Inspecting Soundness of AMR Similarity Metrics in terms of Equivalence and Inequivalence

Kyung Seo Ki*
Department of
Intelligence and Information
Seoul National University
Seoul, Republic of Korea
kskee88@snu.ac.kr

Bugeun Kim*
Department of
Artificial Intelligence
Chung-Ang University
Seoul, Republic of Korea
bgnkim@cau.ac.kr

Gahgene Gweon
Department of
Intelligence and Information
Seoul National University
Seoul, Republic of Korea
ggweon@snu.ac.kr

Abstract

In this study, we investigate soundness of current Abstract Meaning Representation (AMR) similarity metrics in terms of equivalence and inequivalence. Specifically, AMR guidelines provide several equivalence and inequivalence conditions to reflect the meaning aspect of the semantics. Thus, it is important to examine an AMR metric's soundness, i.e., whether the metric correctly reflects the guidelines. However, the existing metrics have less investigated their soundness. In this work, we propose a new experimental method using simulated data and a series of statistical tests to verify the metric's soundness. Our experimental result revealed that all existing metrics such as SMATCH, SEMBLEU, S^{2}MATCH, SMATCH++, WWLK$_\theta$, WWLK$_{e2n}$, and SEMA did not fully meet the AMR guidelines in terms of equivalence and inequivalence aspects. Also, to alleviate this soundness problem, we propose a revised metric called SMATCH$^\sharp$, which adopts simple graph standardization technique that can improve the soundness of an existing metric.

1 Introduction

In this paper, we propose a new experimental method to evaluate soundness of Abstract Meaning Representation (AMR) similarity metrics and try to address the soundness of AMR similarity metrics by proposing a revised metric, SMATCH$^\sharp$. AMR is a widely used formalism that expresses the semantic aspect of natural language sentences. The formalism is based on neo-Davidsonian semantics (Banarescu et al., 2013; Davidson, 1967; Higginbotham, 1985; Parsons, 1990). Therefore, when comparing two AMR graphs, a metric needs to yield results that adhere to such theoretical background, which is implemented in the AMR guidelines (Banarescu et al., 2019). We refer to this criterion as the *soundness* of an AMR metric. Here, we define soundness as a metric's quality to yield well-founded results that adhere to the theoretical background of AMR during the metric's computation process. For example, soundness of a metric can be operationally checked by whether the metric correctly follows AMR guidelines, as AMR guidelines define many special equivalence relationships between two AMRs with different forms along with its theoretical background. Thus, an AMR metric should treat such AMRs as equivalent to meet the soundness criterion.

However, the existing metrics' design has been less focused on evaluating their soundness. Several metrics have been proposed to measure the similarity between two AMRs, including SMATCH (Cai and Knight, 2013), SEMBLEU (Song and Gildea, 2019), S^{2}MATCH (Opitz et al., 2020), WWLK$_\theta$-variants (Opitz et al., 2021; Opitz and Frank, 2022), SEMA (Anchiêta et al., 2019), and SMATCH++ (Opitz, 2023). Although these existing metrics have helped evaluating the quality of various AMR parsers, they do not sufficiently consider soundness. The only exception is SMATCH++, which attempts to address soundness partially by managing some equivalent cases, like reification. Nonetheless, even SMATCH++ has not reported whether their metric adheres to other equivalent cases specified in the AMR guideline.

Therefore, we designed an experiment that investigates the soundness of AMR metrics, using systematically simulated data. We implement both 6 equivalent cases and 7 inequivalent cases according to AMR guideline, to make a systematic data for evaluating the soundness of metrics. We also propose a simple statistical method to verify soundness and a graph standardization method for handling equivalence and inequivalence cases. As a result, we propose SMATCH$^\sharp$, an enhanced version of SMATCH++, as an alternative to prior AMR metrics that better addresses soundness.

Our paper is structured as follows: Section 2 provides theoretical background on AMR and assesses

*These authors contributed equally to this work.

*Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 402 - 409

the designs of existing metrics from the perspective of equivalence and inequivalence. Next, Sections 3 details our experimental design. Specifically, Section 3 outlines the simulated dataset generation, the proposed statistical test for soundness verification, the SMATCH$^\sharp$ metric, and implementation details. Finally, Section 4 presents the results, and discuss their implications. We analyzes the results from applying our experiment to various AMR metrics and examines their soundness issues.

2 Inspecting AMR Similarity Metrics

Here, we discuss seven existing AMR similarity metrics in terms of the way that they handle equivalent and/or inequivalent cases. As widely used similarity metrics adopt a method of giving partial credits to non-exact matching cases, existing metrics differ in how they establish the range of partial credit regarding equivalence and inequivalence of AMR components. Thus, we categorize the existing metrics into two types: (1) allowing credits only to exact equivalent components, and (2) allowing credits also to some inequivalent cases.

First, there are metrics that only give credit for exactly matching/overlapping components when measuring the similarity between two AMRs. SMATCH (Cai and Knight, 2013), SEMBLEU (Song and Gildea, 2019), SEMA (Anchiêta et al., 2019), and SMATCH++ (Opitz, 2023) belong to this category. These metrics approximately compute the maximum overlap between two AMRs, by constructing a mapping between substructures of two AMRs. For example, SMATCH computes overlap as the maximum F_1 score of common triples between two AMRs. Similarly, in SEMBLEU, the metric computes overlap as the BLEU score using n-grams of triples commonly appearing in the two given AMRs. However, these overlap-based metrics can mistakenly identify equivalent AMRs as inequivalent, since they primarily focus on matching exactly the same components without fully considering the AMR guidelines. As the guidelines define some cases where AMRs are syntactically inequivalent but semantically equivalent, the soundness of the evaluation may decrease in some cases.

Second, for the metrics allowing credits also to some inequivalent cases, they try to measure similarity by relaxing the constraint of exact match. Metrics such as S^{2}MATCH, WWLK$_\theta$, and WWLK$_{\text{e2n}}$ (Opitz et al., 2020, 2021; Opitz and Frank, 2022) belong to this category. These metrics attempt to give partial credit for inequivalent AMRs by incorporating the concept of pragmatic sense, acquired by a language model. With a language model, these metrics are able to construct intuitive sense of similarity between some predicates or between some instances. However, the use of language models makes it difficult to verify that these metrics fairly assess the meaning of AMRs independently of any context, which contradicts one of the key assumptions behind AMR - that meaning should be context-independent. More specifically, the AMR guideline tries to ensure context-independency of semantics by using pre-defined ontology of predicate senses, semantic roles, and frame arguments from OntoNotes (Pradhan et al., 2007) and PropBank (Kingsbury and Palmer, 2003). Using a language model may compromise such context-independency when comparing two AMRs, since a language model tries to treat different pre-defined senses, roles, and arguments as similar ones. Moreover, such intuitive sense of similarity may weaken the transparency of the evaluation process.

We suspect that all the metrics in the above two categories may insufficiently handle equivalent and inequivalent cases according to the AMR guidelines. For example, the case illustrated in Appendix A shows that some prior metrics do not correctly evaluate inequivalent AMRs which have different meanings. Note that Goodman (2019) have already shown that not handling these conditions results in an unfair evaluation in SMATCH. We suspect that other metrics have also insufficiently considered the issues raised by Goodman (2019), because other metrics were not designed to properly handle equivalent and inequivalent cases according to the AMR guidelines. Moreover, existing metrics have not systematically verified whether they conform to the equivalence/inequivalence conditions based on the AMR guidelines. Systematic verification of these conditions would therefore be helpful to identify strengths and weaknesses of the existing AMR metrics.

3 Experiment

To verify the soundness of existing metrics, we designed an experiment based on the observations on equivalence and inequivalence aspects. We tested seven existing metrics and one new metric: SMATCH, S^{2}MATCH, SEMBLEU, SMATCH++,

Equivalent cases (6 operations)	
	(When writing PENMAN notation,)
Lift Up	Lift another node as a root.
Reorder	Randomly re-order edges.
Relabel	Randomly re-labeled variables.
Reify	Apply the reification process.
De-reify	Apply the de-reification process.
Duplicate	Duplicate all edge twice.
	(Semantically equivalent due to tautology)
Inequivalent cases (7 operations)	
Insert N	Insert a dummy node.
Insert E	Insert a dummy edge between nodes.
Change N	Change a node's name with a dummy.
Change E	Change an edge's label with a dummy.
Delete N	Delete a node.
Delete E	Delete an edge.
Swap	Swap heads of two edges.

Table 1: List of 13 operations for our simulated dataset

WWLK_{θ}, WWLK_{e2n}, SEMA, and SMATCH$^{\sharp}$. Note that SMATCH$^{\sharp}$ is our revised version of SMATCH++ which tries to handle equivalence and inequivalence cases. Using our simulated data and statistical methods, we tested whether these eight metrics follow the AMR guideline.

SMATCH$^{\sharp}$ To consider the AMR guidelines while upholding the approximation method of the existing metrics, we developed SMATCH$^{\sharp}$. The new metric is a variant of SMATCH++ which standardizes AMR graphs considering both equivalence and inequivalence conditions. As SMATCH++ is the only metric that attempts to handle some of the equivalence conditions, we chose to make SMATCH$^{\sharp}$ based on SMATCH++. Thus, SMATCH$^{\sharp}$ retains the same evaluation process as Smatch++. However, SMATCH$^{\sharp}$ is additionally designed to pass through a single graph standardization pipeline before the evaluation stage. This additional pipeline is a normalization technique that converts any given AMR into a single, standardized form. This normalization is necessary because we want to ensure that semantically equivalent AMRs are treated correctly during evaluation. For example, some cases such as inversion, different variable names, etc. should be treated as equivalent according to AMR's definition, and can be converted into the exact same notation through normalization.

Simulated Dataset with 13 Operations We have designed a novel test method to verify how well existing metrics conform to the AMR guidelines. Our test employs the gold standard dataset, AMR 3.0[1], which is commonly used in the development of existing AMR parsers. First, we extracted 20,000 AMRs by randomly sampling the AMR 3.0 train set. For each AMR in this gold standard dataset, we applied 13 perturbations, shown in Table 1, following the guidelines to create a simulated dataset. This perturbing procedure generated 260,000 simulated pairs. This simulated dataset helps us verify whether an AMR metric can evaluate the original and perturbed cases as equivalent. For six of the perturbations, we applied one of the six equivalent cases described in Part III (Phenomena) of the AMR guidelines, making the original and perturbed pair equivalent. For seven of the pairs, we randomly manipulated the structure of the given AMRs, making the original and perturbed pair inequivalent. Refer to the Appendix B for the detailed illustration and example for each operation. To the best of our knowledge, this is the first attempt to verify the soundness of an AMR metric, which concerns how well the metric adheres to the rules of the representation being evaluated.

Statistical Test for Hypothesis A sound metric should differentiate equivalent pairs and inequivalent pairs. To verify this, we conducted a binomial test to statistically examine the difference between the average score ζ of equivalent pairs and the theoretical maximum score ζ_{max} for each metric. The test process involves three steps. As the first step, we compute each metric score ζ for each graph pairs. In the second step, we compute $P(\zeta = \zeta_{max})$, i.e., the proportion of examples where the score reaches ζ_{max}. Lastly, in the third step, we tested $P(\zeta = \zeta_{max}) > 0.999$ for equivalent cases and $P(\zeta = \zeta_{max}) < 0.001$ for inequivalent cases[2]. So, a sound metric should pass all of the above tests by definition. Corollary, such a metric should prevent overlap between the score ranges of equivalent pairs and ranges of inequivalent pairs.

Implementation Detail Here, we implemented the eight metric as follows. For SMATCH, S^{2}MATCH, SMATCH++, and WWLK_{e2n}, we ran the exact official code. For SEMBLEU and SEMA, we additionally added an outputting code into the original source code to obtain a score for each

[1] https://catalog.ldc.upenn.edu/LDC2020T02

[2] We set $P(\zeta) > 0.999$ for equivalent cases and $P(\zeta) < 0.001$ for inequivalent cases, since the statistical power is greater than 0.999 even with the significance level of 0.001.

AMR pair[3]. Lastly, for WWLK$_\theta$, we used the reified version of STS for zeroth-order learning[4]. For SMATCH$^\sharp$, we used `Penman` (Goodman, 2020) library for graph standardization. The experiment is conducted in a single-run on a PC with the Ubuntu 20.04, an AMD Ryzen 5900X 16-core CPU, and 64GB RAM. Our code[5] used Python 3.11.9 and `statsmodels` (Seabold and Perktold, 2010) for the binomial tests. We provide additional details in Appendix C.

4 Results and Discussion

Table 2 shows the results of soundness and binomial tests on 6 equivalent and 7 inequivalent cases. We present average values for each perturbation cases. In addition, overall $\min(\zeta)$ and $\max(\zeta)$ rows show the minimum/maximum score in equivalent/inequivalent cases, respectively. And, $P(\zeta = 1)$ rows refer to the proportion of items evaluated as equivalent in total. Note that a sound metric should yield statistically significant result on all the tests, without making the overlap between equivalent and inequivalent cases.

First, for the six equivalent cases, prior metrics failed to fully handle equivalent but altered graph structures. They only give ζ_{max} for 48-78% of graph pairs, as seen in the $P(\zeta = 1)$ row for equivalent cases. Specifically, metrics such as SEMBLEU, WWLK$_\theta$, WWLK$_{\text{e2n}}$, SEMA, SMATCH++ successfully gave the ζ_{max} to some equivalent cases (lift up, reordering, relabeling, duplicate). However, SMATCH and S^{2}MATCH failed to give ζ_{max} for above 4 cases. For example, SMATCH assigned an average score of 1.212 in duplicate cases, exceeding ζ_{max}. So, it suggests that SMATCH may overestimate similarity when a graph has multiple tautological edges. Additionally, following the soundness test, we attempted to verify whether the metric accurately assigns ζ_{max} to the completely identical case by using the original AMR as both the reference and hypothesis simultaneously. Surprisingly, SMATCH and SEMA failed to produce the maximum score ζ_{max} even for non-perturbed original cases, yielding scores as low as 0.902 (SMATCH) and 0.833 (SEMA). This result is likely due to the approximation methods employed by these metrics. Note that as SMATCH++ attempts to handle equivalent cases in their design, it shows a better evaluation for the de/reification case compared to other metrics.

Second, for the seven inequivalent cases, some metrics showed incorrect evaluation results by assigning ζ_{max} to certain graph pairs, as seen in the $P(\zeta = 1)$ row for the inequivalent cases. Specifically, SEMA produced a score of 1.103, exceeding the theoretical ζ_{max}. Moreover, among all the metrics, SEMA was the only one that achieved statistical significance in only 3 cases for the inequivalent cases. We suspect these results from SEMA appear to be numerical errors caused by its approximation algorithm. Furthermore, S^{2}MATCH assigned ζ_{max} to some edge deletion pairs (thus, $p > 0.05$), and SEMBLEU did the same for some edge insertion pairs ($p > 0.05$), while SMATCH, WWLK$_{\text{e2n}}$, and SMATCH++ passed all the tests, correctly identifying those cases as inequivalent.

Third, for the overlap between equivalent and inequivalent cases, all existing metrics showed overlap. For example, as SMATCH made overlap between equivalent and inequivalent cases on the interval $[0.371, 1]$, the score positioned in this range cannot be determined either equivalent or inequivalent. Similar overlap happens for SEMA (range of $[0.083, 1.103]$), WWLK$_\theta$ (range of $[0.656, 1]$), SMATCH++ (range of $[0.5, 0.998]$), and so on. Thus, we need to be careful in interpreting a score fall within the overlap range because there may exists incorrect evaluation in terms of AMR's theoretical background. Even the chance of falling in the overlap range is low, the existence of these overlapping sections is sufficient to pose a question about the soundness of existing metrics.

On the other hand, SMATCH$^\sharp$ has proven to be effective in dealing with all the 13 cases. SMATCH$^\sharp$ correctly assigned ζ_{max} in 100% of equivalence cases, achieving the highest possible score, which no other metric accomplished. Furthermore, SMATCH$^\sharp$ did not assign ζ_{max} for any inequivalence cases, as confirmed statistically. Specifically, for the six equivalent cases, SMATCH$^\sharp$ successfully provide ζ_{max}. For the seven inequivalent cases, SMATCH$^\sharp$ showed slight decrease in score compared to SMATCH++, the backbone of SMATCH$^\sharp$. For example, SMATCH++ had an average score of 0.927 for edge deletion case, while SMATCH$^\sharp$ scored an average of 0.907. Moreover, SMATCH$^\sharp$

[3]We provide the modified code at our GitHub repository.

[4]As the WWLK$_\theta$ and WWLK$_{\text{e2n}}$ are defined based on a different score range of $[-1, 1]$ compared to other metrics' range of $[0, 1]$, we normalized the range to $[0, 1]$.

[5]Code for the experiment will be uploaded in `https://github.com/snucclab/ssharp`.

	SMATCH$^\sharp$	SMATCH	S^{2}MATCH	SEMBLEU	WWLK$_\theta$	WWLK$_{e2n}$	SEMA	SMATCH++
6 Equivalent Cases					* Alternative hypothesis $H_A : P(\zeta = 1) > 99.9\%$			
Lift Up	1.000***	.964	.964	.999	1.000	1.000	.990	.949
Reorder	1.000***	.999	.998	1.000***	1.000***	1.000***	1.000	1.000*
Relabel	1.000***	.999	.999	1.000**	1.000***	1.000***	1.000	1.000***
Reify	1.000***	.748	.748	.613	.866	.864	.660	.990
Dereify	1.000***	.988	.988	.975	.994	.994	.990	.990
Duplicate	1.000***	1.212	.986	.470	.910	.937	.780	1.000***
Overall $\min(\zeta)$	1.000	.371	.371	.025	.610	.628	.083	.500
Overall $P(\zeta = 1)\%$	100	48.11	55.77	64.74	64.75	64.75	61.47	78.25
7 Inequivalent Cases					* Alternative hypothesis $H_A : P(\zeta = 1) < 0.1\%$			
Insert Node	.955***	.975***	.973***	.935***	.952***	.944***	.970***	.966***
Insert Edge	.964***	.983***	.980**	.931	.996***	.970***	.980***	.977***
Change Node	.944***	.966***	.961***	.871***	.940***	.951***	.890	.946***
Change Edge	.949***	.966***	.965	.935***	.982	.968***	.950	.952***
Delete Node	.906***	.945*	.946***	.908***	.936***	.933***	.940	.918***
Delete Edge	.907***	.948***	.949	.930***	.959***	.946***	.930***	.927***
Swap	.873***	.918***	.918***	.853	.949***	.954***	.880	.884***
Overall $\max(\zeta)$	.998	1.000	1.000	1.000	1.000	0.999	1.103	.998
Overall $P(\zeta = 1)\%$	.00	.01	.05	.65	5.09	.00	.63	.00

$^{+}p < 0.1$, $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$

Table 2: Result of soundness and binomial test on 13 simulated equivalent/inequivalent cases.

reduced the overlap range into zero, resolving the overlap issue that appeared in all existing metrics. This results suggest that SMATCH$^\sharp$ provides a better demarcation than existing metrics.

5 Conclusion

In this study, we proposed a novel experiment for verifying soundness of an AMR metric using simulated dataset and statistical tests. Through the experiment, our work demonstrated that the soundness problem exists in the previous metrics. Also, we suggest an AMR metric SMATCH$^\sharp$, which is an improved version of SMATCH++ in terms of soundness, using a graph standardization method that follows AMR guidelines. By testing SMATCH$^\sharp$ with the same experiment, we demonstrated that we can alleviate the issue by slightly enhancing the design of metrics. For future work, designing a new AMR similarity metric by considering our experimental results would be an interesting topic to pursue.

Limitations

In this section, we discuss the study's limitations that stem from our adoption of the AMR graph structure and experimental assumptions.

First, adopting the AMR graph structure, which is a standard meaning representation, provides a solid foundation for generating a score metric. However, because we adopted AMR, two limitations that affect our proposed approach also exist: the application of the metric on a single language, i.e. English, and the assumption of a single interpretation of the text.

Second, though we designed equivalence and inequivalence cases based on AMR specification, confirming whether we tested all theoretical variations of equivalence/inequivalence cases would be difficult. Therefore, it may be possible to present additional perturbations of AMR in future work.

Ethics Statement

In accordance with the guidelines of the ACL Ethics Policy, we will release all artifacts, including code, experiment results, and statistics used in this study on our GitHub repository. Also, because this study is an algorithmic consideration of model evaluation, we did not need a hyperparameter optimization process; thus, no such procedure is described. Moreover, due to the characteristics of AMR, a simulated dataset could be constructed without human annotation for equivalence/inequivalence conditions. Thus, we did not perform a human annotation process.

In addition, this study only concerns the evaluation of the output already generated by the model.

Therefore, as our study has no direct relationship to any sociocultural impacts or implications of machine learning models, such as social bias, we have not discussed these concerns.

Lastly, the AMR 2.0 (LDC2017T10) and 3.0 (LDC2020T02) datasets used in this study were purchased according to the license under the LDC User Agreement. Therefore, to create a simulated dataset according to our experimental procedure, a license would need to be purchased for the AMR 3.0 dataset. Furthermore, the LDC User Agreement prohibits the re-distribution of their datasets. For this reason, we can only provide the simulated dataset used in the experiment to parties with a valid license.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2020R1C1C1010162) and the Institute for Information & communications Technology Promotion (IITP) grant (No. 2021-0-02146), both funded by the Korean government (MSIT).

References

Rafael Torres Anchiêta, Marco Antonio Sobrevilla Cabezudo, and Thiago Alexandre Salgueiro Pardo. 2019. Sema: an extended semantic evaluation for amr. In *Proceedings of the 20th Computational Linguistics and Intelligent Text Processing*. Springer International Publishing.

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.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2019. Abstract Meaning Representation (AMR) 1.2.6 Specification.

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

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press.

Michael Wayne Goodman. 2019. AMR normalization for fairer evaluation. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information, and Computation*, pages 47–56, Hakodate.

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

James Higginbotham. 1985. On semantics. *Linguistic Inquiry*, 16(4):547–593.

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

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.

Juri Opitz, Angel Daza, and Anette Frank. 2021. Weisfeiler-leman in the bamboo: Novel AMR graph metrics and a benchmark for AMR graph similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Juri Opitz and Anette Frank. 2022. Better Smatch = better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 32–43, Online. Association for Computational Linguistics.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR Similarity Metrics from Principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.

Sameer S Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. In *International Conference on Semantic Computing (ICSC 2007)*, pages 517–526. IEEE.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

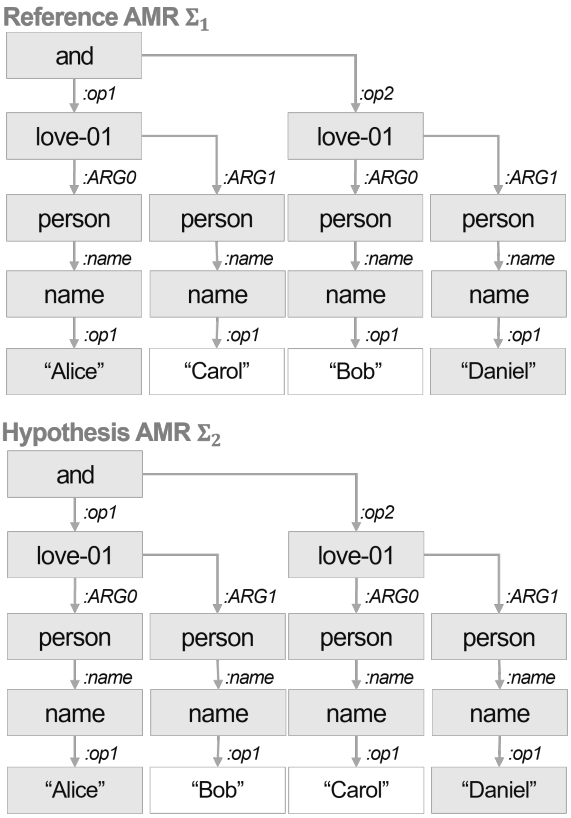

Figure 1: An example case that related to soundness

A Sample Case

Figure 1 illustrates two AMR graphs with different meanings; persons in two different 'love' relations are swapped. The reference AMR graph means a sentence "*Alice* loves *Carol*, and *Bob* loves *Daniel*." But, the hypothesis AMR graph means a sentence "*Alice* loves *Bob*, and *Carol* loves *Daniel*." Thus, these two graphs do not have the same truth condition since *Bob* and *Carol* are different people in general. Therefore, its expected value should not be the theoretical maximum score that corresponds to equivalence. Furthermore, since the subject of 'love' is set differently in both AMRs, it also should not be receive a score that is nearly identical to the theoretical maximum.

So, we computed the similarity between these two graphs using existing metrics. All existing metrics produced a score close to 1: SEMBLEU and WWLK_{e2n} assigned 1.0 and SMATCH and S^2MATCH assigned 0.9231. Specifically for SEMBLEU, we suspect that the maximum length of n-grams used in SEMBLEU is not sufficient to handle this case; official SEMBLEU use 3-grams, which is shorter than the distance between 'love-01' and a person's name, e.g., 'Carol.' In contrast, SMATCH$^\sharp$ assigned a value of 0.8889 for this case, which is lowest score among the metrics.

B Graph Transformation

- Original AMR:

  ID: `DF-199-192794-660_6610.5`

  Sentence: I never missed a day of school.

```
(m / miss-02
  :ARG0 (i / i)
  :ARG1 (t / temporal-quantity
    :unit (d / day)
    :quant 1
    :duration-of (s / school-01))
  :polarity -
  :time (e / ever))
```

- Equivalence Cases:

  Lift Up randomly set other node as a root. According to AMR guidelines, AMR can also be viewed as conjunction of logical triples, omitting root information. Thus, changing root does not harm AMR's truth condition.

```
(t / temporal-quantity
   :ARG1-of (m / miss-02
      :polarity -
      :time (e / ever)
      :ARG0 (i / I))
   :duration-of (s / school-01)
   :quant 1
   :unit (d / day))
```

Reorder randomly changes the displaying order of a graph.

```
(m / miss-02
   :time (e / ever)
   :polarity -
   :ARG0 (i / i)
   :ARG1 (t / temporal-quantity
      :quant 1
      :unit (d / day)
      :duration-of (s / school-01)))
```

Relabel change the head of each node.

```
(r0 / miss-02
  :ARG0 (r1 / i)
  :ARG1 (r2 / temporal-quantity
    :duration-of (r3 / school-01))
    :quant 1
    :unit (r4 / day)
  :polarity -
  :time (r5 / ever))
```

Reify / Dereify According to AMR guidelines, apply Reification/Dereification using PENMAN library.

```
(m / miss-02
  :ARG0 (i / I)
  :ARG1 (t / temporal-quantity
    :ARG2-of (_ / last-01
      :ARG1 (s / school-01))
    :ARG1-of (_2 / have-quant-91
      :ARG2 1)
      :unit (d / day))
  :ARG1-of (_3 / have-polarity-91
    :ARG2 -)
  :ARG1-of (_4 / be-temporally-at-91
    :ARG2 (e / ever)))
```

Duplicate Randomly duplicate the graph component.

```
(m / miss-02
   :ARG0 (i / i)
   :ARG0 i
   :ARG1 (t / temporal-quantity
      :duration-of (s / school-01)
      :duration-of s
      :quant 1
      :quant 1
      :unit (d / day)
      :unit d)
    :ARG1 t
    :polarity -
    :polarity -
    :time (e / ever)
    :time e)
```

Note that the motivation for duplicating edges is that we suspected that score inflation may have occurred in existing metrics when duplication occurred in existing parsers. Indeed, the experiment was useful in that it revealed problems with SMATCH. As a result of the experiment, SMATCH showed a tendency to evaluate higher than the score limit (0-1) when such cases were introduced. This implies the possibility that score inflation may have occurred when using SMATCH to evaluate when duplicates occurred in existing parsers.

C Implementation Detail

- Hardware:

  CPU: AMD Ryzen 5900X
  Memory: 64GB

- Software:

  OS: Ubuntu 20.04.6 LTS (kernel 5.4.0-169)
  Python: 3.11.9 (with virtualenv)

- Python libraries:

  `Penman` 1.3.0
  `networkx` 3.3
  `numpy` 1.26.4
  `statsmodels` 0.13.5
  `pandas` 2.2.1
  `SciPy` 1.12.0