Chinese UMR annotation: Can LLMs help?

Haibo Sun, Nianwen Xue, Jin Zhao
Liulu Yue, Keer Xu, Yao Sun, Jiawei Wu
Brandeis University
{hsun, xuen, jinzhao, liuluyue, keerxu, yaosun, jiaweiwu}@brandeis.edu

Abstract

We explore using LLMs, GPT-4 specifically, to generate draft sentence-level Chinese Uniform Meaning Representations (UMRs) that human annotators can revise to speed up the UMR annotation process. In this study, we use few-shot learning and **Think-Aloud prompting** to guide GPT-4 to generate UMR sentence-level graphs. Our experimental results show that compared with annotating UMRs from scratch, using LLMs as a preprocessing step reduces the annotation time by two thirds on average. This indicates that there is great potential to integrate LLMs into the pipeline for complicated semantic annotation tasks.

Keywords: Uniform Meaning Representation, Large Language Models, Semantic Annotation

1. Introduction

Uniform Meaning Representation (UMR) (Gysel et al., 2021; Bonn et al., 2023) is a graph-based cross-lingual semantic representation that includes a sentence-level representation and a document-level representation. The sentence-level representation is based on Abstract Meaning Representation (AMR) (Banarescu et al., 2013) but has been extended to capture not only predicate-argument structures, word senses, and named entities as AMR does, but also aspectuality of events, person and number attributes of entities, and quantification. Its document-level annotation includes temporal and modal dependencies for events, as well as coreference relations for entities and relations. Such a comprehensive meaning representation is very demanding for human annotators in terms of the linguistic training they needed, as they have to internalize a large inventory of semantic concepts, relations, and attributes, and is very time-consuming to annotate.

One way to speed up the annotation process is to pre-parse the text into "draft" UMRs and have human annotators correct them. However, the parser needs a considerable amount of UMR-annotated data to train, and no large UMR training set exists yet. In UMR release 1.0 (Bonn et al., 2024), each language has fewer than a thousand sentences of annotated UMRs, and it is insufficient to train a parsing model with adequate performance. In this paper, we explore the use of Large Language Models (LLMs) to generate Chinese UMRs that human annotators can correct for the purpose of speeding up the annotation process. We investigated the question of whether using LLMs as a preprocessing step would reduce the amount of time required for human annotators to annotate the same amount of data compared to annotating UMRs from scatch. The answer to this question is determined by several factors. The most important factor is the quality of the UMRs generated by LLMs. If the UMRs generated by LLMs are of poor quality, the human annotator will need to spend so much time deconstructing the structure generated by the LLMs that they are better off starting from scratch. The second factor is the functionalities of the annotation tool used for UMR annotation. If the tool has functionalities that allow the copying of subgraphs of LLM-generated UMRs when constructing the correct UMR, this will lower the threshold of parsing accuracy needed for LLMs to have a positive impact. In our annotation experiments, we use UMR-Writer (Zhao et al., 2021; Ge et al., 2023), and this tool allows subgraphs to be copied and reused. Therefore, the primary factor will be the quality of the UMRs generated by LLMs.

Our experimental results show that using LLMs as a pre-processing step on average reduces the annotation time by about two thirds. The annotators reported that LLM-generated graphs often contain correct top-level structures and subgraphs that save annotator time annotating UMRs. An evaluation of LLM-generated parses shows that their qualities are slightly below that of initial human annotation, but not by far.

The rest of the paper is organized as follows. In Section 2, we describe Uniform Meaning Representation for Chinese to provide a concrete idea of how challenging it is to annotate Chinese UMRs. In Section 3, we introduce our approach to using LLMs to generate the draft graphs and detail several key challenges in constructing UMR graphs. In Section 4, we evaluate LLM-generated parses with respect to their well-formedness and overall evaluation scores against gold UMR graphs. We measure inter-annotator agreement (IAA) between

5th Workshop on Designing Meaning Representations (DMR 2024) @LREC-COLING-2024, pages 131–139

human annotators and the time savings from annotating LLM-generated UMRs compared with UMR annotation from scratch, and we also summarize the feedback from human annotators that reveal the strengths and weaknesses of LLM-generated UMRs as the starting point for human annotation. Related work is discussed in Section 5 and we conclude in Section 6.

2. Chinese UMRs

In this section we briefly illustrate different aspects of UMR annotation with an example in (1). UMR is a representation for entire documents, not just individual sentences, so we show the UMR in Figure 1 for a text snippet of two sentences that forms a minimal document. Solid lines are labeled with semantic relations at the sentence level that include semantic roles and other semantic relations, as well as attributes, while the dotted lines represent relations at the document level.

(1) a. 新时代 集团于1995 年计划 将 城市
New Era Inc. in 1995 plan BA City
电视 售与罗渣士 通讯
Television sell Rogers Communications
集团，以 集中 发展
Inc. , in order to focus on develop
新时代 电视 。
New Era Television .

"The New Era Inc. planned in 1995 to sell City Television to Rogers Communications Inc. in order to focus on the development of New Era Television."

b. 罗渣士 当时 计划 将 之从 有线
Rogers at that time plan BA it from cable
电视台 转型为 地面
TV station transform into terrestrial
广播 频道 ，并 加入十一 种
broadcast channel , and add eleven
语言 的 电视节目 。
language DE TV program .

"At that time, Rogers planned to transform it from a cable TV station into a terrestrial broadcast channel and add TV programs in eleven languages."

Sentence-level representation The sentence-level representation includes word senses and predicate argument structures, named entity types, aspectual attributes of events, person and number attributes of entities. In Figure 1, 计划-01 in the first sentence represents the first sense of 计划 ("plan"), and it is a predicate that has two core arguments, *Arg0* which is a the company 新 时代 集团 ("New Era Group"), and *Arg1* 售-01, which has its own argument structure. It also has a non-core argument 发展-05 ("develop") that serves as its purpose (*:purpose*, and a *date-entity* that serves as its temporal modifier (*:temporal*). In addition to arguments, since 计划-01 is an event, it also has an aspectual attribute that indicates it is a *State*. The semantic relations between the predicate and its arguments and attributes are represented as directed edges from the predicate to the argument or attribute.

In addition to predicate-argument structures, UMR, following AMR, also represents named entity types. The named entity type is represented as a concept that has a list of strings that represent the actual name. For example, in Figure 1, 新 ~ 时代 ~ 集团 ~ ("New Era Group") is a name of the type *company*. Pronouns are typically represented as a concept with *person* and *number* attributes. For instance, 之 is a pronoun that maps to a *thing* concept with person attribute (*ref-person*) that has the value of *3rd*, and a number attribute (*ref-number*) that has the value of *Singular*.

Document-level representation Some semantic relations go beyond sentence boundaries, and these are represented as directed edges between a parent and a child, which can be (but not necessarily) in a different sentence. For example, the *thing* concept that derives from the pronoun 之 is coreferent with the *company* concept that refers to 罗渣士 集团 ("Rogers Inc."), and this is represented with the *same-entity* relation.

Temporal relations hold among events, between events and time expressions, and among time expressions. They are also represented as relations among concepts which can go beyond sentence boundaries or within the same sentence. As an example of temporal relations that go beyond sentence boundaries, the two instances of 计划-01 overlap with each other in terms of their temporal duration, just as the concepts 当时 ~ in the second sentence overlap with the date-entity with the year 1995, as they refer to the same time period. As an example of temporal relations within the same sentence, 计划-01 ("plan") is *before* 售-01 ("sell") and 售-01 ("sell") is *before* 发展-05 ("develop").

Modal dependencies are relations between a *conceiver* or *source* and an event that indicate the level of certainty that the conceiver holds with respect to the event. In most cases the conceiver of an event is the author (AUTH), but it can also be other sources as well if the author cites a different source for the event.

It is very time-consuming for the annotator to annotate such a rich representation as UMR. We are interested in whether LLMs can be used to generate "draft" UMR graphs from raw text that annotators can correct to speed up the annota-

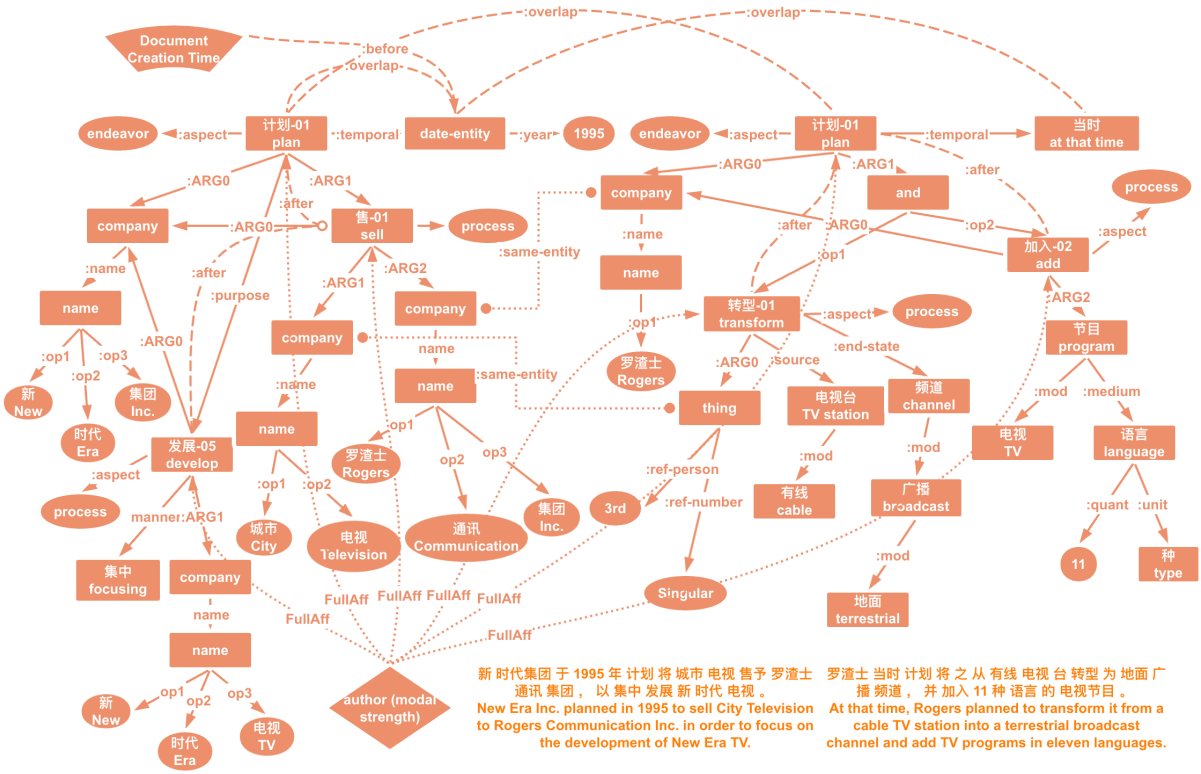

Figure 1: An example of a UMR graph for a mini-document of two sentences.

tion process. To make our study feasible, we conducted only experiments to generate sentence-level UMRs with LLMs.

3. Pre-parsing with LLMs

Prompt design is the key to the quality of LLM-generated UMRs. We explore three different prompting methods to observe their effect on the UMR parsing quality. We conduct our experiments in three settings: zero-shot, few-shot, and Think-Aloud. In the zero-shot setting, LLMs are not given any annotated examples, while in the few-shot setting, they are given a short document of 8 UMR-annotated examples. Finally, in the Think Aloud setting, in addition to the 8 examples, they are also given a step-by-step instruction of how the UMRs are annotated. We use GPT4 in all experiments.

3.1. Zero-shot setting

In the zero-shot setting, we give GPT4 the following prompt without any examples. Since UMR 1.0 was released after GPT-4 [1]was trained, we try to guide it to learn from AMR. However GPT-4 failed to generate any well-formed UMRs so we will not discuss it further.

> You are an expert linguistic annotator. You need to parse a given sentence into Uniform Meaning Representation, which is similar to Abstract Meaning Representation, but you need to name each variable starting with “s”, followed by the number of sentence. All the tokens should only be from the sentence, and you must not hallucinate about any tokens or miss any tokens.

3.2. Few-shot setting

In the few-shot setting, we give GPT-4 the following instruction followed by UMRs of 8 sentences. When selecting the example UMRs, our aim is to have a good coverage of aspectuality attributes and modal strengths [2] that are new in UMR as they are absent in AMR, which has been around for longer periods of time and is therefore more accessible to LLMs. The instructions given to GPT-4 is as follows:

> You are a linguistic annotator. You need to follow the examples to parse a sentence into Uniform Meaning Representation step by step. You must name each variable starting with “s”,

[1]The version we use is gpt-4-0125-preview.

[2]Modal strength is represented at the document level in UMR, but in most cases the conceiver or source is the author and can thus be annotated as a shorthand at the sentence level.

followed by the number of the sentence. All the tokens should only be from the sentence, and you must not hallucinate any tokens. You should identify the main verb as the head of the graph, and analyze the clauses recursively. You will also need to add "modal strength" to any predicates in the format ":modstr" with six possible values: [FullAff, PrtAff, NeutAff, FullNeg, PrtNeg, NeutNeg], and also add an aspect to any predicate in the form of ":aspect" with six possible values [Process, Endeavor, Performance, Activity, Habitual, State], and you will be shown how to use these values later. NEVER combine any tokens separated by space!

In the few-shot setting, no explanation is given to GPT-4, but we attempt to include examples of how common UMR concepts, attributes, and relations are represented. The following are UMR snippets that illustrate the representation of modal strength, aspectuality, and named entities and their relations.

Modal strength In the sentence 4, the main predicate is 讲 ("tell"). It is in a imperative mode, and under the modal verb 不能 ("cannot"), which makes its modal strength NeutNeg, meaning neutral negative.

(2) 这个关于 他 晋升 的 秘密 不能 给
this about he promote DE secret cannot to
任何人 讲 !
any person tell !

"You cannot tell anybody the secret that he got promoted!"[3]

```
(s1x / 讲-01["tell"]
      :mode imperative
      :modstr NeutNeg
      ...)
```

Aspectuality An example of aspectuality represented in the UMR is also provided in 3:

(3) ... 临近 演唱会 尾声
... approaching concert end

"... near the end of the concert"

```
... (s2x2 / 临近-01["approaching"]
            :ARG0 (s2x3 / 演唱会 ["concert"])
            :ARG1 (s2x4 / 尾声 ["end"])
            :aspect State
            :modstr FullAff
            ...)
```

[3]The glossing abbreviations used in this paper are: DE: possessive or genitive marker

Named entities in appositive constructions We also provided GPT-4 some common patterns in UMR annotation, such as appositive constructions that involve a named entity of type *individual-person* that has a particular type of position in some organization, which is often also a named entity:

(4) 美国前 总统 克林顿
US former president Clinton

```
(s41i2 / individual-person
     :name(s41n / name
           :op1 " 克林顿"["Clinton"])
     :ARG1-of (s41h / have-org-role-91
          :ARG2 (s41c / country
                :name (s41n2 / name
                      :op1 " 美国"["US"]))
          :ARG3(s41x3 / 总统 ["president"]
                :mod (s41x4 / 前 ["former"])))))
```

The UMR inherits some of the named entity types from AMR but also adds quite a few new ones that reflect the different types of named entity in different cultures. This type of structure is very common in the data and if they can be correctly generated by LLMs, it would be a big help for human annotators who post-edit these UMRs.

3.3. Think-Aloud Prompting

Inspired by the research on using Chain-of-thought (Wei et al., 2022) when prompting LLMs to perform natural language tasks, we experimented with asking GPT-4 to follow the Think-Aloud Protocol (Jääskeläinen, 2010) to instruct it to explicitly output the thought process before outputting the final answer. This technique is originally an experiment protocol in psychology and ergonomics with human subjects. It asked human subjects to "think-aloud" - voicing their thoughts explicitly when or after completing a given task - to make researchers aware of how and why they take particular actions (Oh and Wildemuth, 2009; Wolcott and Lobczowski, 2021). Because the reasoning process is expressed in natural language and human annotation is guided by latent thoughts, we hypothesize that the reasoning process can serve as an intermediary from texts to semantic graphs. We designed the though process to particularly address the following issues.

Mismatch between the linear order of the sentence and the hierarchical structure of UMR graphs While a sentence is presented linearly from left to right, its UMR graph is a hierarchical structure with no meaning attached to the linear order. This mismatch may worsen the tendency of LLMs to hallucinate or ignore certain parts of

the input sentence. We therefore instruct GPT-4 to "think" hierarchically by selecting the main predicate first and then attending to its arguments. If the argument itself is another predicate, then GPT-4 is asked to recursively identify its arguments in a depth-first manner similar to the annotating process of human annotators. This is illustrated in (5).

(5) **Prompt**:Parse this sentence: 该周报也曾经短暂的拥有过一个属自己的电视节目。
Prompt translation: The weekly report also once briefly had its own television program.
Exemple response: 这是第 3 句话。
我们需要先判断这句话的句式，然后分析这句话中每个词语之间的依存句法关系。
该句子是一个陈述句，top node 是“拥有”
“拥有”的主语是“周报”，“周报”的修饰是“该”，因此“拥有”:arg0“周报”(s3x2)
“拥有”的宾语是“节目”，而“节目”的定语是“电视”，而这里“电视”是“节目”的媒介，所以“媒介”:medium“电视”
“节目”属于“自己”，而“自己”在这里是一个 pronoun，指代的是“周报”，因此“属”:arg1 s3x2
……
Response translation: This is the third sentence. We need to first determine the sentence structure of this sentence and then analyze the dependency syntax relationships between each word in the sentence.
The sentence is a declarative sentence, with 'owning' as the top node.
The subject of 'owning' is 'weekly report,' and 'the' modifies 'weekly report,' hence 'owning': arg0 'weekly report' (s3x2).
The object of 'owning' is 'program,' and the attribute of 'program' is 'television,' where 'television' serves as the medium of 'program,' so 'medium': medium 'television.'
'Program' belongs to 'oneself,' and 'oneself' here is a pronoun, referring to 'weekly report,' thus 'belong': arg1 s3x2
...

Identifying the abstract concepts for discourse relations In UMR, discourse relations are represented as a close set of abstract concepts that form a lattice so that different languages can select discourse relations of different levels of granularity for that language. Discourse relations are represented as abstract concepts because they are often implicit and can only be inferred from the larger context of the discourse. As such it is easy for LLMs to miss them. We attempt to guide LLMs to detect them with Think-Aloud prompting. This is illustrated in (6).

(6) **Prompt**: 目前《世界新闻周报》的纸本已经停刊，不过仍继续发行网络版。
Translation: Currently, the print edition of "World News Weekly" has been discontinued, but the online version is still being published.
Exemple response: 该句子是一个转折句，因此 top node 是 but-91
前半句的谓语是“停刊”，因此 but-91 :arg1 停刊
... ...
后半句的谓语是“继续”，因此 but-91 :arg2 继续
Response translation: The sentence is a contrastive sentence, therefore the top node is but-91.
The predicate of the first half of the sentence is "discontinued", thus but-91 :arg1 is "discontinued".
The predicate of the second half of the sentence is "continues", thus but-91 :arg2 is "continues".

Identifying named entities and their types. The named entity types of UMR also form a hierarchy, and they extend the AMR named entity types to cover a wide range of named entities from different cultures of the world. Some of them may have never been seen by GPT-4, and we instruct it to pay particular attention to them, as illustrated in (7).

(7) **Prompt**: 《世界新闻周报》是一份已经停止发行的美国超市小报。
Translation: The World News Weekly is an American supermarket tabloid that has ceased publication.
Example response: 判断句的主语，是“《世界新闻周报》”，是【命名实体】，是一种 newspaper，因此是 named entity，类型是 newspaper，首字母是 n，因此变量名是 s1n，因此 s1i2 :ARG1 newspaper :name 《世界新闻周报》。
Response translation: The subject of the judgment sentence is 'The World News Weekly', which is a named entity, a type of newspaper. Therefore, it is a named entity, with the type being newspaper, and its first letter is 'n', thus the variable name is 's1n'. Therefore, 's1i2 :ARG1 newspaper :name The World News Weekly'.

Identifying predicates and annotating their aspects and modal strength As aspect and modal strength are only relevant to predicates, to annotate the aspect and model strength values correctly, the first step is to identify the predicates. Aspect annotation is difficult for human annotations due to the lack of explicit aspect markers for most predicate instances, it is even difficult for human annotators. The modal strength value also has different manifestations in the Chinese language, and they can be derived from modal verbs, certain adverbs, or quoted speech. So we designed instructions to guide GPT-4 to pay attention to the right places, as illustrated in (8).

(8) **Prompt**:……阿扎扎称：……，结果竟在拿破仑头骨中发现了一枚无法解释的神秘芯片。
Prompt translation:...Azaza said: ..., surprisingly, a mysterious chip that cannot be explained was found in Napoleon's skull.
Exemple response:【“解释”是一个谓词，它的语法体标记 (:aspect) 只能从 state, performance, activity, habitual, endeavor, process 中选择，它

的语气强度 (modality strength)(:modstr) 只能从 FullAff, PrtAff, NeutAff, FullNeg, PrtNeg, NeutNeg 中选择】。“解释”是一个动作但不一定有结果和开始，因此“解释” :aspect Process；由于解释有一个“无法”作为修饰，表达的是否定意义，而“解释”来自于说话人的内容，无法确定其真实性，只能作推断，因此“解释” :modstr PrtNeg；同时，由于“解释”来自于说话人的内容，需要引用到上一个谓词，因此“解释” :QUOT “称”

Response translation: The verb 'explain' has its grammatical aspect marker (:aspect) that can only be chosen from state, performance, activity, habitual, endeavor, process, and its modality strength (:modstr) can only be chosen from FullAff, PrtAff, NeutAff, FullNeg, PrtNeg, NeutNeg. 'Explain' is an action that may not necessarily have a result or even start, therefore 'explain': aspect Process; since 'explain' is modified by 'unable to', expressing a negative meaning, and 'explain' comes from the speaker's content, its truth cannot be determined, only inferred, therefore 'explain' :modstr PrtNeg; meanwhile, since 'explain' comes from the speaker's content, it needs to refer to the previous predicate, therefore 'explain' :QUOT 'said'.

4. Experiments

We conducted experiments to answer three questions: (i) How does GPT-4 perform in generating UMRs in a few shot and Think-Aloud settings? (ii) How is GPT-4 faring in comparison with human annotators? (iii) Does it take less time for human annotators to correct GPT-generated UMRs than annotating from scratch? We answer these questions through quantitative evaluations and also through qualitative analysis.

4.1. Experiment setup

We selected two articles published in the latter half of 2023 to conduct experiments on to make sure these articles were not included as part of the training data for GPT-4. The articles were chosen from authoritative news agencies to guarantee its grammaticality and factuality. Both articles have 26 sentences so that they can be finished in a reasonable amount of time.

The human annotation experiments are performed by four annotators. These annotators do not have extensive linguistic backgrounds but have taken linguistic courses. In order to have fair comparison of annotation speed under the two conditions, annotating from scratch vs annotating from GPT-generated UMRs, we need to make sure that the same annotator does not annotate the same article twice. We divide the four annotators into two groups, with two annotators in each group. We first have each group annotate one of the two articles from scratch, and then switch to annotate the other article from GPT-generated UMRs. After they finished annotating the articles from scratch, each group met to discuss their differences and arrived at a consensus annotation that we designate as the gold annotation.

4.2. Quality of GPT-generated UMRs

We used GPT-4 to generate UMRs for the two articles in few-shot and Think-Aloud settings, each with temperatures of 0 and 0.7. We thus have four UMR graphs generated under four conditions: few-shot at temperatures of 0 (0F) and 0.7 (7F) , Think-Aloud at 0 (0T) and 0.7 (7T).

GPT-4 generated fully well-formed UMRs under condition 0T, but there are occasional format errors under other conditions, and the higher temperature (0.7) leads to many more format errors. These include:

1. Quoted reentrancy, such as *:ARG0 (s24x)* where the variable should not be bracketed;
2. Duplicated variable names;
3. Extra right brackets ;
4. Unclosed brackets;
5. Multiple unconnected graphs in one sentence;
6. Unrelated content, extra explanations after the graph;

The four GPT-generated UMRs, after corrections of format errors, are tested against the gold data with four AnCast metrics (Sun and Xue, 2024): Labeled Relation F1 (LRM), Unlabeled Relation F1 (ULRM), Weighted Relation F1 (WLRM), and Concept F1 (CM), as well as Smatch (Cai and Knight, 2013) and Smatch++ (Opitz, 2023). All the scores are macro-averaged among the 26 sentences in an article, and the results for each article are presented in Table 1.

As can be observed from Table 1, the SMatch (SM) scores for the two articles are in the 40 and 50 percentage range, while the LRM scores, a harsher metric as a relation matches only if the concepts in the relation match as well, are in the 30 and 40 percentage range. There is no clear pattern as to which of the four conditions fares better, but some conditions work better for some sentences while other conditions work better for others.

4.3. Performance of human annotators

Each article is annotated by two pairs of annotators, with the first pair annotating from scratch and the second pair annotating from GPT-generated UMRs. The draft UMRs used for our human annotation experiment is generated with Think-Aloud prompting at temperature 0, and are fully well

Article 1	A-A	A1-G	A2-G	0F-G	7F-G	0T-G	7T-G
CM	78.52	93.72	88.32	79.03	75.93	65.61	81.90
ULRM	53.97	78.92	70.08	46.93	42.28	48.20	47.00
WLRM	53.05	77.64	70.66	41.16	37.62	42.72	42.88
LRM	52.00	78.08	68.66	43.61	38.47	44.44	43.00
SM	60.85	80.08	75.38	55.58	52.69	54.73	53.92
SM++	60.45	79.93	75.06	55.12	52.18	53.99	53.51
Article 2	A-A	A3-G	A4-G	0F-G	7F-G	0T-G	7T-G
CM	61.65	97.06	77.86	65.82	64.35	72.02	72.51
ULRM	42.88	85.31	42.44	34.35	34.96	31.37	33.69
WLRM	45.17	90.12	43.84	30.46	32.45	32.39	33.28
LRM	40.77	84.97	40.43	31.23	32.12	28.72	31.30
SM	53.15	87.96	55.00	46.81	46.85	41.62	44.35
SM++	53.33	88.23	54.70	47.15	46.74	41.26	44.12

Table 1: Inter-Annotator Agreement (IAA) and Automatic UMR Parsing Accuracy. The scores in the upper half are for Article 1, and that in lower half are for Article 2. The scores in the left half are for the IAA between human annotators and the scores of each annotator pair against the gold graph while the scores in the right half are for the GPT-generated UMRs against gold graphs. The leftmost column indicates the evaluation metrics we used: CM (concept match) measures the F1-score of the set of concepts annotated in two graphs; ULRM (unlabeled relation match) measures the F1-score of parent-child concept pairs in two graphs; LRM (labeled relation match) takes the relation labels into account when measuring the F1 of the parent-child concept pairs; WLRM (weighted labeled relation match) is a weighted version of LRM with more weight given to nodes that have more descendants. The top row indicates what is measured: A-A means inter annotator agreement; A1/3-G and A2/4-G compares the UMRs by two annotators in each article with gold graph; 0F-G, 7F-G, 0T-G, 7T-G: the four LLM parses under different setting compared to the gold graphs. The definitions of settings are explained in section 4.2. The gold graph is obtained by merging the two annotations after a discussion between the two annotators. The discrepancy in scores between the gold graphs and those of different annotators reflect the varying levels of proficiency in UMR annotation for the annotators. Article 2 is more colloquially written than 1, which adds to the difficulty of annotation and results in a lower IAA.

formed. The IAA is calculated based on the annotations from scratch. From Table 1, we can see that the IAA is 60.85 % and 53.15% respectively for the two articles in terms of the SMatch score, and 52% and 40.77% in terms of LRM. Since the annotators are still under training, the IAAs are acceptable. We also computed the average accuracy for each pair of annotators by comparing their annotations with the gold graphs, and as can be seen from the table, the scores for all metrics tend to be higher than the IAA, which is not surprising since the gold graph is the consensus graph that is closer in similarity to each of the annotations.

From Table 1, we can also see that the accuracy of GPT-generated graphs are not substantially lower than the IAA of human annotators. In particular, GPT-generated UMRs are particularly strong in terms of Concept F1, while human annotators are better at judging relations in UMR, as reflected in the much higher scores in terms of LRM and ULRM.

Our users used UMR Writer (Zhao et al., 2021) to annotate the sentence level UMRs from scratch. UMR Writer provides annotators with segmented sentences and dropdown menus for relation labels, abstract concepts, aspect attributes, modal strength values, and other items in the UMR vocabulary. When annotating from scratch, the users need to manually select the segmented words, and then choose the corresponding item in the UMR vocabulary from the dropdown menus to assemble the UMR graph piece by piece; if there is already annotated content, the annotator can use the "move" function to rearrange the subgraphs.

Revising GPT-generated UMRs vs annotating from scratch To answer the question of whether annotating from GPT-generated UMRs can speed up the annotation process, we asked the annotators to carefully record their time when annotating from scratch and from draft graphs, and the results are shown in Table 2. The result shows that annotators on average spend only 1/3 of the time when annotating from draft UMR graphs compared with annotating from scratch. This indicates a significant improvement in efficiency when LLMs are incorporated into the UMR annotation pipeline as a preprocessing step.

Article	Annotator	From Scratch	Annotator	From Draft Graphs	Ratio
1	A1	8h57min	A3	2h47min	3.19
	A2	9h03min	A4	2h52min	
2	A3	6h49min	A1	2h51min	2.61
	A4	8h47min	A2	3h08min	

Table 2: A comparison between the times needed for annotation from scratch and from draft graphs. The method for calculating the ratio involves computing the average annotation time for each sentence, and then taking the average between the two annotators.

After the annotation, we asked the annotators for feedback on what contributed to the speedup in annotation when GPT-generated UMRs are used as the starting point for manual correction and on what the main issues GPT-generated UMRs still have in order to inspect the acceleration with finer granularity. The main advantages of annotating from GPT-generated UMRs are that (i) especially for simple and short sentences, the GPT-generated UMRs are very accurate and are able to correctly annotate many concepts, abstract and concrete, as well as attributes, (ii) Many subgraphs that correspond to common patterns are correctly annotated, (iii) Reentrancies are correctly identified for the most part, and (iv) Some GPT-generated UMRs suggest interpretations of the sentence that even human annotators find difficult.

The annotators also identify areas where GPT-4 typically makes mistakes. They point out that GPT-4 often makes mistakes for long and complicated sentences that involve mulitple clauses, and often messes up the discourse relations between the clauses. GPT-4 also often fails to properly decompose long compounds words, which are very common in Chinese, into concepts. Finally, GPT-4 still tends to hallucinate relation labels that are not in UMR. This means that annotators would have to correct these mistakes when annotating from GPT-generated UMRs.

5. Related Work

Preprocessing in annotation is not a new idea, and it has been deployed in annotation tasks before. Especially for complicated annotation tasks, it has been shown to speed up annotation in treebanking (Chiou et al., 2001). Prior to the availability of LLMs, in order for pre-processing tool to produce annotation of high enough quality, it has to be trained on a significant amount of human annotated data. That means that before such a machine preprocessing - human correction process can start, a significant amount of data, sufficient to train a reasonably accurate machine learning model, has to be annotated by human annotators from scratch first. The availability of LLMs makes it possible to start this process much earlier if they can be prompted to generate the annotation without already having a significant amount of annotated data.

There is also prior work on using LLMs to generate Abstract Meaning Representations (AMRs) using GPT-4 (Ettinger et al., 2023) and comparing the quality of AMRs generated by LLMs with AMR parsers trained on million-plus human annotated AMRs. Their results show that while LLMs have shown some capability of generating AMRs, the quality of AMRs they generated are still substantially below that of state-of-the-art AMR parsers trained on large quantities of human annotated AMRs. They did not conduct experiments on whether the AMRs LLMs generated can help reduce the annotation time compared with human annotation from scratch.

6. Conclusion and Future work

In this paper, we investigated the question of whether LLMs, specifically GPT-4, can be used to speed up UMR annotation. Although the data set we used is relatively small, with only two articles, it is safe to conclude that incorporating LLMs into the annotation pipeline as a preprocess step can significantly reduce the amount of time (and cost) in UMR annotation. We also found that the accuracy of GPT-generated UMRs is not very far from the IAA from human annotators, with the caveat that the human annotators are still undergoing the training phase. The experiment on which prompting strategy produces the most accurate UMRs is inconclusive and additional experiments are needed to get a definitive answer. Future work also includes deploying LLMs to get modality, temporal dependency and coreference annotation at the document for UMR annotation.

Acknowledgements

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

ily reflect the views of NSF. We also wish to extend our appreciation to Cloudbank, which provided an indispensable computational resource for our experiments.

Limitations

The data set used in our experiments are relatively small, with only two documents that each have less than 30 sentences. However, we are confident with our conclusion that using LLMs as a preprocessing step speeds up UMR annotation.

7. Bibliographical References

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

Julia Bonn, Matthew Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jens E. L. Van Gysel, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos Yopán, Nianwen Xue, and Jin Zhao. 2024. Building an infrastructure for uniform meaning representations. In *Proceedings of LREC-COLING 2024*.

Julia Bonn, Andrew Cowell, Jan Hajic, Alexis Palmer, Martha Palmer, James Pustejovsky, Haibo Sun, Zdenka Uresova, Shira Wein, Nianwen Xue, et al. 2023. UMR annotation of multiword expressions. In *Proceedings of the 4th International Workshop on Designing Meaning Representations*.

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

Fu-Dong Chiou, David Chiang, and Martha Palmer. 2001. Facilitating treebank annotation using a statistical parser. In *Proceedings of the first international conference on human language technology research*.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator" : Limits of llms as analyzers of abstract meaning representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263.

Sijia Ge, Jin Zhao, Kristin Wright-Bettner, Skatje Myers, Nianwen Xue, and Martha Palmer. 2023. UMR-Writer 2.0: Incorporating a new keyboard interface and workflow into UMR-Writer. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 211–219.

Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Timothy J. O'Gorman, Andrew Cowell, William 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. *Künstliche Intelligenz*, pages 1–18.

Riitta Jääskeläinen. 2010. Think-aloud protocol. *Handbook of translation studies*, 1:371–374.

Sanghee Oh and B Wildemuth. 2009. Think-aloud protocols. *Applications of social research methods to questions in information and library science*, pages 178–188.

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

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the 30th International Conference on Computational Linguistics*. To appear.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Michael D Wolcott and Nikki G Lobczowski. 2021. Using cognitive interviews and think-aloud protocols to understand thought processes. *Currents in Pharmacy Teaching and Learning*, 13(2):181–188.

Jin Zhao, Nianwen Xue, Jens Van Gysel, and Jinho D Choi. 2021. UMR-Writer: A web application for annotating uniform meaning representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 160–167.