complex semantic tasks.

Keywords: Uniform Meaning Representation, Abstract Meaning Representations, Animacy Parsing, Neuro-Symbolic Learning, Low-Resource Setting

1. Introduction

Meaning representation graphs are hierarchically structured discrete representations of meaning that allow for sentence and document-level meanings to be abstracted away from syntactic structures. They utilize graphical representations where sentences with similar meanings share similar graph structures, even if worded differently. Abstract Meaning Representation (AMR) graphs model sentence-level meanings (Banarescu et al., 2013), and although they can be applied to different languages, the annotation guidelines are closely tied to English, for instance, by not supporting polysynthetic languages. Uniform Meaning Representation (UMR) (Gysel et al., 2021) addresses this limitation by extending AMR to support both sentence and document-level representations, and providing a typologically-motivated, language-agnostic schema for representing meaning.

Direct human annotation of texts with UMR graphs is time-consuming and requires considerable domain expertise. In order to speed up production of data, we take a first step towards automatically converting existing AMR annotations[1] to the more detailed, richer UMR schema.[2] Figure 1 shows a side-by-side comparison. Generating a preliminary graph for annotators to refine, even if noisy, could significantly reduce the human effort required. There are roughly 60,000 annotated English AMR sentences, and parallel UMR annotations previously existed for only about 200 of those. This means we have minimal parallel data from which to train a model on the conversion task.

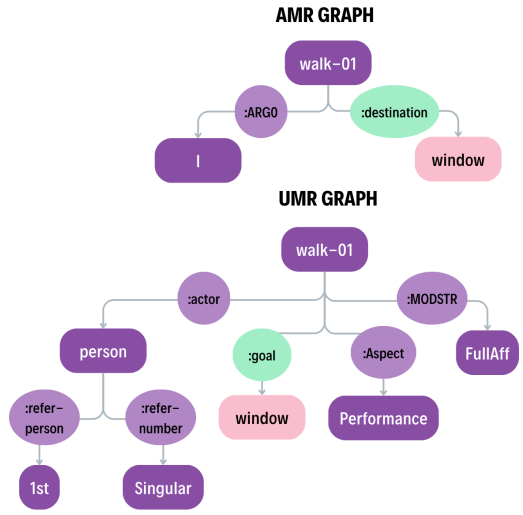

Figure 1: Example of one type of graph conversion of the AMR :destination role to the UMR role :goal in "I walked up to the window".

This paper presents an automated method for partial graph conversion, specifically addressing non-deterministic changes arising from AMR to UMR. [3]. AMR graphs contain individual semantic rolesets that convert into multiple rolesets in UMR. These rolesets between AMR and UMR are known as "split-roles" and contain a non-deterministic,

*Equal Contribution
[1]AMR site: https://amr.isi.edu/.
[2]UMR site: https://umr4nlp.github.io/web/
[3]Our codebase can be foud at: https://github.com/clairepost/AMRtoUMR

5th Workshop on Designing Meaning Representations (DMR 2024) @LREC-COLING-2024, pages 140–150

1:many relationship. This non-determinism motivates our focus on these rolesets, as previous work suggested human annotation would be necessary for their conversion (Bonn et al., 2023).

To address this challenge, we propose a modular, neuro-symbolic framework that utilizes an animacy parser to assist logic rules in automatically determining split roles, minimizing the need for human input in UMR annotation. Our framework combines the flexibility afforded by neural methods to identify patterns in raw data, with a way to promote the schematic constraints of the conversion task. To train and evaluate our framework, we curate a dataset of 587 manually annotated role conversions and 10,635 weakly annotated role conversions, spanning 14 different split role types.

This paper focuses on English AMRs, but the methods presented can be adapted to AMRs in other languages. This adaptability stems from the inherent language-agnostic nature of the underlying graph structure. However, future work in other languages may encounter additional challenges, particularly in accessing an animacy parser. While adapting the approach for AMRs in languages like Chinese may be more feasible due to the availability of resources, languages with limited NLP resources, such as Cherokee, may pose greater difficulties. We limit the scope of these AMR-to-UMR conversions to sentence-level, leaving document-level graph creation for future work.

In summary, we make the following contributions: (1) We frame AMR to UMR conversion as a prediction task, (2) We curate and annotate a dataset focused on split role conversion from AMR to UMR, (3) We propose an extensible, modular framework that combines neural networks and domain knowledge in the form of rules to make this prediction, and (4) We show that we can accurately predict the majority of the non-deterministic roles with limited supervision.

2. Related Work

While AMR has established itself as a powerful tool for semantic representation, its limitations in handling low-resource languages and complex linguistic phenomena hinder its broader applicability. These limitations include challenges with morphology, like polysynthesis, and capturing relationships beyond the sentence level in document-level annotations. UMR, recently proposed by Gysel et al. (2021), offers a compelling alternative with a richer semantic framework and multilingual focus. It introduces document-level representations alongside sentence-level analysis, capturing more nuanced semantic information such as co-reference, temporal, and modal dependencies that go beyond sentence boundaries. However, despite its advantages, UMR adoption is currently hampered by the scarcity of annotated data. This section positions our work within the context of related efforts bridging the gap between AMR and UMR, particularly through automated conversion approaches. Additionally, our efforts complement the work on bootstrapping UMR annotations for low-resource languages, as presented in (Buchholz et al., 2024). This paper provides a non-neural method for UMR graph creation from interlinear glossed text, complementing our focus on the conversion process.

Initial work by Bonn et al. (2023) and Wein and Bonn (2023) provides an analysis of the fine-grained structural distinctions between AMR and UMR, delving into key differences like tense, modality, scope, and document-level dependencies in monolingual and multilingual settings. Building upon this foundation, Bonn et al. (2023) offer a specific road-map for bridging the gap. This paper meticulously details the structural differences between AMR and UMR representation techniques for semantic categories, highlighting crucial aspects like tense, modality, scope, and document-level temporal relations. It also sheds light on the fundamental differences in graph structure, with AMR relying on predicate-argument structures and UMR accommodating polysynthetic and agglutinating languages with more complex morphologies.

By leveraging these insights, our work aims to tackle a key piece of this conversion puzzle. We focus on applying a neuro-symbolic method to address the data scarcity challenge by leveraging domain knowledge and neural learning to facilitate robust and accurate conversion, paving the way for wider UMR adoption and enhanced cross-lingual semantic analysis capabilities. We focus specifically on the non-deterministic roleset changes, contributing to a more robust and comprehensive conversion process.

This work proposes a novel data augmentation approach specifically designed for AMR to UMR role conversion. Our model builds upon the concept of constrained indirect supervision (Wang and Poon, 2018), and combines noisy examples with interdependent label constraints to address data scarcity. Several studies have explored data augmentation for NLP tasks in low supervision settings, including active learning (Quteineh et al., 2020) and rule-based approaches (Zhao et al., 2021). We leverage active learning principles by selecting informative AMR graphs containing split roles like *:destination*, *:cause*, and *:source*. Then, we incorporate animacy parsing, which is crucial for role determination, and derive logic rules from UMR guidelines to generate additional training examples and guide the neural network towards accurate role mappings. This combined approach efficiently utilizes limited labeled data and addresses the chal-

Documents	Number of Sentences	Number of Aligned Split Roles
Lindsay Text	2	1
Phillippines Landslide News Text	28	36
Putin News Text	12	15
Edmund Pope News Text	9	9
Pear Story	141	30
Total	**192**	**91**

Table 1: Parallel AMR-UMR documents with their sentence counts, and the number of split roles

lenges of low supervision settings.

3. Data

The available published parallel AMR-UMR data we utilized consists of five documents (Bonn et al., 2024), all in English, as detailed with sentence counts in Table 1. These documents vary in length and sentence complexity, ranging from short examples, like the *Lindsay Text*, to longer news stories with complex sentence structures, such as the *Philippines Landslides News Text*. In the roughly 200 AMR/UMR graphs, only about 100 split-rolesets are available for analysis.

Although prior literature has indicated the expected split role mapping (Bonn et al., 2023), initial tests have shown that this mapping is not fully captured in the data. Figure 2 shows the counts of AMR and UMR roles from all data overlaying the expected mapping. The data does not reflect a clean 1:many mapping relation. For example, the AMR role *:destination* should split into *:goal* and *:recipient*. The AMR documents consist of 2 instances of the *:destination* role but the UMR documents contain 3 instances of a *:goal* role, meaning that a different AMR role turned into the UMR role *:goal*. This does not reflect the clean splits shown in (Bonn et al., 2023). This analysis highlights the need for a more nuanced approach to role conversion.

3.1. Alignment

To gain deeper insights, we perform partial alignment of AMR and UMR graphs, focusing on the role edges. A partial alignment is possible because the information being captured is just the split-role in question. The meaning representation graphs are directed, node and edge-labeled graphs. Each edge is a semantic relation or role that connects one concept node (the head node) to another concept node (the tail node). In our data, of the 106 AMR roles that we explore, 90 have their head and

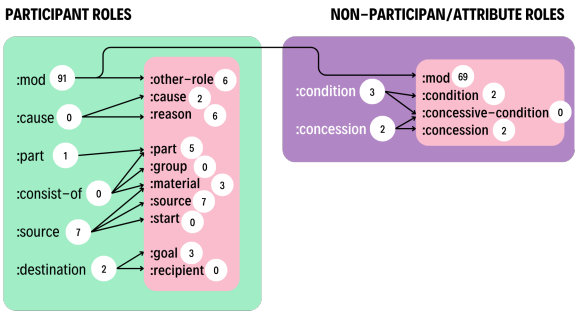

Figure 2: Split role mapping from AMR to UMR with counts from the data

UMR Label	Gold-Standard	Silver-Standard
:group	109	1
:source	90	1168
:goal	59	18
:part	59	94
:mod	58	0
:cause	53	4921
:reason	46	1419
:material	43	176
:start	34	552
:condition	16	2286
:recipient	12	0
:Cause-of	4	0
:other-role	3	0
:Material-of	1	0
Total	**587**	**10635**

Table 2: Counts of UMR Roles in gold-standard data (labels created by human annotators) and silver-star data (labels generated by Rules Model)

tail nodes aligned to corresponding UMR graph nodes, and 70 have a matching edge in the UMR graph. Changes in UMR guidelines and structural differences between the graphs explain most misalignments[4].

3.2. Data Augmentation

Because of the small amount of available parallel data, we use data augmentation to produce more **gold-standard evaluation data** and a large amount of **silver-standard training data**. The resulting dataset statistics are reported in Table 2.

Gold Standard Data To produce additional evaluation data, we employ task-specific data augmentation, leveraging elements of active and curriculum learning techniques (Jafarpour et al., 2021). This approach efficiently utilizes labeled data by manually converting AMR graphs containing split-roles

[4]UMR Guidelines: https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

to provide the most informative samples for training. We converted 40 additional AMR graphs to UMR graphs, preferring graphs that include roles from the less represented splits in the data. Specifically, we focus on the AMR roles of *:destination*, *:cause*, *:consist-of*, and *:source*. The sentences were chosen from the AMR data and guidelines.[5]

In a second augmentation step, we run additional data from the published AMR dataset through the rule-based model detailed in section 4.2. For a targeted set of AMR graphs, an annotator assessed the UMR role assigned by the rule-based model and corrected those labels as needed. This approach yielded 470 additional gold-standard split-role labels.

Silver Standard Data To generate additional, automatically-labeled, and thus noisy, training data, we next run the rest of the non-parallel AMR data through the rule-based model. This data is comprised of around 70,000 additional rolesets. Nearly 60,000 of these are labeled with the *:mod* role, which is both over-represented in the data and nearly always maps to the same role in UMR. For this reason, we exclude *:mod* from the silver-standard training data. The remaining 10,635 rolesets are used as silver-standard training data in Experiment 2 (see section 5).

4. Methodology

Within the broader task of automated AMR-to-UMR graph conversion, we address the specific challenge of non-deterministic role changes, reducing the need for intervention from expert human annotators. This section describes our methodologies for incorporating animacy information and logical rules into a neural architecture.

System Overview We first extract detailed information about roles from both AMR and UMR graphs, including roleset labels, head and tail entities, and their connection to the original sentence and graph context, as explained in section 3.1. An animacy recognition module, detailed in section 4.1, then determines the animacy of each role's tail, as animacy plays a crucial role in UMR role determination.

Next, all of the extracted information serves as input for a rule-based role-labeling component. The rules were formulated manually through our investigation of the logic detailed in the UMR guidelines, and they rely heavily on animacy information, as explained in section 4.2. The rule-based module outputs potential split-role conversions for the AMR role, along with their initial weights, which are determined by analyzing the frequency of role splits based on the implemented rules and the distribution of such splits within the initial UMR published data.

Final role predictions are done by three different models (section 4.3): a baseline rules-only model, a baseline neural network, and a hybrid model combining rules with neural learning. Each model receives the extracted role information, animacy data, and initial weights, utilizing them in different ways to predict the most likely UMR role.

4.1. Animacy parsing

Accurate animacy depiction is crucial for the rule-based decision-making module of our framework. According to the UMR guidelines, certain rolesets should only be used for animate or inanimate entities. Therefore, we test several existing animacy parsers and named entity recognizers (NERs), in addition to using information found within the AMR graph, to synthesize an animacy recognition module tailored to our framework from four components. Certain split-roles, such as *:mod*, were excluded from animacy parsing. Roles such as *:mod* do not need animacy information in order to determine their split, so they were excluded in order to make the model run more efficiently.

1. BERT-Finetuned-Animacy: The first component of the animacy parser is a BERT-finetuned-animacy model (Tobin, 2022). This model takes the sentence to be converted as input, and outputs entities it identifies as persons or animals.

2. BERT-NER: Next, we include a popular NER model, again taking the sentence as input and outputting labeled named entities (person, named organization, named place, and misc) (Lim, 2023).

3. Pronouns: Next, we search for any pronouns within the sentence. While pronouns such as "I", "you", and "she" are not always necessarily animate, they are enough of a proxy for animacy in our data that we chose to include them in the animacy distinction, marking them as "person" roles.

4. AMR Named Entities: The AMR guidelines define various named entities (NEs) in the tails of many role instances. We manually assign animacy labels ("animate" or "inanimate") to each of the NE types. However, akin to the limitations of using pronouns for animacy prediction, this approach overclassifies entities as animate. Overprediction of the "animate" label helps to balance against the animacy parser's tendency to default to "inanimate".

[5]AMR guidelines: https://github.com/amrisi/amr-guidelines/blob/master/amr.md#reification.

Even with over-prediction, the model only produced animate tags as opposed to inanimate tags 2.88% of the time on the full augmented dataset.

Animacy Integration We use the outputs of the various components to make a binary animate/inanimate distinction for each role. First, we check the tail of each role against the items returned as animate. If there is no match for the tail, we next check for a child role, in cases where the tail has a role sense (e.g., *believe-01, leave-14, survive-02*). If there are no matches between the sentence and the outputs of the animacy parser, we treat the role as inanimate.

4.2. Split-Role Rules

Both for prediction and for creating silver-standard data, we encode a set of logical rules capturing tendencies in the mapping of AMR roles to UMR roles. This section details the rules, organized according to original AMR roleset. The rules were created manually as detailed in each section through study of the AMR and UMR guidelines, as well as by referencing UMR examples in our training dataset. In the future, we see the potential to create more rules-based modules to help with the conversion of other split-roles.

Destination Roleset Bonn et al. (2023) substantiate that the AMR *:destination* role splits into the UMR roles *:goal* and *:recipient*. The UMR guidelines additionally specify information about the animacy of certain rolesets. For the *:recipient* role, the UMR guidelines define *:recipient* as an "animate entity that gains possession (or at least temporary control) of another entity". The *:goal* role does not have specified animacy. The resulting rule is that if the AMR *:destination* role is inanimate, the UMR role must be *:goal*. If the AMR *:destination* role is animate, the UMR role may be *:recipient* or *:goal*.

Cause Roleset The second rule addresses the AMR *:cause* role. Similar to the *:destination* role, this role is split using animacy into *:cause* and *:reason*. The UMR guidelines note that the UMR *:cause* role is an "inanimate entity that causes the action to happen." The resulting rule is that if the tail of the AMR *:cause* role is animate then the UMR role must be *:reason*. Otherwise, if it is inanimate, the UMR role may be *:reason* or *:cause*.

Source Roleset The third rule addreses the AMR role *:source*, as illustrated in Figure 3. The *:source* role may split into three different UMR roles: *:source*, *:start*, *:material*. The UMR guidelines give helpful information about animacy for these roles,

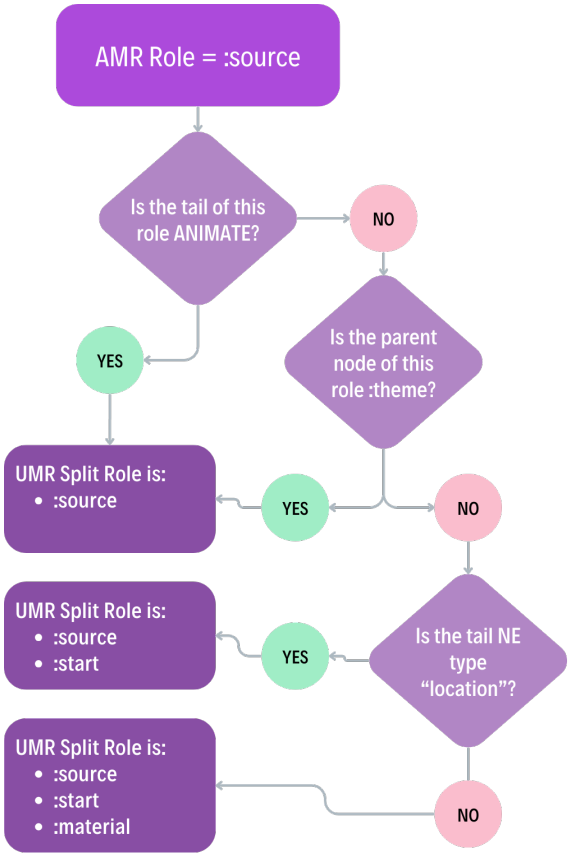

Figure 3: Animacy logic rule for UMR :source, :start, and :material roles from AMR :source role

as well as guidance on the parent role of the instance. For instance, the guidelines provide that the tail of the *:source* roled must be animate. We encode this information by first checking if the tail roleset of the AMR role is animate. If so, the UMR role is set to *:source* since the other roles are generally inanimate. Then, we check if the parent node of the AMR *:source* is *:theme*, as the UMR guidelines specify that *:source* is the "entity from which the *:theme* detaches". In this case the UMR role chosen is *:source*.

Next, the animacy and NE info from the animacy parser is checked to see if it contains a location. If so, the UMR role chosen is *:source* or *:start*. Finally, if the tail roleset of *:source* is inanimate, then the role is either *:source*, *:start*, or *:material*. We obtain initial probabilities for these rule assignments using the distributions observed in the gold-standard UMR graphs (e.g., 0.6 for *:source*, 0.3 for *:start*, and 0.1 for *:material*).

Consist-of Roleset This rule relies on animacy to determine the AMR *:consist-of* role-split. The UMR role *:group* is the only animate role and will always be chosen if the tail AMR roleset is animate. Otherwise, the UMR roles *:group*, *:part*, or *:material* may be the correct split-role choice.

Additional Rolesets The roles *:part* and *:condition* deterministically split into the UMR roles with identical names in English.

The final rule addresses the AMR role *:mod*. This role only rarely maps into the UMR role *:other-role*. Due to the lack of clear rules for this role, we rely on the neural methods to improve prediction accuracy. Initial weights favor *:mod* over *:other-role*.

4.3. Models

We investigate three different models: one using rules alone, one simple neural architecture with no rules, and one combined model.

Rules-only model: For each AMR role, there are 1-3 possible UMR roles. The possible roles are determined by the previously-defined rules, given the AMR role, its predicted animacy, and the AMR graph information. When there is not enough information for the rules to narrow down to just one possible role, the model randomly selects a role label according to the probability distributions seen for that AMR role in the gold-standard parallel UMR data.

Neural Network: Our neural network implementation is a three layer feed-forward neural network. It takes as input the Sentence-BERT embedding of the sentence (Reimers and Gurevych, 2019), concatenated with a feature representing the source AMR role. Although it does not use the animacy rules to influence training, we incorporate external knowledge in constraining the outputs to be only what is possible for the AMR-role to convert to given the UMR guidelines. (For instance, *:destination* can only be converted to *:goal* or *:recipient*). The constraints can be viewed in Figure 2. To train the classifier, we use the cross-entropy loss.

NN with Rule Information: We opt for a simple implementation of a neural network that has access to the rule information in an attempt to leverage the logic of the rules with the predictive power of a neural network. We incorporate the rule information in two ways: 1) We concatenate the probability distribution of the possible roles provided by the rules to the sentence embedding and the AMR role, as the input to the NNet, and 2) We add an additional layer to combine the output (argmax) of the neural network and the rules as: $w_1 * output_{NN} + w_2 * output_{rules}$, where w_1 and w_2 correspond to the trainable parameters of the additional layer. The classifier is then trained end-to-end using the cross-entropy loss.

5. Experiments

We evaluate our three models in two different settings: one training only on gold-standard data, and one adding noisily-labeled (silver-standard) data to the training sets. To better understand how performance is influenced by the difficulty of the particular decision, we categorize the roles into four bins. The first bin, "easy," includes roles with deterministic picks for the English data. The second bin, "medium," consists of roles for which accurate animacy information should lead to accurate role determination. The third bin, "medium/hard," includes roles with more than one choice within each animacy category. Finally, roles in the "hard" bin do not have the benefit of guidance of animacy and have multiple split-roles they could fall into. See Table 5 in appendix for further details.

Experimental Settings For all experiments, we use stratified 5-fold cross-validation and report average results. In each iteration, we use 4 folds for training and 1 for testing. The rules-only model involves no training, so the results shown are based on the predictions for each fold's test set. Results are averaged over 5 runs. Experiment 1, our low-data experiment, uses only the gold-standard data for training and testing. In experiment 2, we use the same folds as experiment 1, now augmenting every fold's training data with the 10,635 silver-standard data points (sec. 3.2). With these settings, we evaluate only on gold-standard data, always include some amount of gold data in training, and ensure comparability across experiments. Our main evaluation measure is macro F1. We also report the weighted F1, which takes into account the label distribution in the test data.

For training, both neural models use a learning rate of 0.001 and train over 50 epochs.

5.1. Experiment 1 - Low data

In this experiment, only gold-standard data is used for training, with an average of just 470 training instances per fold. Per-fold performance is shown in Fig. 4. The Rules model and NN_Rules model perform similarly, and the NN struggles, with high variation across folds. We aggregate the five test sets to evaluate per-class performance as reported in Table 3. The NN_Rules model has the highest F1 score in 7 classes, more than either the Rules model or the NN model. In a small data setting like this, it is not unexpected to see the NN struggle to perform well.

5.2. Experiment 2 - Weak supervision

This experiment combines the gold-standard and silver-standard sets for training, allowing for more

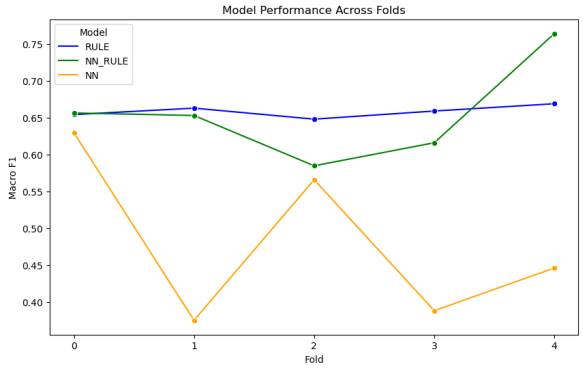

Figure 4: Experiment 1: Macro F1 performance of the three models across 5 folds.

Difficulty	Label	NN	NN_RULE	RULE	support
easy	:condition	1.000	1.000	1.000	16
	:mod	0.838	**0.966**	0.956	58
	:part	0.622	**0.778**	0.775	59
medium	:goal	0.894	0.873	**0.950**	59
	:other-role	0.118	**0.500**	0.000	3
medium/hard	:group	0.724	0.815	**0.913**	109
	:reason	0.000	0.407	**0.653**	46
	:source	0.610	**0.802**	0.689	90
hard	:Cause-of	0.240	**0.857**	0.000	4
	:Material-of	0.000	0.000	0.000	1
	:cause	0.637	**0.743**	0.737	53
	:material	0.419	**0.582**	0.552	43
	:recipient	0.000	0.222	**0.840**	12
	:start	**0.379**	0.351	0.265	34
	macro avg F1	0.463	**0.635**	0.595	
	weighted avg F1	0.603	0.738	**0.761**	587

Table 3: Per-class F1-scores from experiment 1, arranged by the difficulty of the split decision. Bolded values are the highest in each class.

training data in a low-supervision setting. The macro-F1 performance of all of the models across the folds can be seen in Figure 5. Once again, the Rules model and NN_Rules model perform similarly across the folds, and although the basic NN shows reduced performance, there is more consistency across the folds. Per-class performance

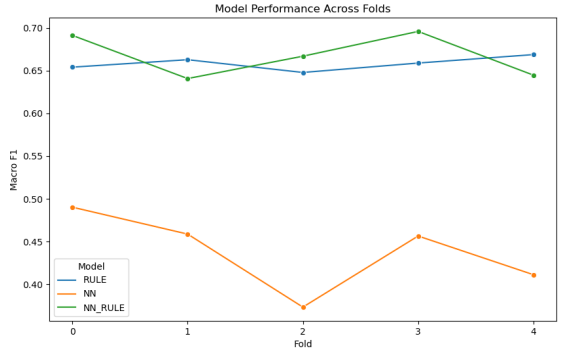

Figure 5: Experiment 2: Macro F1 performance of the three models across 5 folds.

(aggregating all folds) is reported in Table 4. In this

Difficulty	Label	NN	NN_RULE	RULE	support
easy	:condition	1.000	1.000	1.000	16
	:mod	**0.958**	0.948	0.956	58
	:part	0.652	**0.775**	**0.775**	59
medium	:goal	0.894	**0.949**	**0.949**	59
	:other-role	0.000	**0.333**	0.000	3
medium/hard	:group	0.607	**0.936**	0.913	109
	:reason	0.000	0.600	**0.653**	46
	:source	**0.754**	0.685	0.689	90
hard	:Cause-of	**0.114**	0.000	0.000	4
	:Material-of	0.000	0.000	0.000	1
	:cause	0.640	**0.758**	0.737	53
	:material	0.182	**0.645**	0.551	43
	:recipient	0.000	**0.846**	0.840	12
	:start	0.207	0.212	**0.265**	34
	macro avg F1	0.429	**0.621**	0.595	
	weighted avg F1	0.589	**0.767**	0.761	587

Table 4: Per-class F1-scores from experiment 2, arranged by the difficulty of the split decision. Bolded values are the highest in each class.

experiment, the NN_Rules model scores higher than the Rules model on summary statistics, and it achieves the highest score in more classes.

6. Discussion

In this section, we discuss our experimental results, perform a detailed error analysis, and outline directions for future work.

6.1. Experimental Results

In our experiments, we observed that the vanilla neural network struggled to obtain good performance across all configurations. While we saw improvements in the augmented data scenario, the amount and the quality of the supervision was not enough to outperform a simple rule based model. The rule-based model, on the other hand, delivered good average performance across all configurations. This is in line with the highly constrained nature of our task. However, we saw that the NN augmented with rules scored the highest in average when exposed to additional training data. This suggests that hybrid models are a good alternative in weak supervision scenarios, where the neural network can take advantage of the augmented data, while the structured knowledge can help guide the model towards valid answers. For all models, performance was higher for the easy cases than for the hard cases.

To showcase the impact of having access to high-quality annotations, we will consider the *:cause* role. Within the silver-star data, over 40% of the roles were labeled *:cause*. The Rules model performs well on *:cause* with an F1-score of 0.737. Having a large number of high quality labels for training is reflected in the performance of both the NN and the NN with Rules in the *:cause* role. Both models perform better for this class in experiment 2 than they did in experiment 1. Conversely, when examining

the performance for the class *:start*, which has over 500 labeled instances in the silver-standard set, we see that the Rules model's low performance (0.314) adversely affects the ability of the neural methods to predict this class.

6.2. Error Analysis

In this section we discuss specific types of errors made by various models. Some additional examples appear in Table 6, in the appendix.

One prevalent error made by the Rules model comes from adhering to the initial label distributions, as this model has no ability to take context into account. For example, in the sentence *"I saw a cloud of dust"*, the Rules model maps the AMR *:consist-of* role to the UMR role *:group*, for the tail *dust*. In contrast, the NN_Rules model correctly identifies that *:consist-of* should be mapped to the UMR role *:material*. The NN_Rules model leverages learned information to make more informed predictions. All roles in the "medium/hard" category are subject to this type of error.

Another error class occurs when the Rules model fails to make a correct prediction due to inaccuracies in animacy determination. For instance, in the sentence *"A letter from the victim's family,"* the tail role *"family"* was incorrectly parsed as *inanimate*, leading to an incorrect choice of role label. However, the NN_Rules model is not affected by this incorrect animacy parsing, demonstrating better performance in "medium" difficulty scenarios, where correct animacy parsing is needed for the rules to make accurate labeling decisions.

All models encounter difficulty with inverse participant roles such as *:Cause-of* and *:Material-of*. Inverse participant roles, as described in the UMR Guidelines, involve moves like annotating events as modifiers or referring expressions, requiring more complex graph modifications than we currently handle. They are also very infrequent in the data. These rolesets are part of the "hard" category.

Despite similar overall performance to the Rules model, the NN_Rules model shows improvements for roles in "hard," "medium-hard," and "medium" difficulty scenarios. This result highlights the potential of combining symbolic and neural approaches for improved AMR-to-UMR conversion.

6.3. Future Work

Animacy Given the strong influence of the animacy parser, this is an obvious avenue for improvement. Recent studies (e.g. Hanna et al., 2023) highlight challenges for language models in handling subtle shifts in animacy cues within text. While our current approach incorporates animacy information from UMR guidelines, including context-dependent animacy shifts for typical entities, it is still under development in terms of capturing the full spectrum of animacy variations. Additionally, treating animacy as a binary decision might not fully capture the nuances explored in studies like Ji and Liang (2018), which propose a hierarchical spectrum of animacy even within inanimate nouns. For example, "robot" might exhibit more animacy than "chair" due to its potential for movement and agency.

Alternative Modeling Strategies Our NN_Rules model incorporates rules into the neural network in a naive way. In the near future, we intend to investigate alternatives like combining neural networks with Probabilistic Soft Logic (PSL) (Bach et al., 2017) or employing neuro-symbolic methods that leverage rules like DRAIL, a deep relational learning framework (Pacheco and Goldwasser, 2021). An improvement to our current implementation could make use of the full graph-structure of the MRs, instead of just extracting relevant edges. Additionally, different approaches to using and combining the silver-standard and gold-standard datasets could prove beneficial. For example, curating the silver-standard data to remove the labels from low-quality classes, and using a split of the gold-standard data during development, may leverage the strength of both datasets more effectively.

Expansion to other UMR Components In the future, we believe this methodology can be applied to other parts of the AMR-UMR conversion process, starting with expansion to all of the semantic roles, not just this subset of role changes. By thoughtfully constructing rules, we can potentially aid annotators throughout the entire annotation process. Graph preprocessing approaches like handling inversion and reification could prove beneficial to more complex changes.

User Study In the spirit of demonstrating the usefulness of our tool to the UMR annotator audience it is intended for, we propose an experiment evaluating its impact on annotation speed and accuracy. This experiment would involve experienced UMR annotators working on two sets of AMR graphs each:

1. Traditional: Annotators complete the conversion task without any additional information or assistance.
2. Tool-assisted: Annotators leverage our model's predicted split-role conversions alongside the AMR graphs.

By comparing annotation times and accuracy between the two groups, we can assess the potential

benefits of our tool in expediting and potentially improving the UMR annotation process. This evaluation aligns with our goal of providing UMR annotators with valuable resources to streamline their workflow.

7. Conclusion

This work presented a novel, modular methodology for automated AMR-to-UMR graph conversion, with a primary focus on accurately predicting non-deterministic role changes that often require human intervention. Our approach integrates animacy parsing, logic rules, and neural learning to achieve promising accuracy.

Key contributions include introducing a modular framework for easy integration with future techniques, promoting extensibility and broader applicability. Furthermore, the incorporation of animacy information enhances decision-making in role prediction, while the fusion of structured knowledge with neural learning offers flexibility and robustness. The model's encouraging performance on the test data highlights its potential to streamline the conversion process and thus accelerate UMR adoption.

While acknowledging the promising results, we recognize limitations arising from data scarcity and the binary representation of animacy. Future work will involve expanding animacy parsing to capture richer semantic information and context-dependent nuances, potentially employing non-binary representations to improve accuracy. Additionally, user studies will be conducted to assess the impact of our methodology on UMR annotation speed and accuracy, providing valuable insights into its practical utility. Finally, we envision expanding our approach to encompass broader aspects of AMR-UMR conversion, further contributing to the advancement of cross-lingual semantic analysis and unlocking the full potential of UMR for multilingual NLP tasks.

This research demonstrates the benefits of combining symbolic and neural approaches for complex NLP tasks in data-constrained scenarios. By overcoming data scarcity challenges and facilitating accurate UMR conversion, our method paves the way for enhanced cross-lingual semantic analysis capabilities, ultimately impacting various NLP applications that rely on accurate semantic representation and understanding.

8. Limitations

Animacy, the distinction between animate and inanimate entities, plays a crucial role in determining split roles within our rule-based model. It influences the roles a referent can take on, for instance, requiring animacy for the agent role. While existing animacy classifiers like those presented in Tobin (2022); Jahan et al. (2018) exist, they can be imperfect and miss participants within sentences where animacy is nuanced or context-dependent. This limitation can lead to inaccurate role predictions in certain cases.

As well, this work faces several data-related challenges that limit the scope of model development. The limited availability of parallel AMR-UMR annotations, consisting of an extremely small dataset of only 200 graphs from five documents (see Table 1), constrained our ability to train and evaluate models effectively. Moreover, inconsistencies between expected and observed role mappings (as illustrated in Figure 2) suggest a more nuanced conversion process than a simple 1:many relationship, complicating model training and interpretation. Our current focus on sentence-level conversion also limits the applicability of our model to larger discourse contexts. And finally, data imbalances, particularly with over-represented roles like ":mod," created issues in the analysis and data augmentation steps.

9. Acknowledgements

This work is supported by a grant from the CNS Division of National Science Foundation (Award no: NSF_2213805) entitled "Building a Broad Infrastructure for Uniform Meaning Representations." We extend our sincere gratitude to Julia Bonn for her invaluable insights, suggestions, and expertly crafted spreadsheets elucidating AMR and UMR. Special thanks are due to the participants of the CU Boulder 2023 Neuro-Symbolic Approaches to NLP class for their constructive feedback during the development of this work. Lastly, we express our appreciation to the reviewers for their helpful comments and feedback.

10. References

Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-Loss Markov Random Fields and Probabilistic Soft Logic. *Journal of Machine Learning Research (JMLR)*, 18(1):1–67.

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.

Julia Bonn, Matthew 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 B. 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 an Infrastructure for Uniform Meaning Representations. Language Resources and Evaluation (LREC), The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation.

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 Buchholz, Julia Bonn, Claire B. Post, Andrew Cowell, and Alexis Palmer. 2024. Bootstrapping UMR Annotations for Arapaho from Language Documentation Resources. Language Resources and Evaluation (LREC), The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation.

Jens Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Timothy J. O'Gorman, Andrew Cowell, W. Bruce Croft, Chu-Ren Huang, Jan Hajic, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a Uniform Meaning Representation for Natural Language Processing. *KI - Künstliche Intelligenz*, 35:343 – 360.

Michael Hanna, Yonatan Belinkov, and Sandro Pezzelle. 2023. When Language Models Fall in Love: Animacy Processing in Transformer Language Models. *arXiv preprint arXiv:2310.15004*.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active Curriculum Learning. In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 40–45, Online. Association for Computational Linguistics.

Labiba Jahan, Geeticka Chauhan, and Mark A Finlayson. 2018. A New Approach to Animacy Detection. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1–12.

Jie Ji and Maocheng Liang. 2018. An animacy hierarchy within inanimate nouns: English corpus evidence from a prototypical perspective. *Lingua*, 205:71–89.

David S. Lim. 2023. Model: Bert Base NER.

Maria Leonor Pacheco and Dan Goldwasser. 2021. Modeling Content and Context with Deep Relational Learning. *Transactions of the Association for Computational Linguistics*, 9:100–119.

Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. 2020. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Andrew Tobin. 2022. Model: Bert Finetuned Animacy.

Hai Wang and Hoifung Poon. 2018. Deep Probabilistic Logic: A Unifying Framework for Indirect Supervision. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Shira Wein and Julia Bonn. 2023. Comparing UMR and Cross-lingual Adaptations of AMR. *Proceedings of the 4th International Workshop on Designing Meaning Representations*.

Xinyan Zhao, Haibo Ding, and Zhe Feng. 2021. GLaRA: Graph-based labeling rule augmentation for weakly supervised named entity recognition. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3636–3649.

A. Appendix

Role	Animacy	Determinate	Choices #	Difficulty	Note	Comboned Class Difficulty
	animate	n/a	0			
:Cause-of	inanimate	no	2	hard	impossible for rules	hard
	animate	n/a	0			
:Material-of	inanimate	no	3	hard	impossible for rules	hard
	animate	n/a	0			
:cause	inanimate	no	2	hard		hard
	animate	n/a	0			
:condition	inanimate	n/a	2	easy		easy
	animate	no	2	hard		
:goal	inanimate	yes	1	medium	as long as animacy is correct	medium
	animate	yes	1	medium	as long as animacy is correct	
:group	inanimate	no	3	hard		medium/hard
	animate	n/a	0			
:material	inanimate	no	3	hard	can come from both consist-of and source	hard
	animate	n/a	2	easy		
:mod	inanimate	n/a	2	easy		easy
	animate	n/a	2	medium	very random and hard to determine	
:other-role	inanimate	n/a	2	medium		medium
	animate	n/a	0			
:part	inanimate	yes	1	easy		easy
	animate	yes	1	medium	as long as animacy is correct	
:reason	inanimate	no	2	hard		medium/hard
	animate	no	2	hard	because animacy could be wrong and still need to pick	
:recipient	inanimate	n/a	0			hard
	animate	yes	1	medium	as long as animacy is correct	
:source	inanimate	no	3	hard		medium/hard
	animate	n/a	0			
:start	inanimate	yes	3	hard		hard

Table 5: Difficulty of the decision of each role, reflected in the number of possible roles the model must choose from, even with the animacy information and the rules.

ID	Sentence	AMR Role	Tail	Animacy	UMR role	Rules Prediction	NN Prediction	NN with Rules Prediction	Analysis
1	I saw a cloud of dust.	:consist-of	dust	inanimate	:material	:group	:group	:material	Bad distrubtion pick
2	"U-m and he's ge he's getting down out of the tree,"	:source	tree	inanimate	:source	:start	:source	:source	Bad distrubtion pick
3	That's not right or fair and I think it's unhealthy for you because you're blaming yourself when he's effectively made a bigger mistake.	:destination	you	animate	:recipient	:goal	:goal	:recipient	Bad distrubtion pick
4	A Letter from the Victim's Family	:source	family	inanimate	:source	:start	:source	:source	Innacurate animacy
5	After the disintegration of the former Soviet Union , these troop clusters were transferred to Russian ownership .	:consist-of	troop	inanimate	:group	:material	:group	:part	Innacurate animacy

Table 6: Error analysis of several common error types ran from Experiment 2.