NEUROSPECT: Automating Aspect Annotation for UMR

Alvin Po-Chun Chen, Claire Benét Post, Saksham Khatwani
Karthik Sairam, Paul Bontempo, August Milliken
Nicholas Derby, Sumeyye Nabieva
University of Colorado Boulder
{alvin.chen, benet.post, saksham.khatwani,
karthik.sairam, paul.bontempo, august.milliken,
nicholas.derby, sumeyye.nabieva}@colorado.edu

Abstract

Uniform Meaning Representation (UMR) is successor to Abstract Meaning Representation (AMR), graph-based framework for representing the semantics of natural language data. Graphs from both frameworks require extensive effort by trained annotators to produce, motivating the need for automating parts of the annotation process. While current approaches struggle to produce faithful UMR graphs from natural language inputs, converting existing AMR graphs (of which there are plenty) is a more tractable task. A key part of the conversion process requires incorporating features added to the UMR framework that are not present in AMRs. One such addition is aspect, which marks the temporal structure of eventive predicates. In this paper, we introduce RULESPECT and GRAPHSPECT, two novel methods for producing UMR aspect annotations using existing AMR graphs and purely natural language inputs, respectively.

1 Introduction

Aspect annotation is a core component of Uniform Meaning Representation (UMR), a graph-based semantic framework designed to represent meaning in a cross-linguistically applicable and computationally tractable way (Van Gysel et al., 2021). Unlike tense—which encodes *when* an event occurs—aspect captures the *how*: the internal temporal structure, duration, and completion of events (Comrie, 1976; Croft, 2012; Donatelli et al., 2018). It allows a semantic system to distinguish between, for example, habitual, ongoing processes, or completed achievements, enabling a more nuanced interpretation of event semantics.

In UMR, aspect is applied to all *eventive* elements, otherwise known as *eventualities*, in a sentence—typically the concept aligned with the main finite verb, as seen in Figure 1. However, the label refers to the full predication, encompassing the verb and its arguments (Donatelli et al., 2019;

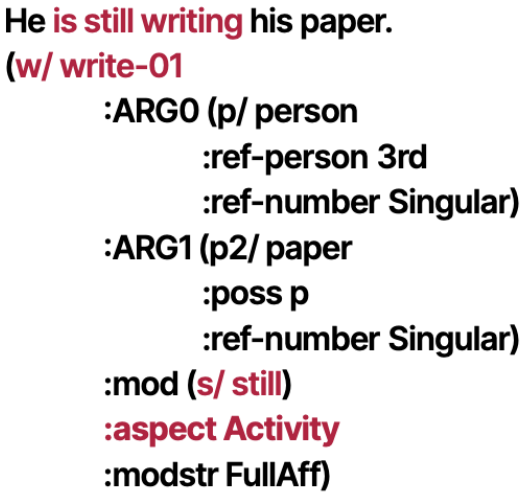

Figure 1: Example UMR graph with the eventive item highlighted and the *Activity* aspectual marker.

Kingsbury and Palmer, 2003). These aspectual categories align with well-established event typologies, such as states, activities, accomplishments, achievements, and processes (Bach, 1986), organized into a lattice that supports both coarse- and fine-grained granularity. Unlike surface-level grammatical cues (e.g., auxiliaries or verb morphology), UMR aspect is a semantic feature meant to generalize across languages, and thus it is informed by deeper event structure rather than morphosyntactic form that can generalize across typologically diverse languages (Van Gysel et al., 2021).

Annotating aspect is no simple feat. Theoretical debates span decades, including disagreements about the universality of aspectual categories, the granularity of classifications, and their interaction with tense and modality (Reichenbach, 1947; Vendler, 1957; Comrie, 1976; Langacker, 2011; Dowty, 1986; Hinrichs, 1986; Moens and Steedman, 1988; Klein, 2013; Chang et al., 2022; Partee, 2011; Croft, 2012). There are also debates on its annotation corpora and how to go about computation-

ally modeling (Pustejovsky et al., 2003; Derczynski, 2017; Pustejovsky et al., 2017; Friedrich and Palmer, 2014; Friedrich et al., 2016; Mostafazadeh et al., 2016; Laparra et al., 2018; O'Gorman et al., 2016).

From a typological perspective, some languages encode aspect more saliently than others, further complicating annotation for multilingual or cross-linguistic frameworks. For example, American Sign Language and Mandarin Chinese prioritize aspectual distinctions over tense (Li and Thompson, 1989; McDonald, 1982), while Hindi includes a dedicated aspect morpheme separate from tense or mood (Van Olphen, 1975). In contrast, many Indo-European languages conflate aspect and tense morphologically, often obscuring the underlying semantic distinctions.

Given these complexities, manual aspect annotation is time-consuming, error-prone, and highly sensitive to annotator interpretation. Yet its inclusion in UMR is foundational to achieving a more expressive, cross-linguistic meaning representation system. UMR builds on earlier formalisms such as Abstract Meaning Representation (AMR) (Banarescu et al., 2013), where aspect was initially introduced to support event-based reasoning but was never fully adopted into standard annotation guidelines. Donatelli et al. (2018, 2019) formalized aspect in UMR, laying out annotation principles and aligning event types with lexical frames.

To address the bottleneck of manual annotation and support UMR's broader adoption, we propose a system for automatically assigning aspectual labels to English meaning representation graphs using neural and neurosymbolic approaches. We address two central tasks:

- **Task 1:** Given an AMR graph, can a language model predict the appropriate UMR aspect label? This task treats the semantic graph as a structured input to train graph-aware models that predict aspect labels.

- **Task 2:** Can we bypass the graph altogether and predict UMR aspects directly from the raw text? This task explores whether large language models can map natural language to aspect categories in the absence of explicit semantic parsing found in the AMR or be found more efficiently through other pipelines. This method necessitates finding both the predicates and the aspects of those predicates. This method is ideal given the future use case of having an automated UMR parser that does not rely on partial AMRs.

To support both tasks, we committed to a large-scale annotation project that resulted in a new dataset of $1,184$ manually annotated aspectual labels added onto the eventive predicates of AMR graphs. This annotation effort provides gold-standard supervision for model training and evaluation.

We develop and evaluate two primary modeling approaches: RULESPECT, a neurosymbolic model that integrates AMR graph structure, sentence text, and hand-engineered symbolic rules; and GRAPHSPECT, a fully neural graph-attention pipeline that operates over sentence text alone.

These are evaluated against a set of benchmark systems, each designed to highlight different modeling challenges and trade-offs. The benchmarks include an attempted comparison against AUTOASPECT, a prior system developed in Chen et al. (2021a) for AMR graphs. We also tested Large Language Model (LLM) prompting using few-shot or in-context learning setups. This baseline serves two purposes: (1) as a performance lower bound for non-graph-based models, and (2) as a generator of silver-standard data used for pretraining or augmentation in our main models. Finally, a benchmark classifier with embeddings that tests whether static or contextual word embeddings extracted from LLMs e.g., sentence-level embeddings from BERT or GPT) are sufficient for aspect prediction.

Our findings highlight the value of integrating symbolic reasoning and graph structure for aspect classification, while also demonstrating the feasibility of text-only models for future automated UMR pipelines. While our current results show LLM to be the most promising strategy, previous experiments on smaller subsets of the data demonstrated that our neurosymbolic approach can outperform LLM prompting in certain cases.

2 Related Work

In this section, we further detail information on UMR annotation, describe other automated annotation methods, and provide an overview for graph neural networks and neurosymbolic approaches.

2.1 Aspect Representation

The semantics of aspect has been a long-standing topic of debate in linguistic theory. Seminal works by Reichenbach (1947), Vendler (1957), and Comrie (1976) lay the foundation for distinguishing between types of eventualities—states, achievements, activities, accomplishments—based on their temporal and structural properties. Dowty (1986) and Langacker (2011) further explore the interaction between aspect, argument structure, and lexical semantics. These formalisms inform how events are modeled in UMR today.

Later developments such as Hinrichs' interval-based models (1986), Moens and Steedman's narrative structure theory (1988), and Klein's temporal logic (2013) introduce more formal ways to encode event structure and its temporal entailments. These insights highlight the need for frameworks like UMR to go beyond grammatical tense and directly encode aspectual distinctions based on semantic content.

Aspect has been incorporated into semantic annotation and event modeling efforts, particularly in temporal information extraction. TimeML (Pustejovsky et al., 2003) and its follow-up projects such as the TempEval competitions (Derczynski, 2017) include annotation for aspect, though typically via shallow textual cues. More recent work seeks to automate aspect classification using linguistic features (Friedrich and Palmer, 2014), discourse roles (Friedrich et al., 2016), and LSTM-based models that integrate context (Mostafazadeh et al., 2016; Laparra et al., 2018).

While effective to some degree, these systems often operate over flat text or shallow syntactic representations. They do not handle the rich predicate-argument structures or graph-based semantics found in UMR or AMR. Moreover, they treat aspect as a downstream feature, rather than an integral part of event structure representation.

Efforts to include aspect in AMR were initiated by Donatelli et al. (2018), but aspect is not part of AMR's core schema. On the other hand, UMR incorporates aspect explicitly into its annotation guidelines, enabling more structured reasoning about events across languages. This is part of a broader effort to develop UMR into a multilingual semantic representation system, as outlined by Van Gysel et al. (2021).

This paper is concerned with the aspect annotation of English sentences. Per Van Gysel et al. (2021), the aspectual categories chosen for English annotation include a set of base-level distinctions—*Performance*, *Endeavor*, *Activity*, *State*, and *Habitual*—and a more coarse-grained value for event nominals amongst other more coarse grained events—*Process*.

2.2 Automating Annotation for UMR

Due to the small amount of available UMR data, prior work has focused primarily on methods for generating UMR graphs without training. Chun and Xue (2024) propose a multi-step strategy for converting AMR graphs into UMR graphs by using a variety of existing automation tools, such as by using a modal dependency parser. Similarly, Sun et al. (2024) experiment with few-shot and Think-Aloud prompting on LLMs to generate Chinese UMR graphs without AMR data as input. While these approaches produce positive results, we find in our investigations that training is necessary for further improvement.

Our task is primarily influenced by AUTOASPECT, a rules-based approach specifically for classifying UMR aspects in English UMR graphs (Chen et al., 2021b). These rules can incorporate contextual information around the predicate such as auxiliary verbs (e.g. *have written* vs. *wrote*) and completive markers (e.g. *wrote the whole paper* vs. *wrote the paper*) to indicate what the aspect of the predicate might be. For their approach, Chen et al. (2021b) develop a structured set of rules which closely follow the UMR annotation guidelines and decision lattice. Our work seeks to address the obstacles that AUTOASPECT encountered with edge-case aspect relations and with inaccurate event identification.

Semantic Role Labeling (SRL) is the task of identifying the events and arguments of a given phrase, essentially defining *who* did *what* to *whom*, and sometimes *how*. SRL is a mainstay task of the NLP world and many tools have been developed over the years to label a sentence's semantic roles, especially for English data. This widespread task led to the creation of the Proposition Bank, a database resource which simplifies and supports SRL implementations (Kingsbury and Palmer, 2002; Pradhan et al., 2022). In our graph network approach that only takes in a sentence as input, we leverage an SRL tool (which is itself built on PropBank rolesets) to identify the events and their associated arguments, in order to aid in aspect

prediction (Gardner et al., 2017).

2.3 Neural Methods

Graph Neural Networks (GNNs) (Defferrard et al., 2017) are useful for capturing complex relationships represented as a graph. There are various kinds of GNNs such as Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) and Graph Attention Networks (GATs) (Schlichtkrull et al., 2017) which are useful for NLP tasks. In Wang et al. (2022), the authors introduce a text classification framework using GCN called InducT-GCN, which creates nodes for training documents and unique words in those documents. Word nodes are represented using one-hot vectors, while document nodes are represented using TF-IDF vectors.

Neurosymbolic approaches are a way to incorporate some logical structure into the model, either by modifying certain layers of the model, such as Li and Srikumar (2020), or utilizing a teacher-student framework for distillation of symbolic knowledge in the model (Hu et al., 2016). Both Li and Srikumar (2020) and Hu et al. (2016) use first order logic rules as the symbolic knowledge incorporated in the model. The rules are scored using t-norms (Bach et al., 2017), which also allow for the rules to be differentiable. Neurosymbolic approaches are also helpful in maintaining consistency in the models.Li et al. (2019) utilize logic rules to regularize the models away from inconsistency.

3 Data

Our dataset is sourced from the soon-to-be-released UMR 2.0 Dataset which contains roughly 30k UMR graphs in different stages of conversion from AMR graphs from Knight et al. (2020) and Bonn et al. (2020). Due to the lack of UMR aspect data, we annotate aspect labels for part of the dataset that has yet to be annotated with aspect. To ensure broad coverage for training and evaluation, we select four corpora in the dataset to annotate:

1. The Little Prince corpus, a set of sentences from the English translation of The Little Prince by Antoine de Saint-Expupéry.

2. The Minecraft corpus, a set of dialogues and corresponding grounding data from a collaborative structure-building task in Minecraft (Narayan-Chen et al., 2019).

3. The BOLT DF corpus, which contains English-language forum posts crawled as part of the DARPA BOLT project.

4. The Weblog corpus, comprised of weblog and online news articles.

Table 1 shows the distribution of aspect labels and inter-annotator agreement across the annotated corpora as well as the distribution of labels for the existing dataset. A detailed summary of aspect label statistics by corpora for the existing dataset can be found in Appendix A.

3.1 Annotation

Given the complexity of aspect annotation and its theoretical underpinnings, we determined early on that all members of the team should develop a strong understanding of the UMR aspect schema. Our goal was to ensure that each annotator not only contributed data but also possessed the linguistic foundation necessary to support modeling decisions later in the project. To this end, we conducted weekly training sessions throughout the annotation phase.

Training materials were primarily drawn from expert annotator Julia Bonn's Georgetown tutorials and supplemented with newly designed resources, including an accessible slide deck[*] summarizing the UMR guidelines [†] with added clarifications and examples. These materials — available in a shared Google Drive folder[‡] — formed the core of our instructional sessions. Each week for several months, team members presented on different topics from these materials and discussed example annotations as a group to clarify issues.

To solidify our understanding, we conducted an initial practice task in which each team member annotated up to 50 predicates from the Pear Story corpus (Bonn et al., 2023). This dataset was selected for its short, visually grounded sentences, which reduced ambiguity and facilitated discussion. We then held a follow-up session to review inter-annotator disagreements and recurring errors.

[*] https://docs.google.com/presentation/d/1QUAnh2LWlgfvp_0NAj7K-YEdAbjG7zXmCLfEq0sfoa0/edit?usp=sharing

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

[‡] https://drive.google.com/drive/folders/1_ou3WW4UV7gQHtglMTEbu4TlTbO5xdHt?usp=share_link

Aspect	Little Prince	Minecraft	BOLT DF	WB	Existing Labels	Total
State	172	14	101	14	430	731
Habitual	41	0	4	0	52	97
Process	31	0	38	8	58	135
Activity	15	2	10	3	132	162
Performance	163	43	69	32	317	624
Endeavor	15	0	4	0	16	35
None	158	49	121	77	-	405
Total	**595**	**108**	**347**	**134**	**1,005**	**2,189**
Fleiss' Kappa	0.78	0.82	0.45	0.40	-	-

Table 1: Label Distribution by Corpus and Aspect. We report Fleiss' Kappa between the two initial annotators and do not include disagreements in the reported total.

Category	Metric	Value
Accuracy	State	0.82
	Habitual	0.63
	Activity	0.52
	Performance	0.80
	Endeavor	0.11
	Overall Accuracy	0.74
	Perfect Accuracy	0.35
F1	Macro F1	0.49
	Weighted Macro F1	0.76
IAA	Fleiss' Kappa	0.55
	Gwet's AC1	0.66

Table 2: Practice Annotation Results: *Overall Accuracy* is the ratio of the total number of correct annotations over the total number of predicates annotated. *Perfect Accuracy* is the ratio of predicates that were correctly annotated by all annotators. No occurrences of *Process* aspects were present in the practice set.

Table 2 shows the results from the practice task. Label-wise accuracy showed that some categories — such as *State* and *Performance* — were more reliably identified, while others like *Endeavor* and *Habitual* were less consistent. These findings guided a focused error correction session in which we reviewed common sources of confusion, such as distinctions between *State* vs. *Performance* and *Performance* vs. *Endeavor*. Due to the different number of annotations each person performed, we also report Gwet's AC1 as a measure for inter-annotator agreement (IAA) since this metric can be calculated for different numbers of labels. Fleiss' Kappa is only calculated for predicates that were labeled by all annotators. We find moderate to good IAA for the practice round, motivating the need for additional training.

Following the practice round, we moved into full-scale corpus annotation. Each subcorpus was assigned to two annotators for independent labeling. When disagreements arose, they were resolved through a tie-breaking process in which a third annotator made the final decision. Each numbered predicate within the AMR graphs was annotated with one of six UMR aspect labels or marked with NONE if the node was non-eventive. The *NONE* label was frequently used for adjectival or adverbial concepts, which often receive FrameNet mappings in AMR but do not participate in eventualities.

In addition to aspect labeling, each graph required alignment between graph variables and their corresponding word indices in the sentence. This alignment step was essential for enabling supervised learning over surface forms and graphs. Annotators manually aligned each node (e.g., `s2o2`) to its referring word span using index pairs (e.g., `s2o2: 6-7`). Discontinuous spans were represented using comma-separated index pairs (e.g., `s95w: 4-4,7-7`). Overall, this manual process tended to take more time than the actual aspect annotations themselves.

Once initial annotation and alignment were complete, all sentences with conflicting aspect labels were routed to a tie-breaker process. A third annotator reviewed the original annotations and sentence context to make a final determination. Tie-breaking decisions were recorded in a shared sheet for transparency and consistency.

Throughout the project, the team met regularly to discuss edge cases, resolve confusion, and refine our shared understanding of the guidelines. For particularly complex disagreements—such as

differentiating *Endeavor* from *Performance*—we consulted directly with Julia Bonn to align our annotations with expert interpretations.

3.2 Neurosymbolic Rules

We analyzed the properties of the alignment labels and made some generalizations that we are incorporating as first order rules in the proposed *RuleSpect* model. Let us say that our model is able to predict if an event has: 1. Ended and 2. Completed. Then, we can make the following rules.

1. If the model predicts an event as **Performance**, then it must also predict that it has **Ended** and **Completed**.

2. If the model predicts an event as **Endeavor**, then it must also predict that it has **Ended** and **Not Completed**.

3. If the model predicts an event as **Activity**, then it must also predict that it has **Not Ended**.

The above rules can be expressed in terms of first-order logic as:

$$\forall n \in \text{Performance},\ n \Rightarrow \text{End} \wedge \text{Complete} \quad (1)$$

$$\forall n \in \text{Endeavor},\ n \Rightarrow \text{End} \wedge \neg\text{Complete} \quad (2)$$

$$\forall n \in \text{Activity},\ n \Rightarrow \neg\text{End} \quad (3)$$

Furthermore, our analysis of the predicates also revealed that certain predicates are more likely to be certain aspect values. For example, *say-01* is more likely to be **Performance**, and cannot be **State**. Similarly, for certain predicates, we are keeping track of predicates and are looking to incorporate additional rules based on the observations. Work was previously done with VerbNet (Schuler, 2005), in part by Julia Bonn, to classify the inherent aspectual qualities of certain classes of verbs.

For instance, the "hunt" verb, and the verbs within its class, entail an *Endeavor* aspect because, unless otherwise specified, the action of hunting and finding the object of the hunt is incomplete. This logic extends to many other verbs, but has not been formalized with the PropBank propositions that AMR or UMR use. Efforts are currently underway to match up these propositions to their possible aspectual values as an extension of the rules guided method.

4 Methodology

In this section, we first detail our investigation of three methods that we use to benchmark our performance before presenting our proposed approaches, RULESPECT and GRAPHSPECT. Our first benchmark is a re-implementation of the AUTOASPECT system to use as a baseline. For our second benchmark, we prompt LLMs with a few prompts to evaluate the performance of models out of the box and to determine the feasibility of using LLMs to generate noisy data on which to train. For our final benchmark, we train a simple feed-forward classifier using the contextual embeddings of the surface verb represented by each predicate.

4.1 Re-Implementing AutoAspect

We attempt to reimplement the AutoAspect rules-based classifier on our novel set of annotated UMR graphs, in order to compare its performance against the neural approaches as a benchmark. However, due to dependency issues with the semantic parser used by the original AutoAspect code, we are unable to report this benchmark on our dataset, and instead provide the AutoAspect classifier's performance on the dataset with which it was published, as a reference for rules-based approaches in general. Model details and hyperparameters can be found in Appendix D.

4.2 Large Language Model Prompting

We prompt several LLMs to evaluate their capability for aspect annotation out-of-the-box. By running this experiment, we both produce a baseline to compare other methods and investigate the possibility of using LLMs to generate synthetic data for training. Prior literature demonstrates that noisy data can be used to train models (Jung et al., 2024) that outperform the teacher model, meaning that the results from LLM prompting are useful even if the approach performs sub-optimally.

Although finding the optimal prompt for our task is intractable, we try three strategies to gauge the impact of prompt structure on LLM performance. Initially, we manually draft a list of short definitions for each aspect based on the learnings from our annotator training session. We then provide the initial prompt and instruct the model to generate a better prompt for our task. Finally, we simply provide the UMR guidelines for aspect [§] in their entirety

[§] github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md#

and prompt the model to predict a label using the rules and examples therein. We test our strategies on three LLMs: Llama-3.1 8B Instruct (Grattafiori et al., 2024), GPT-4o (OpenAI et al., 2024), and DeepSeek V3 (DeepSeek-AI et al., 2025).

We were unable to implement the third strategy for the Llama model due to its limited context window, so instead we created several conditional rules (negation, hypotheticals, commands) based on the UMR guidelines. Furthermore, we also find that the Llama model performed significantly worse when the text of UMR graphs were presented so for these experiments we only provide the surface form of the sentence as input. We report the results for the best performing prompt for evaluation, details on prompts can be found in Appendix B and details on cost can be found in Appendix C.

4.3 Feed Forward Classifier

Although our experiments with LLM prompting produce results that are sufficient for generating synthetic data, we do not find any useful strategies for further improving our results. Performance from in-context learning has been shown to vary drastically based on slight changes to prompt structure (Lu et al., 2022), and other work suggests that LLMs lack meta-linguistic reasoning capability (Bonn et al., 2024). Our second method investigates the ability of LLMs to capture representations that may be useful for aspect classification based on the hypothesis that contextual embeddings encode a broad range of linguistic phenomena (Arora et al., 2024).

To evaluate the usefulness of LLM embeddings out of the box, we pass the natural language sentence to Llama-3.1 8B (Grattafiori et al., 2024) and select the first output embedding corresponding to the predicate under consideration. We then train a simple Feed Forward Network to predict one of the seven aspect labels using that embedding. If the contextualized embedding accurately captures the aspect of each predicate, our model should be able to classify that aspect with relative accuracy. Additionally, the results from this method also serve as a useful benchmark for evaluating the performance of more complex strategies. Details of the model implementation can be found in D.

`part-3-3-1-Aspect`

4.4 RuleSpect

To address our first task, we introduce RULESPECT, a GNN which takes AMR/UMR graphs as input and is augmented with neurosymbolic rules for training. For implementation, we first create a graph utilizing the UMR graph and BERT-large embeddings of the sentence. Using the Penman Library, we parse the UMR graph and extract all the nodes and edges, creating a directed graph data-structure where the nodes represents the predicates in the UMR graph. We then pass the graph data structure through a linear projection layer, 2 graph convolutional layers (GCN), and finally a linear classification layer to predict an aspect label.

As a secondary training objective, the model predicts additional labels for **Completeness** and **Termination**, which are correlated with the primary aspect label. Together, these labels form a structure inference task, producing additional outputs that are used to calculate loss functions. The previously-defined logic rules 1, 2, 3 are also used to generate the following t-norm rules (Bach et al., 2017) which are then used to calculate additional Hinge Loss functions:

$$\begin{aligned} A \wedge B &= max(A + B - 1, 0) \\ A \vee B &= min(A + B, 1) \\ \neg A &= 1 - A \end{aligned} \tag{4}$$

Since this is a multi-task learning approach, the complete loss function combines the cross-entropy loss for the **Aspect** labels, the binary cross-entropy losses for predicting **Completeness** and **End State** labels, and the t-norm loss for the logic rules:

$$\begin{aligned} TotalLoss = CrossEntropyLoss(Aspect) + \\ BCELoss(Completion) + BCELoss(End) + \\ \lambda * RulesLoss \end{aligned} \tag{5}$$

Where λ denotes the weight attributed to each logic rule. Details of the model implementation can be found in D.

4.5 GraphSpect

To address our second task, we introduce GRAPHSPECT, a GNN which takes SRL graphs processed from natural language sentences. For implementation, we propose a pipeline structure that takes only a sentence as input and returns an aspect label

Category	Method	Accuracy	Macro F1	Weighted Macro F1
Deterministic	AutoAspect*	0.39	0.23	0.40
Prompting	Llama3.1 8B Instruct	0.15	0.12	0.15
	GPT-4o	**0.58**	**0.37**	**0.55**
	DeepSeek V3	0.44	0.17	0.34
Neural	Classifier**	0.35	0.13	0.23
	RuleSpect (w/o Rules)**	0.34	0.21	0.33
	RuleSpect**	**0.36**	**0.23**	**0.34**
	GraphSpect***	0.16	0.08	0.16

Table 3: Performance Metrics by Method: Prompting on extremely large LLMs outperforms AUTOASPECT and neural methods can outperform prompting in certain experiments. *These metrics calculated using dataset from Chen et al. (2021b). **Trained and evaluated using the same 80/20 train-test split with equal distribution of aspect labels. ***Trained and evaluated using a subset of the **data with uneven label distribution, due to the SRL's inability to capture event nominals.

prediction for each event in the sentence, according to the UMR aspect guidelines. We first produce contextualized embeddings from the Llama 3.1 8B model for each token in the input, then match these embeddings with the event and argument identifications for that sentence from the AllenNLP SRL output, in order to create our preprocessed data.

These embeddings with event and argument structure are passed to a two-layer graph attention network. Each of the nodes on the first level corresponds to one of the 30 predefined potential arguments that can be identified by the SRL step, including semantic role arguments (e.g. *agent*, *patient*, *theme*), along with manner arguments which correspond to verbal modifications (e.g. *negation*, *reciprocity*, *causativity*) (Carreras and Màrquez, 2005). These argument nodes are connected by unidirectional edges to the single node in the second layer, which represents the predicated event. Edge weights are learned during training, and any nodes whose arguments are not relevant for a given predicate are masked during both training and prediction. The output from the second layer of this graph network is then passed to a feedforward classifier, from which a softmax function selects the most likely aspect label for the input sentence.

Due to differences in tokenization from the Llama model, the SRL, and the annotators of the data, aligning the SRL data with the embeddings and the gold data proves extremely challenging and inconsistent. This inconsistency degrades learning during training as well as prediction accuracy. Details of the model implementation can be found in D.

5 Results

Table 3 displays the accuracies and F1-scores (both Macro and Weighted Macro) across all methods tested. We report Weighted Macro F1 for our experiments to account for the large imbalance of label distribution. The metrics for AutoAspect were calculated using the dataset from its original paper: Chen et al. (2021b). All other methods used our novel dataset. Each neural method used the exact same 80/20 train-test split, with aspect labels distributed equally across the train and test sets. Detailed results for the performance of Llama-3.1 8B-Instruct by prompt can be found in E.

AutoAspect acted as a deterministic model baseline for our tests, while our Llama-embeddings-based feed-forward classifier acted as a neural model baseline. To this end, prompting on GPT-4o exceeded both baselines on all three evaluation metrics, achieving the best results overall. Among neural models, only our full RuleSpect model improved upon our baseline neural classifier model on all three metrics, though RuleSpect without rules still achieved superior F1 scores over the baseline. None of our neural models achieved superior accuracy to the AutoAspect baseline, though RuleSpect did achieve an equivalent Macro F1 score.

6 Discussion

6.1 AutoAspect

As previously referenced, AutoAspect could not be reimplemented on our novel set of annotated UMR graphs; however, we were still able to get performance metrics for AutoAspect on the dataset with which it was published. Though ideally we would

report AutoAspect's performance on our novel dataset, this serves as our deterministic aspect-classification model benchmark. To this end, our novel methods largely met or exceeded the performance of AutoAspect on its original dataset. Among our methods, prompting GPT-4o exceeded AutoAspect's performance across all three evaluation metrics, while RuleSpect was able to achieve comparable performance. These results demonstrate the efficacy of LLM prompting and neural architectures in making aspect classifications.

6.2 LLM Prompting

Prompt	Accuracy (%)	Correct Predictions
Prompt 1	23.60%	105 / 445
Prompt 2	29.66%	132 / 445
Prompt 3	21.80%	97 / 445

Table 4: Llama-3.1 8B-Instruct Performance by Prompt

The irrelevance of changing prompts to the results obtained suggests that UMR aspect classification is an inherently challenging task for LLMs, likely due to the subtle semantic distinctions required and the significant class imbalance present in the dataset. Table 4 shows the performance of the three prompts for the Llama experiment, which we discuss below:

Prompt 2 The superior performance (29.7% accuracy) is attributed to Prompt 1's greater detail, structured definitions, and explicit rules. Providing clear criteria for distinguishing 'Endeavor' from 'Performance', precise handling of modals ('State'), and an improved definition of 'Activity' likely reduced ambiguity for the LLM. This correlates directly with the improved recall for 'Activity' (0.59 vs 0.09 and 0.15) and reasonable performance on 'State'.

Prompt 1 The less detailed definitions and lack of explicit contrasting rules likely led to more errors compared to Prompt 1. The tendency to default to the 'Process' category (high recall 0.83, low precision 0.03) suggests the model struggled with finer distinctions based on this prompt.

Prompt 3 The drop in accuracy is strongly suspected to be caused by introducing the 'NONE' category, which was absent in the gold standard and likely confused the model. Additionally, the complex conditional rules (negation, hypotheticals), while potentially useful, might have added noise or been difficult to apply correctly without more examples, hindering performance compared to the clearer, simpler core definitions in Prompt 1. The highly conservative prediction for 'Performance' (high precision 0.92, low recall 0.12) might also stem from this prompt's structure.

The dataset's significant imbalance, with the 'state' aspect being dominant and many classes being rare, heavily influenced the results. Performance on minority classes was negligible regardless of the prompt, indicating that prompt engineering alone may not overcome severe data sparsity. Weighted F1 scores were skewed by the majority class, while low macro F1 scores accurately reflected poor performance across most categories.

6.3 LLM Errors and Faulty Reasoning

Predicted Aspect	Llama3.1	GPT-4o	DeepSeek V3	Overall
State	0.157	**0.425**	**0.509**	**0.323**
Habitual	0.015	0.039	0.036	0.027
Process	0.089	0.090	0.082	0.087
Activity	**0.604**	0.046	0.050	0.307
Performance	0.095	0.272	0.290	0.195
Endeavor	0.040	0.083	0.007	0.042
None	0.000	0.041	0.019	0.016

Table 5: Representation of Error Categories by LLM

Sentence: "Good morning," said the little prince.
Predicate: good-02
Selected aspect: State
Gold aspect: None
Justification: The predicate 'good-02' describes a stative property of 'morning,' indicating a quality or condition without any change or process involved. Therefore, it is classified as a state.

Figure 2: Examples of LLM reasoning for selecting the incorrect aspect value. GPT-4o incorrectly identifying a 'none'-aspected predicate as 'state.'

During the analysis of erroneous responses by LLMs to the prompts, several noteworthy trends were observed. For both GPT-4o and DeepSeek, the largest category of errors were misidentifica-

Sentence: They attacked swiftly and fearlessly and coordinated with tacit understanding, but looking back at the Chinese women's soccer team it was in an inferior position either in time-control or coordination.
Predicate: attack-01
Selected aspect: Activity
Gold aspect: Performance
Justification: This sentence is talking about a military attack, and the action of attacking is ongoing and not completed within the time frame of the sentence. Therefore, the aspect value is "activity".

Figure 3: Examples of LLM reasoning for selecting the incorrect aspect value. Llama3.1 incorrectly identifying a 'performance'-aspected predicate as 'activity.'

tions of the aspect value as 'state' (42.5% for GPT-4o, 50.9% for DeepSeek). 'Performance' was the second-most overidentified value by both models (27.2% for GPT-4o, 29.0% for DeepSeek). This result largely aligns with the fact that the data contained a large number of both 'state' and 'performance' aspect labels, incentivizing the models to assign these labels to predicates even when a different label should have been applied. However, errors committed by Llama3.1 8B Instruct were mostly false labelings of 'activity' (60.4%), which had a much smaller share in other models' errors. Table 5 provides percentages of each falsely identified aspect over all errors in labeling for each model.

Most over-identifications of 'state' were for predicates that should have had a 'none' aspect value (88.8% for GPT-4o, 92.6% for DeepSeek). Similarly, a large portion of over-identifications of 'performance' were for predicates labeled 'none' in the gold set (29.9% for GPT-4o, 30.8% for DeepSeek), as well as 'process' (23.4% for GPT-4o, 26.0% for DeepSeek) and 'state' (22.1% for GPT-4o, 16.6% for DeepSeek). In the case of Llama-3.1, most 'activity' labels were for predicates that should have been identified as 'none' (33.9%), 'performance' (33.1%), or 'state' (17.9%).

Figures 2 and 3 depict two examples of LLM reasoning when making some of the most prominent types of errors for the respective model. As shown in Figure 2, GPT-4o had some trouble recognizing differences between a stative event and a non-eventive predicate, such as 'good-02' in this example. Llama3.1, on the other hand, seemed to over-identify events as ongoing or incomplete (Figure 3), perhaps in an attempt to exercise caution in stating definitely that an event has ended or would have ended had it occurred, hence the large number of false 'activity' labels.

6.4 Classifier

Performing simple classification using Llama embeddings resulted in middling performance, coming short in all three evaluation metrics when compared to AutoAspect and most LLM prompting methods. Although LLM embeddings have been seen to capture significant semantic information, these results demonstrate that LLM embeddings alone are insufficient for capturing aspect information.

6.5 RuleSpect

For RuleSpect, we have 2 approaches, one without the rules and the other with the rules. We can see that both were unable to beat the results we get from prompting. There are several reasons which also suggests a lot of scope future improvements:

1. The graph nodes are created from BERT-large embeddings, which are quite small as compared to the Llama embeddings.

2. We should also revisit the annotations, because any mistakes in predicting the completion and result labels would also sway the model in an incorrect direction.

3. The training data is also not uniformly distributed in terms of the aspect labels, we can expect that out of **Performance**, **Endeavor**, and **Activity**, the model would have strong preference to select Performance as it has seen this the most. Perhaps we can improve the rules to account for this.

Comparing RuleSpect with rules and without rules, we can see that the rules do have some minor performance benefits.

6.6 GraphSpect

Overall, GraphSpect, our combined SRL/GNN model, was unable to meet the benchmark evaluation metrics on the test set. GraphSpect falls short

of the AutoAspect benchmark, our embeddings-based classifier benchmark, and all LLM prompting methods (except for Llama) on all three of our evaluation metrics.

One limitation of GraphSpect that contributed to these lower evaluation metrics is that the AllenNLP SRL tool does not identify event nominals, which largely fall under the 'Process' aspect label. Therefore, no examples of 'Process' were able to be correctly identified in the test set, reducing the recall and accuracy metrics. Moreover, the SRL does not identify any of the verbs which have a 'none' aspect label. Although this is correct from an event-identification perspective (because any predicate with a 'none' aspect label in the gold annotation is by definition non-eventive), it also reduces the reported metrics of the model by making it impossible for GraphSpect to report any recall or accuracy for the 'none' class.

Additionally hindering the performance of GraphSpect is the inconsistency in tokenization and event identification between the SRL tool, the Llama tokenizer, and our gold annotators. The SRL tool and the Llama tokenizer produce differences in subword tokenization and punctuation handling which are difficult to account for when formatting inputs for the GNN. Furthermore, the SRL tool and the Llama tokenizer often do not identify the same predicates/events as are present in the Gold Data, resulting in many missed classifications.

Though we hypothesized that Semantic Role Labeling combined with a Graph Neural Network could improve aspect classification, this architecture was unable to capture the complexities of aspectual distinctions within the data. The underwhelming performance of GraphSpect suggests that the use of Semantic Role Labeling did not improve model understanding of predicate aspect. Furthermore, the complex mix of pre-existing tools and embeddings made this implementation especially difficult and hindered overall performance.

7 Conclusion and Future Work

In this paper, we introduce two tasks for annotating UMR aspect labels, the first using existing AMR or UMR graphs and the second using only the surface sentences. For each task, we respectively present RULESPECT, a neurosymbolic GNN that uses logic-rules for training, and GRAPHSPECT, a GNN that accepts SRL graphs parsed from the surface sentence only. As a baseline, we compare our models using an existing rules-based approach, LLM prompting, and a simple feed forward classifier as benchmarks. We find that while neurosymbolic approaches show promise, LLMs are currently the best performing strategy for this task. Regardless, the overall performance of the surveyed methods have room for improvement and are not yet ready to be used for creating UMR annotations.

For future work, we believe that a larger dataset needs to be created for training to become an effective strategy. We suggest that LLM prompting should be used for synthetic data generation, which would sharply reduce the cost of creating training data. Due to the inherent complexity and disagreement that arises from aspect annotation, noisy training data would not necessarily impact the ability for models to learn better representations. Secondly, we believe that the current iterations of our models have a lot of room for improvement. For RULESPECT, a larger LLM can be used for generating embeddings and more logic rules can be derived from the UMR guidelines. For GRAPHSPECT, we can consider alternatives to SRL such as by using dependency parsing, which could provide better alignment with the LLM tokenizer. Finally, we should take into account the impact of corpus distribution in the training data. Style and structure vary greatly between corpora, which could impact our training results.

8 Acknowledgments

We would like to give a special thank you to Julia Bonn for her expert consultation on UMR aspect annotation, helpful guideline learning materials, and information on the relation of VerbNet for rules creation.

We are grateful to Daniel Chen for his prompt response and helpful advice regarding the reimplementation of the AutoAspect classifier.

References

Aryaman Arora, Dan Jurafsky, and Christopher Potts. 2024. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14638–14663, Bangkok, Thailand. Association for Computational Linguistics.

Emmon Bach. 1986. The algebra of events. *Linguistics and philosophy*, pages 5–16.

Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-loss markov random fields and probabilistic soft logic. *Preprint*, arXiv:1505.04406.

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

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Julia Bonn, Harish Tayyar Madabushi, Jena D. Hwang, and Claire Bonial. 2024. Adjudicating LLMs as PropBank adjudicators. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 112–123, Torino, Italia. ELRA and ICCL.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Nancy Chang, Daniel Gildea, and Srini Narayanan. 2022. A dynamic model of aspectual composition. In *Proceedings of the twentieth annual conference of the cognitive science society*, pages 226–231. Routledge.

Daniel Chen, Martha Palmer, and Meagan Vigus. 2021a. 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*.

Daniel Chen, Martha Palmer, and Meagan Vigus. 2021b. 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.

Jayeol Chun and Nianwen Xue. 2024. Uniform meaning representation parsing as a pipelined approach. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 40–52, Bangkok, Thailand. Association for Computational Linguistics.

Bernard Comrie. 1976. Aspect cambridge university press.

William Croft. 2012. *Verbs: Aspect and causal structure*. OUP Oxford.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2017. Convolutional neural networks on graphs with fast localized spectral filtering. *Preprint*, arXiv:1606.09375.

Leon RA Derczynski. 2017. *Automatically ordering events and times in text*. Springer.

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

Lucia Donatelli, Nathan Schneider, William Croft, and Michael Regan. 2019. Tense and aspect semantics for sentential amr. *Society for Computation in Linguistics*, 2(1).

David R Dowty. 1986. The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? *Linguistics and philosophy*, pages 37–61.

Annemarie Friedrich and Alexis Palmer. 2014. Automatic prediction of aspectual class of verbs in context.

Annemarie Friedrich, Alexis Palmer, and Manfred Pinkal. 2016. Situation entity types: automatic classification of clause-level aspect. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1757–1768.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh

Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Erhard Hinrichs. 1986. Temporal anaphora in discourses of english. *Linguistics and philosophy*, pages 63–82.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. 2024. Impossible distillation: from low-quality model to high-quality dataset model for summarization and paraphrasing. *Preprint*, arXiv:2305.16635.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Paul Kingsbury and Martha Palmer. 2003. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *Preprint*, arXiv:1609.02907.

Wolfgang Klein. 2013. *Time in language*. Routledge.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Tim O'Gorman, Martha Palmer, Nathan Schneider, and Madalina Bardocz. 2020. Abstract meaning representation (AMR) annotation release 3.0.

Ronald W Langacker. 2011. Remarks on english aspect. In *Tense-aspect: Between semantics & pragmatics*, pages 265–304. John Benjamins Publishing Company.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. 2018. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Transactions of the Association for Computational Linguistics*, 6:343–356.

Charles N Li and Sandra A Thompson. 1989. *Mandarin Chinese: A functional reference grammar*. Univ of California Press.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. *ArXiv*, abs/1909.00126.

Tao Li and Vivek Srikumar. 2020. Augmenting neural networks with first-order logic. *Preprint*, arXiv:1906.06298.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Betsy Hicks McDonald. 1982. *Aspects of the American Sign Language predicate system*. State University of New York at Buffalo.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. Caters: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd workshop on computing news storylines (CNS 2016)*, pages 47–56.

Barbara H Partee. 2011. Nominal and temporal semantic structure. In *Prague Linguistic Circle Papers: Travaux du cercle linguistique de Prague nouvelle série. Volume 3*, pages 91–108. John Benjamins Publishing Company.

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'gorman, James Gung, Kristin Wrightbettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.

James Pustejovsky, Harry Bunt, and Annie Zaenen. 2017. Designing annotation schemes: From theory to model. *Handbook of Linguistic Annotation*, pages 21–72.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Hans Reichenbach. 1947. Elements of symbolic logic.

M. Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Haibo Sun, Nianwen Xue, Jin Zhao, Liulu Yue, Yao Sun, Keer Xu, and Jiawei Wu. 2024. Chinese UMR annotation: Can LLMs help? In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 131–139, Torino, Italia. ELRA and ICCL.

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.

Herman Van Olphen. 1975. Aspect, tense, and mood in the hindi verb. *Indo-Iranian Journal*, 16(4):284–301.

Zeno Vendler. 1957. Verbs and times. *The philosophical review*, 66(2):143–160.

Kunze Wang, Soyeon Caren Han, and Josiah Poon. 2022. Induct-gcn: Inductive graph convolutional networks for text classification. *Preprint*, arXiv:2206.00265.

A Aspect Label Distribution from Existing UMR Data

B LLM Prompts

Listing 1: Prompt template used in Experiment 2.

```
You are an expert annotator of aspect
    under the Uniform Meaning
    Representation (UMR) framework.
    Aspect is a single variable in a UMR
     graph that needs to be annotated
    for every predicate in a given
    sentence. For this task, it can take
     on six different values:

1) State: no change takes place over the
     course of the event. This includes
    stative verbs such as "be," "want,"
    "love," but also ability modals ("
    can"/"could"/"be able to"),
    perception verbs ("see," "feel," "
    hear"), and other verbs describing
    conditions rather than an active
    event.

2) Habitual: this label is for events
    that happen repeatedly. In English,
    this usually correlates with the
    present simple tense, but also with
    "used to" or "would" when the
    habitual event is in the past tense.

3) Process: a superset of Activity,
    Endeavor, and Performance. The
    Process tag is reserved for ongoing
    events with uncertain or unspecified
     beginning or endedness. It is
    typical for events expressed as
    nouns, such as "wrongdoing" or "
    creation," unless context is
    provided on whether the event has
    ended or the span of the event.

4) Activity: a type of process that
    clearly does not start or end within
     the time window of the predicate.
    This includes both cases where the
    processive event is ongoing, e.g.,
    in "He is still writing his paper,"
    but also cases where there is no
    evidence that the event has ended,
    as in "He was writing his paper
    yesterday" or "He started playing
    the violin."

5) Endeavor: a type of process that ends
     within the time window of the
    predicate but does not reach a
    particular end state. Some evidence
    for the "Endeavor" aspect label: 1)
    terminative aspectual marking, e.g.,
     "stop" as in "Mary stopped mowing
    the lawn" 2) durative adverbials, e.
    g., "for a long time," "all summer"
    3) non-result path, e.g., "through
    the valley," "along the beach" (
    motion or attempt to change without
    a clear result or destination)

6) Performance: a type of process that
    reaches a result state. It covers
    achievements (instantaneous binary
    change) and accomplishments (process
     before and up to the moment the
    change happens). Some evidence for
    the "Performance" aspect label as
    opposed to "Endeavor": 1) completive
     aspectual marker, e.g., "finished"
    as opposed to "stopped" 2) container
     adverbial, e.g., "in thirty minutes
    " as opposed to "for thirty minutes
    ."

Your task is to determine the aspect
    value for a specific predicate in a
```

Aspect	Little Prince	Minecraft	BOLT DF	WB	Pear Story	Lorelei	UMR 1.0	Total
State	63	20	119	45	121	0	62	430
Habitual	1	0	17	4	28	0	2	52
Process	2	0	5	5	44	1	1	58
Activity	9	0	31	14	57	0	21	132
Performance	35	2	43	18	159	3	57	317
Endeavor	0	0	0	0	2	0	14	16
Total	**110**	**22**	**215**	**86**	**411**	**4**	**157**	**1005**

Table 6: Aspect Label Distribution from Existing UMR Data

```
    UMR graph for a given sentence.

Sentence: "{sentence}"

The predicate you need to analyze is: "{
    predicate}" with variable name: "{
    variable_name}"

You need to determine the aspect value
    for this predicate. The aspect value
     must be one of the following 6
    values:
1) state
2) habitual
3) process
4) activity
5) endeavor
6) performance

The aspect value of the predicate "{
    predicate}" under the variable {
    variable_name} is:
```

Listing 2: Prompt template used in Experiment 2.

```
Aspect is a grammatical category that
    expresses how a verbal action, event
    , or state extends over time, it is
    a single value that is annotated for
     every event. For this task, aspect
    can take on six different values:

1) State: No change takes place over the
     course of the event. This includes
    stative verbs ("be," "want," "love")
    , perception verbs ("see," "feel," "
    hear"), all modal verbs, and events
    under the scope of ability modals.
    Examples of modal verbs include
    those expressing:

Ability: "can," "could," "be able to" (e
    .g., "She can speak five languages")
Possibility: "may," "might" (e.g., "It
    may rain later")
Necessity: "must," "has to" (e.g., "You
    must wear a seatbelt")
Permission: "may," "can" (e.g., "You may
     leave early")
Obligation: "must," "should" (e.g., "You
     should apologize")
Deduction: "must be," "can't be" (e.g.,
    "She must be the new teacher")

All ability modals ("can," "could," "be
    able to") are always annotated as
    state aspect, regardless of the
    action they modify.


2) Habitual: this label is for events
    that happen repeatedly. In English,
    this usually correlates with the
    present simple tense, but also with
    "used to" or "would" when the
    habitual event is in the past tense.

3) Process: a superset of Activity,
    Endeavor, and Performance. The
    Process tag is reserved for ongoing
    events with uncertain or unspecified
     beginning or endedness. It is
    typical for events expressed as
    nouns, such as "wrongdoing" or "
    creation," unless context is
    provided on whether the event has
    ended or the span of the event.

4) Activity: A type of process that
    doesn't start or end within the
    predicate's time window. This
    includes ongoing processes ("He is
    still writing his paper") and cases
    with no evidence of ending ("He was
    writing his paper yesterday").
    Events in present tense are
    typically activities. Activities can
     be further classified as:

i) Directed activity: Change occurs
    gradually along a qualitative scale
    (e.g., "The soup was cooling")
ii) Undirected activity: Change doesn't
    progress incrementally (e.g., "The
    cat was meowing")

The activity label applies when there's
    no evidence the event has ended,
    whether clearly ongoing or ambiguous
     about continuation, often depending
     on context and real-world knowledge
    .

5) Endeavor: a type of process that ends
```

```
     within the time window of the
    predicate but does not reach a
    particular end state. Some evidence
    for the "Endeavor" aspect label: 1)
    terminative aspectual marking, e.g.,
     "stop" as in "Mary stopped mowing
    the lawn" 2) durative adverbials, e.
    g., "for a long time," "all summer"
    3) non-result path, e.g., "through
    the valley," "along the beach" (
    motion or attempt to change without
    a clear result or destination)

6) Performance: a type of process that
    reaches a result state. It covers
    achievements (instantaneous binary
    change) and accomplishments (process
     before and up to the moment the
    change happens). Some evidence for
    the "Performance" aspect label as
    opposed to "Endeavor": 1) completive
     aspectual marker, e.g., "finished"
    as opposed to "stopped" 2) container
     adverbial, e.g., "in thirty minutes
    " as opposed to "for thirty minutes
    ."

The markers that indicate "endeavor"
    versus "performance" aspects:

Durative adverbials (indicating duration
     like "for a long time," "all summer
    ") suggest Endeavor
Container adverbials (time periods that
    contain events like "in the morning
    ," "during class") suggest
    Performance
Terminative aspectual markers ("stop," "
    cease," "quit") suggest Endeavor
Non-result paths (motion without
    reaching endpoints like "through the
     valley," "along the beach") suggest
     Endeavor
Completive aspectual markers ("finished
    ," "had eaten," "ran in under four
    hours") suggest Performance

Your task is to determine the aspect
    value for a specific predicate in a
    given sentence.

Sentence: "{sentence}"

The predicate you need to analyze is: "{
    predicate}" with variable name: "{
    variable_name}"

You need to determine the aspect value
    for this predicate in that sentence.
     The aspect value must be one of the
     following 6 values:
1) state
2) habitual
3) process
4) activity
5) endeavor
6) performance

The aspect value of the predicate "{
    predicate}" under the variable {
    variable_name} is:
```

Listing 3: Prompt template used in Experiment 2.

```
Your task is to determine the aspect
    value for a specific predicate in a
    given sentence, based on UMR aspect
    annotation guidelines. Aspect
    expresses how a verbal action, event
    , or state extends over time.

Analyze the predicate within the context
     of the sentence and assign one of
    the following seven aspect values:

1)  State: No change takes place.
    Includes stative verbs (be, want,
    love), perception verbs (see, feel),
     modal verbs (can, must, should, may
    , etc.), and inactive actions (sit,
    lie, think). Rule: Ability modals (
    can, could, be able to) are always '
    State'.
2)  Habitual: Event happens repeatedly
    or usually. Often indicated by
    present simple tense, or phrases
    like "used to", "often", "every year
    ".
3)  Process: An ongoing event where the
    beginning or end is uncertain or
    unspecified. Rule: Often the default
     for events expressed as nouns (
    nominals like "wrongdoing", "
    creation") unless context suggests
    otherwise.
4)  Activity: A process that clearly
    does not start or end within the
    predicate's time window. Includes
    ongoing processes (e.g., progressive
     tense "is writing", "was writing")
    or cases with no evidence of ending.
5)  Endeavor: A process that ends within
     the time window but does not reach
    a specific result state. Often
    indicated by terminative markers ("
    stop"), durative adverbials ("for
    two hours"), or non-result paths ("
    walk along the beach").
6)  Performance: A process that ends and
     reaches a result state (an endpoint
     or change). Covers achievements (
    instantaneous change, "shatter", "
    arrive") and accomplishments (
    durative process leading to change,
    "build a house", "write a book").
    Indicated by completive markers ("
    finished") or container adverbials
    ("in thirty minutes").
7)  NONE: Use this if the predicate is
    not eventive (e.g., adjectives,
    structural elements, some epistemic
    modals) or doesn't have a clear
    aspect.

Additional Rules & Guidelines:
```

```
 Prioritize: Aim for more specific
     distinctions like 'Activity', '
     Endeavor', or 'Performance' before
     defaulting to the general 'Process'
      category.
 Negation: If an event is negated (e.g.,
      "did not go"), analyze the aspect
     as if the event did happen.
 Hypotheticals: If the event is
     hypothetical ("If he went..."),
     analyze the aspect by pretending
     the event is real.
 Commands: For commands ("Go!", "Be
     happy!"), use the canonical
     interpretation of the event's
     aspect ('Performance' for "Go", '
     State' for "Be happy").

Input:

Sentence: "{sentence}"
Predicate to analyze: "{predicate}" (
    Variable: "{variable_name}")

Task:

Determine the single aspect value for
    the predicate "{predicate}" ({
    variable_name}) in the given
    sentence.

Output:

Provide only the aspect value word (
    state, habitual, process, activity,
    endeavor, performance, or NONE).

Aspect:
```

C LLM Prompting Costs

The cost to run the experiment of all three prompts on the GPT-4o model was $15.56. The DeepSeek V3 model cost $1.20 for the same set of experiments. The prompting experiments using the Llama-3.1 8B Instruct model required 49 minutes of compute runtime, and it was accessed using the Transformers library through Hugging Face.

D Neural Model Details and Hyperparameters

The Feed Forward Classifier uses 4 feed forward layers of dimension (4096×1024), (1024×512), (512×256), and (256×7) respectively. We used a learning rate of 0.01 and a batch size of 16 to train for 1000 epochs, although peak metrics were achieved within the first 10 epochs. Running CPU, the model took roughly 20 minutes to train.

For RULESPECT, the optimal weights for the loss function were $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, meaning that an equal weighting was given to the binary cross-entropy loss for the **Completeness** and **End State** label predictions as well as to the hinge loss for the logic rules. The implementation for RULESPECT without rules had both lambdas set to 0. In either case, we used a learning rate of 0.001 and train for 1000 epochs on a Google Colab GPU, which took on average 84 minutes per implementation.

For GRAPHSPECT, we trained using a learning rate of 0.001 for 300 epochs on a local GPU, which took roughly an hour.

E Classification Metrics Summary

F AutoAspect Dataset Statistics

Table 8 reports the performance metrics for the two main tasks of AutoAspect, namely the correct identification of events (both predicative events and event nominals), and the aspect label prediction for all identified events. The authors note that recall is the only representative metric for the event identification task, because precision (and thus F1) would penalize the classifier's identification of events which were not identified in the gold data; the authors sought to avoid this because of the ambiguity in the annotation guidelines and the inconsistency of event identification even by human annotators.

G Detailed Metrics by Aspect for All Experiments

	Prompt 1			Prompt 2			Prompt 3		
Aspect	P	R	F1	P	R	F1	P	R	F1
State	0.85	0.32	0.47	0.77	0.34	0.47	0.81	0.28	0.41
Activity	0.21	0.09	0.13	0.14	0.59	0.23	0.26	0.15	0.19
Performance	0.83	0.10	0.18	0.35	0.16	0.22	0.92	0.12	0.22
Process	0.03	0.83	0.06	0.00	0.00	0.00	0.03	0.75	0.06
Macro Avg	0.19	0.14	0.08	0.14	0.11	0.10	0.20	0.13	0.09
Weighted Avg	0.68	0.24	0.32	0.53	0.30	0.34	0.68	0.22	0.30
Accuracy	0.2360			**0.2966**			0.2180		

Table 7: Detailed Classification Metrics Summary for Llama. Performance on minority classes (e.g., 'directed-achievement', 'generic', 'habitual') was negligible (often 0.00 F1-score) across all experiments and is omitted from the table for brevity.

Task	Recall	Accuracy
Event Identification	76.17	n/a
Event Aspect Prediction	n/a	62.57

Table 8: AutoAspect Event Identification and Aspect Prediction Performance

	Prompting			Neural			
Aspect	Llama3.1 8B Instruct	GPT-4o	DeepSeek V3	Classifier	RuleSpect (w/o Rules)	RuleSpect	GraphSpect
State	0.44	**0.77**	0.68	**0.95**	0.14	0.11	0.14
Habitual	0.09	0.21	**0.96**	0.00	**0.38**	**0.38**	0.00
Process	0.02	0.11	**0.90**	0.00	**0.22**	0.12	0.00
Activity	0.60	0.23	**0.96**	0.00	0.00	0.00	0.00
Performance	0.09	**0.84**	0.79	0.09	**0.41**	0.39	0.23
Endeavor	0.06	0.17	**0.98**	0.00	0.05	0.05	0.00
None	0.00	0.24	**0.66**	0.00	**0.42**	**0.42**	0.17
Overall	0.15	**0.58**	0.44	0.35	**0.36**	0.34	0.16

Table 9: Accuracy of Prompting and Neural Methods by Aspect Label