# Parsing Meaning Representations: is Easier Always Better?

**Zi Lin**[*]
Peking University
zi.lin@pku.edu.cn

**Nianwen Xue**
Brandeis University
xuen@brandeis.edu

## Abstract

The parsing accuracy varies a great deal for different meaning representations. In this paper, we compare the parsing performances between Abstract Meaning Representation (AMR) and Minimal Recursion Semantics (MRS), and provide an in-depth analysis of what factors contributed to the discrepancy in their parsing accuracy. By crystalizing the trade-off between representation expressiveness and ease of automatic parsing, we hope our results can help inform the design of the next-generation meaning representations.

## 1 Introduction

Meaning representation (MR) parsing is the task of parsing natural language sentences into a formal representation that encodes the meaning of a sentence. As a matter of convention in the field of natural language processing, meaning representation parsing is distinguished from semantic parsing, a form of domain-dependent parsing that analyzes text into executable code for some specific applications. Earlier work in semantic parsing focused on parsing natural language sentences into semantic queries that can be executed against a knowledge base to answer factual questions (Wong and Mooney, 2006; Kate and Wong, 2010; Berant et al., 2013; Artzi and Zettlemoyer, 2013). More recently, this line of work has been extended to parsing natural language text into computer programs (Ling et al., 2016; Yin and Neubig, 2017) and parsing tabular information in texts. Here we focus on the parsing of natural language sentences into domain-independent MRs that are not geared towards any one particular application, but could be potentially useful for a wide range of applications.

The challenge for developing a general-purpose meaning representation is that there is not a universally accepted standard and as a result, existing MRs vary a great deal with respect to which aspects of the linguistic meaning of a sentence are included and how they are represented. For example, existing MRs differ in whether and how they represent named entities, word sense, coreference, and semantic roles, among other meaning components.

These design decisions have consequences for the automatic parsing of these MRs. Among two of the meaning representations for which large-scale manual annotated data exist, the state-of-the-art parsing accuracy for AMR is generally in the high 60s and low 70s (May, 2016; May and Priyadarshi, 2017), while state-of-the-art parsing accuracy for (variations of) MRS is in the high 80s and low 90s (Oepen et al., 2014). Little has been done thus far to investigate the underlying causes for this rather large discrepancy. For purposes of developing the next generation MRs, it is important to know i) which aspects of the MR pose the most challenge to automatic parsing and ii) whether these challenges are "necessary evils" because the information encoded in the MR is important to downstream applications and has to be included, or they can be simplified without hurting the utility of the MR.

To answer these questions, we compare the parsing results between AMR and MRS, two meaning representations for which large-scale manually annotated data sets exist. We use the same parser trained on data sets annotated with the two MRs to ensure that the difference in parsing performance is not due to the difference in parsing algorithms, and we also use the same evaluation metric to ensure that the parsing accuracy is evaluated the same way. The evaluation tool we use is SMATCH (Cai and Knight, 2013), and the

[*] Work done during the internship at Brandeis University.

*Proceedings of the First International Workshop on Designing Meaning Representations*, pages 34–43
Florence, Italy, August 1st, 2019 ©2019 Association for Computational Linguistics

parser we use is CAMR (Wang et al., 2015a,b), a transition-based parser originally developed for AMR that we adapt to MRS. To make CAMR as well as SMATCH work on MRS data, we rewrote the native MRS data in PENMAN notation. Ideally, the parser needs to be trained on the same source text annotated with these two MRs to isolate the contributions of the MR from other factors, but this is not currently possible, so we fall back on the next best thing, and use data sets annotated with AMR and MRS that are similar in size.

Our experimental results show that the SMATCH score for MRS parsing is almost 20% higher than that for AMR. A detailed comparative analysis of the parsing results reveals that the main contributing factors into the lower parsing accuracy for AMR are the following:

- AMR concepts show a higher level of abstraction from surface forms, meaning that AMR concepts bear less resemblance to the word tokens in the original sentence.

- AMR does a much more fine-grained classification for the named entities than MRS, which contributes to errors in concept identification.

- Semantic relations are defined differently in AMR and MRS. While in AMR a semantic role represents a semantic relation between a verbal or nominal predicate and its argument, in MRS the predicate can also be a preposition, adjectives, or adverbs. Another difference is that while in AMR, the semantic roles for the core arguments of a predicate are interpretable with respect to an external lexicon, the semantic roles in MRS reflect the level of obliqueness and are linked to an external lexicon.[1]

We hope that by clearly identifying aspects of the MR that contributed to the challenges in automatic meaning representation parsing, we can help researchers make more informed decisions on the trade-off between representation expressiveness and ease of automatic parsing when developing the next-generation MRs.

The rest of the paper is organized as follows: Section 2 briefly describes the key elements of MRS and AMR; Section 3 reports our experiment setup and main parsing results for the two MRs; Section 4 provides a detailed analysis of the impacts of different aspects of the MR on automatic parsing. Section 5 concludes the paper.

## 2 Meaning Representations

In this section, we provide a brief description of the meaning representations that we investigate, Minimal Recursion Semantics (MRS) and Abstract Meaning Representation (AMR). Both MRs can be visualized as a graph with labeled nodes and edges. Figure 1 shows the MRS and AMR representations for the sentence “it has no bearing on our work force today”, which we will use to illustrate the various aspects of the two meaning representation frameworks.

### 2.1 Minimal Recursion Semantics

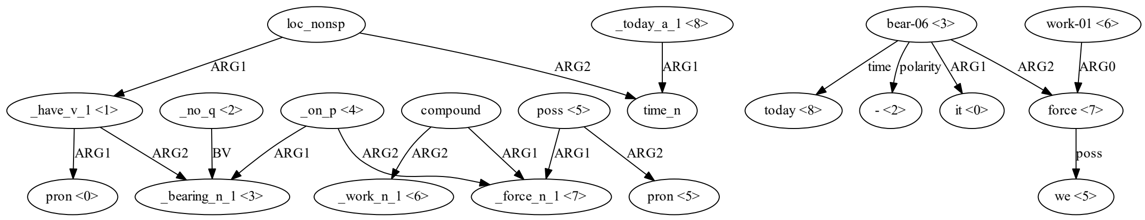
MRS serves as the logical-form semantic representation of the English Resource Grammar (ERG; Flickinger, 2000)[2], a broad-coverage grammar of English and an implementation of the grammatical theory of Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994). For our experiments, we use a variation of MRS called Elementary Dependency Structures (EDS; Oepen and Lønning, 2006), which retains the structural aspect of MRS that is of interest to us but excludes the morpho-syntactic features and the (underspecified) scopal information.

As can be seen from Figure 1a, nodes in an MRS representation are labeled with semantic predicates (e.g. `_bearing_n_1` and `compound`). MRS makes the distinction between surface and abstract predicates. A surface predicate consists of a lemma followed by (1) a coarse part-of-speech tag and (2) an optional sense label, which can be a number indicating the sense ID, a particle in the verb-particle construction (e.g., look up), or a case-marking prepositions (e.g., rely on). Examples of surface predicates are illustrated below:

- `_look_v_1`: Look how much air is moving around!

[1] These do not necessarily account for all the factors that might contribute to the discrepancy in performance between the two meaning representations. As one reviewer points out, the lack of manual alignment between word tokens in a sentence and the concepts in its AMR graph may also have contributed to challenge in parsing AMRs. Annotation consistency in the data set may also be a contributing factor. There are no obvious way to quantify these factors and we leave these to future research.

[2] http://www.delph-in.net/erg



(a) MRS graph

(b) AMR graph

```
(e3 / _have_v_1<1>
  :ARG1 (x5 / pron<0>)
  :ARG2 (x9 / _bearing_n_1<3>
    :ARG1-of (e14 / _on_p<4>
      :ARG2 (x15 / _force_n_1<7>
        :ARG1-of (e27 / compound
          :ARG2 (x26 / _work_n_1<6>))
        :ARG1-of (e21 / poss<5>
          :ARG2 (x20 / pron<5>))))
     :BV-of (_2 / _no_q<2>))
   :ARG1-of (e23 / loc_nonsp
     :AEG2 (x33 / time_n
       :ARG1-of (e23 / _today_a_1<8>))))
```

(c) MRS in PENMAN notation

```
(b / bear-06<3>
  :polarity -<2>
  :ARG1 (i / it<0>)
  :ARG2 (f / force<7>
    :poss (w2 / we<5>))
    :ARG0-of (w / work-01<6>)
  :time (t / today<8>))
```

(d) AMR in PENMAN Notation

Figure 1: The graphs and PENMAN notations of MRS and AMR for the sentence “it<0> has<1> no<2> bearing<3> on<4> our<5> work<6> force<7> today<8>” (From wsj_0003_30).

- _look_v_up: Researchers can look up credit ratings, and even question neighbors.
- _rely_v_on: We’ll rely very much on their leadership.

No lexical item can be associated with multiple surface predicates in MRS, but some lexical items bring abstract predicates, which is distinguished with no leading underscore. For example, in Figure 1a, the pronouns represented uniformly as pron, the compound (compounding *work* and *force*), loc_nonsp (an implicit locative without a specific preposition), and time_n decomposing the lexical item *time* are abstract predicates [3].

The edges in an MRS graph are labeled with a small set of roles that indicate the relation between a predicate and its argument (e.g., ARG1, ARG2) or between a predicate and a quantifier (e.g., BV). These roles are used to provide a numerical ID for the arguments of a predicate that occur in a sentence, and they are not interpretable with respect to an external taxonomy or valency lexicon. As a result, these numerical IDs are ordered and consecutive and it is not possible to have an ARG3 without an ARG1 and an ARG2. In general, ARG1 always corresponds to the first (least oblique) argument, ARG2 the second (next least oblique) argument, and so on.

### 2.2 Abstract Meaning Representation

AMR represents the meaning of a sentence as a rooted, labeled, directed, and acyclic graph (DAGs), as illustrated in Figure 1b. The nodes in an AMR graph are annotated with AMR concepts, which can also be concrete (surface) or abstract. A concrete concept is “evoked” by one or more lexical items in the sentence, while an abstract concept is inferred from a particular semantic context. A concrete concept can be a sense-tagged predicate (e.g., “bear-06” in Figure 1b) drawn from the Propbank (Palmer et al., 2005), or the lemma of a word in the sentence (e.g., “force” in Figure 1b. In general, only predicates that can be found in the PropBank frame files have their senses defined and annotated in AMR. Here are the four senses defined for the verb “bear” (excluding phrasal verbs)

- bear-01: hold, support, endure.
- bear-02: bear children.
- bear-03: move
- bear-06: has relation to

There is also a third type of concrete concepts that diverge further from their corresponding sur-

[3] For more details of the abstract predicates, please see: http://moin.delph-in.net/ErgSemantics/Basics

face lexical units and as we will show in Section 4, this is one aspect of AMR that poses a great deal of challenge to automatic parsing. For example, the modal verb “can” corresponds to the AMR concept “possible”. There are also other cases where a concept corresponds to a morpheme instead of the entire word. For example, the word “investor” is analyzed as

```
(p / person
    :ARG-of (i / invest-01))
```

and the concept “person” corresponds to the suffix “-or”.

In addition to concrete concepts, AMR also has abstract concepts that do not correspond to any lexical unit. For example, the concept “have-org-role-91” can be inferred from just the phrase “U.S. President Obama” as it implies that a person named “Obama” holds the position of the “president” in the organization that is the U.S. government:

```
(p / person
    :name (n / name :op1 "Obama")
    :ARG0-of (h / have-arg-role-91
        :ARG1 (c / country
            :name (n2 / name
                :op1 "US"))
        :ARG2 (p2 / president)))
```

The edges in an AMR graph are annotated with AMR relations, most of which can be viewed as semantic roles an argument plays with respect to its predicate. Although the naming convention of the semantic roles defined for the core arguments of a predicate in AMR is very similar to that used in MRS — both use an integer prefixed by “Arg” (e.g., ARG0, ARG1), that’s where the similarity ends. Unlike MRS, the semantic role for each core argument is defined for a specific sense of a predicate in the PropBank frame files, and can thus be interpreted. For example, for the predicate `bear-06`, the semantic roles for the core arguments are:

- `ARG1`: topic
- `ARG2`: related topic

In addition to the semantic roles for the core arguments, AMR uses a rather large set of semantic relations for non-core arguments. The semantic relations not tied to a specific predicate and include MANNER, TIME, DEGREE, etc. In total, there are 83 AMR relations.

# 3 Data preparation and parsing results

## 3.1 Data Preparation

We conduct the experiments on the dataset SDP2015[4] for MRS parsing and LDC2016E25[5] for AMR parsing. We use the PENMAN notation as the serialization format for both AMR and MRS. The PENMAN notation is the native format for the AMR data set, and we convert the MRS data to the PENMAN notation using the pyDelphin library. We use the training/development/test splits as recommended in the dataset releases. Some key statistics of the two data sets are presented in the top half of Table 1.

As we can see from the table, the number of sentences/graphs in the two data sets is similar in size, and this is important for purposes of comparing the parser performance on the two data sets. The number of nodes per token in MRS is much greater than that in AMR, this is mainly due to (1) the large number of abstract nodes in MRS and (2) the fact that the MRS concepts are much closer to the surface form than AMR (e.g., AMR does not have node representation for determiners, the infinitive marker “to”, prepositions that introduce oblique arguments and etc, while for the most cases, MRS does encode information for these function words).

## 3.2 Choosing a parsing model

Many parsers have been developed recently either for AMR parsing (Lyu and Titov, 2018; Groschwitz et al., 2018; Guo and Lu, 2018; Dozat and Manning, 2018; Wang et al., 2018; Wang and Xue, 2017; Wang et al., 2015a; Flanigan et al., 2014) or MRS parsing (Chen et al., 2018) , but relatively few parsers are capable of parsing both MR formalisms (Buys and Blunsom, 2017). To compare parsing results on MRS and AMR using the same parsing model, we need a parser that can parse another MR with minimal adaptation. In our experiment, we use CAMR, a transition-based parser [6] (Wang et al., 2015a) originally developed for AMR parsing that we also adapt to MRS parsing.

CAMR performs MR parsing in two steps. The first step is to parse a sentence into a dependency

[4] http://sdp.delph-in.net/2015/data.html

[5] https://catalog.ldc.upenn.edu/LDC2016E25

[6] https://github.com/c-amr/camr

| | MRS | | | AMR | | |
|---|---|---|---|---|---|---|
| | **Train** | **Dev** | **Test** | **Train** | **Dev** | **Test** |
| number of graphs/sentences | 35,315 | 1,410 | 1,410 | 36,521 | 1,368 | 1,371 |
| number of tokens per sentence | 22.33 | 22.92 | 23.14 | 17.83 | 21.59 | 22.10 |
| number of nodes per token | 0.96 | 0.97 | 0.93 | 0.68 | 0.70 | 0.70 |
| | **Node** | **Edge** | **SMATCH** | **Node** | **Edge** | **SMATCH** |
| CAMR | 89.4 | 81.1 | 85.3 | 78.7 | 57.1 | 68.0 |
| Buys and Blunsom (2017) | 89.1 | 85.0 | 87.0 | - | - | 61.2 |
| Chen et al. (2018) | 94.5 | 87.3 | 90.9 | - | - | - |
| Lyu and Titov (2018) | - | - | - | 85.9 | 69.8 | 74.4 |

Table 1: Statistics and parsing results for MRS and AMR on the test set

tree with an off-the-shelf parser, and the second step is to transform the dependency tree into the target MR graph by performing a series of actions each of which changes a parsing state to a new state. See Wang et al. (2015b,a) for details on how CAMR works.

As we described in Section 2, both AMR and MRS abstract away from the surface lexical units and the nodes in the MR graph are not simply word tokens in the sentence. In order to train CAMR, the word tokens in the sentence need to be aligned with nodes in the meaning representation graph to the extent that is possible. The MRS data comes with manual alignments, but the AMR data set does not, so we utilize the automatic aligner in `JAMR` (Flanigan et al., 2014) to align the word tokens in the sentence with nodes in the AMR graph.

In our experiment, we use the Stanford CoreNLP toolkit (Manning et al., 2014) to produce the dependency structure that we use as input to CAMR. We also use this toolkit to produce part-of-speech tags and name entity information for use as features. Considering the need for cross-framework MR parsing, we do not make use of a semantic role labeler as the original CAMR does, as semantic role labeling is irrelevant to MRS parsing. This hurts the AMR parsing somewhat but not by too much. When adpating CAMR to MRS, we perform the following post-processing steps: (1) changing the AMR-style naming convention for named entities `name` and `:op` to MRS-style `named` (or other date-entity nodes) and `:carg`; (2) if the word is unknown to the parser, copying the lemma and the predicted POS tag to form an "unknown word"; (3) disabling the functionality for classifying named entities; (4) adding the abstract node "nominalization" if a predicate has been nominalized.

### 3.3 Parsing Results

The results based on the SMATCH score (Cai and Knight, 2013) are reported in Table 1. We also include the state-of-the-art parsers for each framework (an SHRG-based parser for MRS (Chen et al., 2018) and a neural AMR parser (Lyu and Titov, 2018)) as well as a cross-framework neural parser in Buys and Blunsom (2017). For CAMR, the gap in F1 between the two frameworks is 17.3% and the difference is larger for Buys and Blunsom (2017), which is more than 20%.

## 4 What makes AMR parsing difficult?

To investigate which aspects of the MRs contribute to the large gap in performance between AMR and MRS parsing, we perform a detailed examination of different aspects of the meaning representation parsing process.

### 4.1 Concept Detection

The first step in constructing a meaning representation graph is concept identification, or determining the nodes of the meaning representation graph. As should be clear from our description in Section 2, the concepts in an AMR or MRS graph abstract away from the surface lexical units in a sentence, and as a result, it is non-trivial to predict the concepts in a meaning representation graph based on the word tokens in a sentence. This process can be as simple as performing lemmatization, but it can also be as complicated as performing word sense disambiguation or even inferring abstract concepts that do not correspond to a particular word token.

**Word sense disambiguation** For AMR parsing, word sense disambiguation means recognizing the sense defined in the PropBank frame files (e.g., `bear-01` vs. `bank-06`) and needs to be

|  |  |  | MRS |  |  |  |
|---|---|---|---|---|---|---|
| POS | % | #lemma | #sense | average | score | WSD |
| n | 34.46 | 1,420 | 1,434 | 1.01 | 95.35 | 99.76 |
| v | 20.37 | 838 | 1,010 | 1.21 | 85.56 | 90.58 |
| q | 13.97 | 25 | 25 | 1.00 | 98.22 | 100.00 |
| p | 12.86 | 96 | 123 | 1.28 | 81.29 | 76.11 |
| a | 11.45 | 637 | 648 | 1.02 | 90.58 | 99.90 |
| c | 4.20 | 17 | 19 | 1.12 | 94.46 | 99.61 |
| x | 2.69 | 80 | 81 | 1.01 | 73.65 | 99.74 |
| total | 100.00 | 3,113 | 3,340 | 1.07 | 90.78 | 97.06 |
|  |  |  | AMR |  |  |  |
| pred | - | 1,292 | 1,440 | 1.11 | 77.93 | 94.54 |

Table 2: Node identification and WSD results on MRS in terms of noun (n), verb (v), quantifier (q), preposition (p), adjective (a), conjunction (c), and others (x), and on AMR in terms of predicate (pred). Both are measured on the test set in terms of accuracy based on SMATCH.

performed on verbal, nominal and other predicates. For MRS parsing, word sense disambiguation needs to be performed all the concepts that are not constants (number, date and named entities) or abstract concepts (`compound`, `subord`, etc.).

Table 2 reports the accuracy based on the SMATCH for concept detection in general [7], and concepts that requires word sense disambiguation to identify on the test set. We also present a concept detection accuracy breakdown by the part of speech of the words that they are aligned to. As we can see from the table, the overall concept detection accuracy is much lower for AMR than MRS. However, for concepts that involve word sense disambiguation, the difference is rather small, indicating that word sense disambiguation is not a major contributor in the performance gap.

**Concept abstraction** Now that we have established that word sense disambiguation is not a major contributor to the difficulty in concept detection for AMR parsing, we take a closer look at how concept detection fared for lexical categories that are known to have a complex mapping to the concepts they "evoke". For lack of a better term, we call this "concept abstraction". We will examine how abstraction of verbs (*v.*), nouns (*n.*), adjectives (*adj.*), adverbs (*adv.*), prepositions (*prep.*), conjunctions (*conj.*), phrasal verbs (*p.v.*) and modal verbs (*mod.*) impact concept detection accuracy.

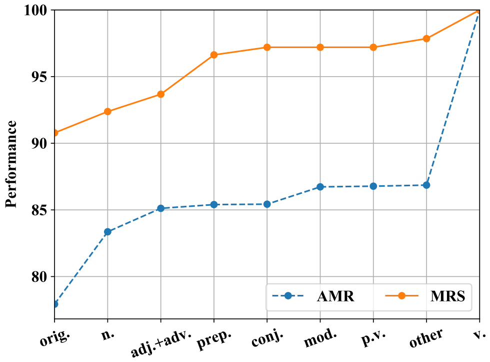


Figure 2: Relative improvement of performance on the test set after correcting each type of POSs or constructions in AMR

- **Phrasal verbs** AMR tends to normalize phrasal verbs to single verbs where possible. For example, the same predicate `bathe-01` is used for both "take a bath" and "bathe".

- **Nouns**. The verb and its nominalization often share the same predicate in AMR. For example, the predicate for both "destruction" and "destroy" is `destroy-01`.

- **Adjectives**. Like nouns, an adjectival predicate is normalized to the same form as that of its verbal counterpart if the adjective is derived from a verb. For example, the predicate for "attractive" is `attract-01`. This obviously does not apply adjectives like "quick" and "tall", which do not have a verbal counterpart.

- **Adverbs with the suffix -ly**. The Predicate of an adverb is often normalized to its adjectival form. For example, for both "quickly" and "quick", the predicate is `quick-01`.

- **Prepositions**. Most prepositions do not map to a concept in AMR except for idiomatic constructions such as "out of mind", whose predicate is `out-06`.

- **Conjunctions**. The concepts for coordinating conjunctions can be very different from their surface form. For example, the concept for "but" is `constrast-01`.

- **Modal verbs**. The AMR concepts for modal verbs are also very different from its surface form. For example, the predicate for the modal verb "can" is `possible-01`, the same as that for the adjective "possible".

[7] The accuracy is calculated between the gold and the parsed graphs, regardless of the alignment to surface substrings.

| type | *n.* | *adj.* | *adv.* | *prep.* | *conj.* | *mod.* | *p.v.* | other | *v.* |
|---|---|---|---|---|---|---|---|---|---|
| % | 35.09 | 10.05 | 1.87 | 1.17 | 1.01 | 2.59 | 0.31 | 0.15 | 47.76 |
| Performance | 83.01 | 84.44 | 80.73 | 73.53 | 96.61 | 66.96 | 83.33 | 44.44 | 74.07 |

Table 3: Individual percentage and score for different types of AMR's predicates

| Entity type | Example | AMR | MRS |
|---|---|---|---|
| calendar | lunar calendar | `(d / date-entity :calendar (m / moon))` | - |
| month | December (8th) | `(d / date-entity :month 12)` | `(x1 / mofy :carg "Dec")` |
| weekday | Monday | `(d / date-entity :weekday (m / monday))` | `(x1 / dofw :carg "Mon")` |
| day | (December) 8th | `(d / date-entity :day 8)` | `(x1 / dofm :carg "8")` |
| dayperiod | night | `(d / date-entity :dayperiod (n / night))` | - |
| named entity | New York | `(c1 / city`<br>`  :name (n1 / name`<br>`    :op1 "New" :op2 "York"))` | `(x1 / named :carg "York"`<br>`  :ARG1-of (e1 / compound`<br>`    :ARG2 (x2 / named :carg "New")))` |

Table 4: Date-entity of AMR and MRS. The `carg` in MRS means "constant argument", which takes as its value a string representing the name of the entity.

To identify the lexical categories or constructions that evoke the concepts, we first extract words or word sequences that are aligned with these concepts, and then use a set of heuristics based on morpho-syntactic patterns to determine the exact type of abstraction in the test set. We measure the improvement in concept detection accuracy if concepts for each additional category are correctly detected. If there is a big improvement in accuracy if we assume the concepts are correctly detected for that category, that means concept detection for that category is a big challenge. The accuracy will remain unchanged if the type is undefined for that MR (e.g. *p.v.* for MRS). MRS labels most of the adverbs as its corresponding adjective form, so we merge these two types together.

The individual result is reported in Table 3 and the improvement is illustraed in Figure 2, which shows that concept detection accuracy in AMR is mainly dragged down by nouns and verbs due to their relatively large proportions. While prepositions play an important role in concept detection in MRS, most prepositions do not map to concepts in AMR and thus do not contribute to the errors in AMR concept detection.The concept detection for modal verbs is also difficult for AMR but not for MRS.

**Named and date entities** We next examine how well entities are detected in AMR and MRS parsing. Named and date entities are typically multi-Word expressions (MWEs) that do not have a simple mapping between the word tokens in a sentence and the concepts in a meaning representation graph. In AMR, date entities are mapped to a `date-entity` concept with an attribute that indicates the specific type of entity. Named entities are mapped to a `name` concept with a detailed classification of the named entity type (e.g., `city`, `country`). AMR defines 124 total entity types, a very fine-grained classification. In MRS, date entities map to a date entity type ("season") with an attribute that is a constant ("winter"). Named entities are treated as a type of a `compound` that has a `named` concept as its argument. MRS does not provide a detailed classification of named entities. More examples of AMR and MRS date (the first five rows) and named entities (the last row) are provided in Table 4.

| dataset | MRS | | AMR | |
|---|---|---|---|---|
| | # | score | # | score |
| date entity | 266 | 92.48 | 273 | 66.67 |
| NE detection | 2,555 | 81.96 | 2,065 | 91.09 |
| NE classification | - | - | - | 76.46 |

Table 5: Results on entity recognition on the test set

The results for detecting date and named entities on the test set are presented in Table 5. A date or named entity is correctly detected if the entire predicted subgraph matches the gold subgraph for the entity. For named entities, we evaluate the named entity detection and named entity classification separately, given the fact that MRS does not classify named entities at all. We can see that the date entity detection accuracy for AMR is much lower than that for MRS, indicating some of the normalization that is needed to map word tokens to AMR concepts is difficult for the parser ("lunar calendar" to `(d/ date-entity`

`:calendar (m / moon))`. For named entities while the named entity detection accuracy is higher for AMR than MRS, but since AMR parsing also requires named entities be correctly classified, overall correctly parsing named entities in AMR is still much harder.

### 4.2 Relation Detection

In this section, we consider the subtask of relation detection in meaning representation parsing, which involves identifying and labeling the edges in the meaning representation graph. We focus on Semantic Role Labeling (SRL) of the core arguments, arguments that receive the label `ARG-i`, where `i` is an integer that ranges from 0 to 5. In order to isolate the effect of SRL, we only consider cases where the concepts (nodes) have been correctly detected. The results on the test set are presented in Table 6. The overall results are based on the SRL smatch computed on `:ARG-i` roles using the toolkit `amr-eager`[8]. Here "all matched" refers to complete match, i.e., the predicted subgraph rooted in the predicate [9] match the gold subgraph. Note that both MRS and AMR graphs contain reentrancy, meaning that the same concept can participate in multiple relations, so we also include a separate evaluation of reentrancy.

As we can see, the accuracy for both SRL in general and reentrancy in particular is much lower for AMR than MRS, and the number of reentrancies is much greater for AMR than MRS. [10] A closer look reveals that the main cause for the difference in performance lies in the different ways of how MRS and AMR represent the prepositional phrases and coreferene, as well as how the semantic roles are defined for the two MRs.

**Prepositional phrases** MRS treats prepositions as predicates, and labels their arguments, while AMR just drops the preposition when it introduces an oblique argument for a verbal predicate so the object of the preposition becomes an argument of the verbal predicate, resulting in non-local relations. This explains why SRL is more difficult for AMR than MRS, illustrated in the top example in Figure 3, where there are different representations for the prepositional phrase in the sentence. The MRS design choice, in this case, leads to more structures to predict, compared with just one structure in AMR. Assuming these sub-graphs are comparatively easy to predict, this may contribute to higher scores in MRS parsing.

| dataset | MRS | | AMR | |
|---|---|---|---|---|
| | # | score | # | score |
| Overall | - | 81.76 | - | 61.52 |
| All matched | 3,398 | 63.48 | 4,975 | 44.77 |
| *ARG0* | 3,087 | 62.00 | 3,680 | 49.43 |
| *ARG1* | 2,985 | 68.45 | 5,377 | 53.97 |
| *ARG2* | 339 | 35.09 | 1,614 | 37.86 |
| *ARG3* | 7 | 57.13 | 123 | 14.63 |
| *ARG4* | - | - | 39 | 20.51 |
| Reentrancy | 807 | 81.28 | 1,723 | 43.91 |

Table 6: Results on SRL. MRS's argument number begins at 1 so we just move all the argument to begin at 0 to make them comparable.

**Coreference** AMR resolves sentence-level coreference, i.e., if there is more than one expression in the sentence referring to the same entity, that entity will be an argument for all the predicates that it is an argument of. In contrast, MRS does not resolve coreference and each instance of the same entity will be a separate concept in the MRS graph. This is illustrated in bottom example in Figure 3. The labeled arguments for the predicate "eat" in the two MRs are totally different but actually they refer to the same entities. Not having to do coreference resolution makes MRS parsing easier and this also explains the lower SRL accuracy for AMR.

**Interpretability of semantic roles** To see the difference in how the semantic roles are defined between MRS and AMR, we conduct a controlled experiment on a subset of 87 graphs in both datasets that all annotate the same source text. After extracting the overlapping predicates (based on the alignments for each MR, gold for MRS and automatic alignment for AMR) and computing the agreement between the semantic roles in the two MRs, we find an interesting fact: the labeled agreement in the subset is rather low ($F1 = 52.22$), but the unlabeled agreement is

[8] `https://github.com/mdtux89/amr-eager`

[9] For MRS we only count the verbs, so the number of predicates and arguments is much greater for AMR than MRS.

[10] This may seem to contradict the observation in Kuhlmann and Oepen (2016) where they show MRS has more re-entrancies than AMR. This is because in our experiments we removed the edge linking a conjunction to its conjunct to remove the cycles that would have a negative impact on parsing accuracy but do not offer further information. This accounts for most of the re-entrancies in the EDS variant of MRS.

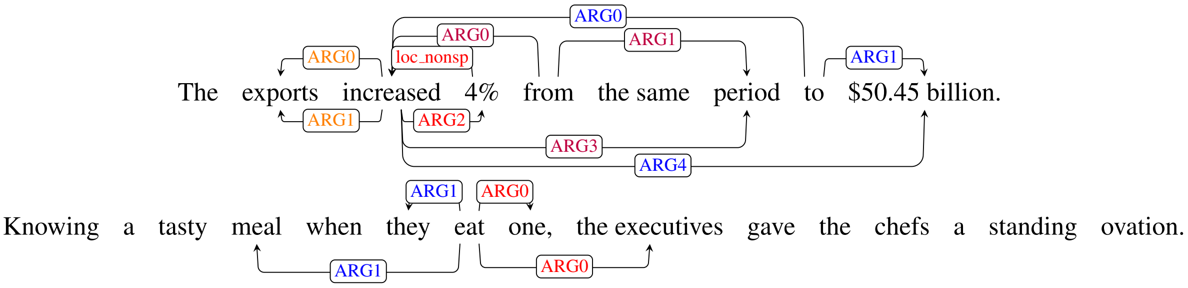

ARG0
ARG0
ARG1
ARG0
loc_nonsp
ARG1
The exports increased 4% from the same period to $50.45 billion.
ARG1
ARG2
ARG3
ARG4
ARG1
ARG0
Knowing a tasty meal when they eat one, the executives gave the chefs a standing ovation.
ARG1
ARG0

Figure 3: The SRL representations of MRS (edge above) and AMR (edge below) for the sentences "the exports increased 4% from the same period to $50.45 billion" and "knowing a tasty and free meal when they eat one, the executives gave the chefs a standing ovation". For `increase-01`, PropBank defines the `ARG0` and `ARG1` as "cause of increase" and "thing increasing", so "the exports" here will be labeled as `ARG1` instead of `ARG0`.

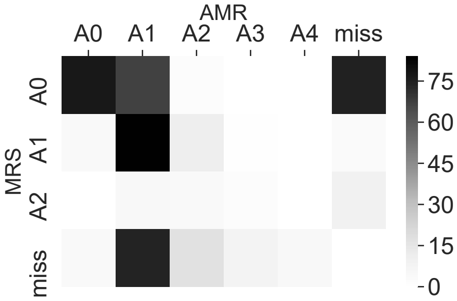


Figure 4: Confusion matrix between MRS and AMR

much higher ($F1 = 77.83$). The low labeled agreement can be explained by the different ways of how semantic roles are defined. We illustrate this difference using the confusion matrix in Figure 4. The numeric value of the semantic roles tends to be smaller in MRS than in AMR. As discussed in Section 2, while the semantic roles in MRS represent the level of obliqueness of arguments realized in a particular sentence, the semantic roles in AMR are defined for the *expected arguments* of a predicate in an external lexicon that is independent of any particular sentence. The semantic roles for the arguments that actually occur in a particular sentence may be discontinuous in a particular context, making them more difficult to predict.

## 5 Conclusion

In this work, we evaluated the similarities and differences in the semantic content encoded by Minimal Recursion Semantics (MRS) and Abstract Meaning Representation (AMR). After parsing the two MRs using the same parser and evaluating them using the same metric, we provide a detailed analysis of the differences between the two MRs in both substance and style that leads to a large gap in automatic parsing performance. In doing so, we help crystalize the trade-off between representation expressiveness and ease of automatic parsing and hope this study will inform the design and development of next-generation MRs.

## Acknowledgement

We thank the anonymous reviewers for their helpful comments, and Junjie Cao for discussing the details of meaning representations with us. This work is supported by a grant from the IIS Division of National Science Foundation (Award No. 1763926) entitled "Building a Uniform Meaning Representation for Natural Language Processing" awarded to the second author. All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

## References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. *arXiv preprint arXiv:1704.07092*.

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)*, volume 2, pages 748–752.

Yufei Chen, Weiwei Sun, and Xiaojun Wan. 2018. Accurate shrg-based semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 408–418, Melbourne, Australia. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2018. Simpler but more accurate semantic dependency parsing. *arXiv preprint arXiv:1807.01396*.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.

Dan Flickinger. 2000. On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Jonas Groschwitz, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. Amr dependency parsing with a typed semantic algebra. *arXiv preprint arXiv:1805.11465*.

Zhijiang Guo and Wei Lu. 2018. Better transition-based amr parsing with a refined search space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1712–1722.

Rohit J Kate and Yuk Wah Wong. 2010. Semantic parsing: The task, the state of the art and the future. *Tutorial Abstracts of ACL 2010*, page 6.

Marco Kuhlmann and Stephan Oepen. 2016. Towards a catalogue of linguistic graph banks. *Computational Linguistics*, 42(4):819–827.

Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, Andrew Senior, Fumin Wang, and Phil Blunsom. 2016. Latent predictor networks for code generation. *arXiv preprint arXiv:1603.06744*.

Chunchuan Lyu and Ivan Titov. 2018. Amr parsing as graph prediction with latent alignment. *arXiv preprint arXiv:1805.05286*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Jonathan May. 2016. Semeval-2016 task 8: Meaning representation parsing. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1063–1073.

Jonathan May and Jay Priyadarshi. 2017. Semeval-2017 task 9: Abstract meaning representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 536–545.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based mrs banking. In *LREC*, pages 1250–1255.

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

Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Chuan Wang and Nianwen Xue. 2017. Getting the most out of amr parsing. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1257–1268.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015a. Boosting transition-based amr parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 857–862.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015b. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.

Yuxuan Wang, Wanxiang Che, Jiang Guo, and Ting Liu. 2018. A neural transition-based approach for semantic dependency graph parsing. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yuk Wah Wong and Raymond J Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 439–446. Association for Computational Linguistics.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. *arXiv preprint arXiv:1704.01696*.