Referenceless Parsing-Based Evaluation of AMR-to-English Generation

Emma Manning **Nathan Schneider**
Georgetown University
{esm76, nathan.schneider}@georgetown.edu

Abstract

Reference-based automatic evaluation metrics are notoriously limited for NLG due to their inability to fully capture the range of possible outputs. We examine a referenceless alternative: evaluating the adequacy of English sentences generated from Abstract Meaning Representation (AMR) graphs by parsing into AMR and comparing the parse directly to the input. We find that the errors introduced by automatic AMR parsing substantially limit the effectiveness of this approach, but a manual editing study indicates that as parsing improves, parsing-based evaluation has the potential to outperform most reference-based metrics.

1 Introduction

Natural language generation (NLG) is notoriously difficult to evaluate well due to its one-to-many nature: thanks to the infinite capacity of human language, any given meaning can be expressed in a potentially unlimited number of ways. Thus, listing the 'right answer(s)' and comparing a system's output against such a list, which is a possible evaluation method for many other tasks, is fundamentally limited for NLG. Nevertheless, automatic evaluation of NLG has traditionally been dominated by reference-based metrics like BLEU (Papineni et al., 2002).

In recent years, however, referenceless evaluation metrics have been gaining popularity in NLG and related fields. In this paper we examine the potential and limitations of one such approach: using semantic parsing to compare a generated sentence to a meaning representation from which it was generated, in order to measure semantic adequacy. We focus on generation of English text from Abstract Meaning Representation graphs ("AMRs"; Banarescu et al., 2013). Figure 1 shows an example of an AMR, which represents the meaning of a sentence. AMR does not represent certain morphological and syntactic details such as tense, number,

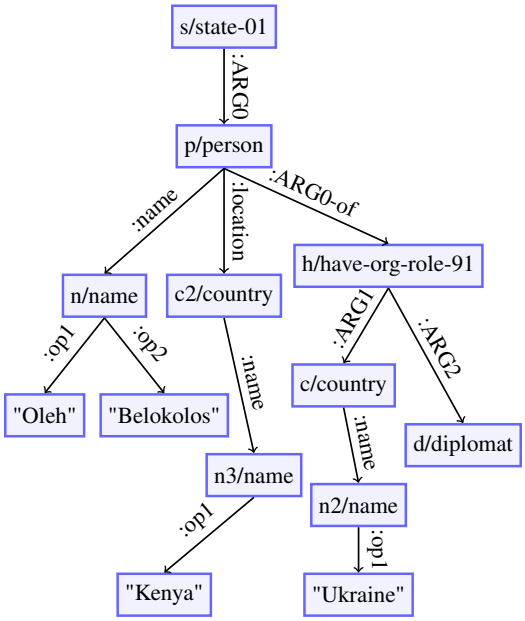

Figure 1: AMR graph for the sentence "Ukrainian diplomat in Kenya oleh belokolos stated –"

definiteness, and word order, so the graph shown could represent a number of alternate sentences, such as:

- Oleh Belokolos, the Ukrainian diplomat in Kenya, states:
- A Ukrainian diplomat in Kenya named Oleh Belokolos has made a statement.

Ideally, then, a sentence generated from an AMR graph should be judged on how well it expresses the elements of meaning given in the graph, ignoring the details that are not included.

We examine the hypothesis that we can measure the semantic adequacy of a sentence *generated* from an AMR by performing the reverse operation—namely, *parsing* the generated sentence into AMR—and measuring the similarity of the parsed AMR graph to the original. In essence, this idea exploits complementarity of English-to-AMR parsers and AMR-to-English generators being evaluated. Assuming an accurate parse, we would ex-

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 114–122

https://doi.org/10.26615/978-954-452-056-4_012

Var	Description
r	Reference sentence
a	Gold AMR, created from r
g	Sentence automatically generated from a
p	AMR automatically parsed from g
p'	Manually-corrected version of automatic parses p

Table 1: Summary of notation used in this paper, with a description of each type of sentence and AMR used.

pect this to be a good measure of the adequacy of the generated sentence, since a sentence that accurately expresses the meaning in the original AMR should have the same AMR. We further formalize this approach in §2. As discussed in §3, this method has also been suggested by Opitz and Frank (2021); we contribute new analyses of its validity, in particular by measuring its correlation with human adequacy judgments collected by Manning et al. (2020). We find that errors made by an automatic AMR parser substantially limit the quality of parsing-based evaluation as a proxy for human evaluation, resulting in a lower correlation with adequacy scores than many reference-based metrics (§5). To approximate an upper bound for the potential of this evaluation approach with improved parsing, we conduct an additional study using manually-corrected AMR parses; we find that this substantially improves the quality of the metric (§6).

2 Parsing-Based Evaluation

AMR-to-English generation is the task of taking an input AMR graph a and generating a sentence g expressing the meaning content of the AMR in English. Ideally, we would evaluate generation by comparing g directly to a to determine how well g expresses the meaning in a; this is what the human annotators whose judgments we use (see §4) did. However, we don't know of an existing way to directly compare a sentence to an AMR graph; instead, our metrics tend to compare two items of the same type. Reference-based metrics compare the generated sentence g to r, the English sentence for which the AMR was created, and which is typically used as the sole reference in evaluation. We analyze the hypothesis that we can more accurately capture the details relevant to AMR by comparing AMRs to each other: specifically, comparing a to either p, an automatic parse of g, or to p', a manually-corrected version of p. Our notation is summarized in table 1.

3 Background

The evaluation method we analyze in this paper is closely related to the $\mathcal{MF}_\beta$ metric suggested by Opitz and Frank (2021), which combines a measure of meaning preservation, $\mathcal{M}$, with a language model–based measure of grammatical form, $\mathcal{F}$. Their meaning preservation metric, $\mathcal{M}$, assigns a score to a generated sentence by parsing it into AMR and computing the parse's similarity to the gold AMR. They use the AMR parser by Cai and Lam (2020) and the S^2match similarity metric; we experiment with these as well as other options for both parser and metric. While they perform a number of pilot experiments to test the robustness of $\mathcal{MF}_\beta$, such as its performance with different parsers, Opitz and Frank do not test the correlation of their metric with human judgments; thus, the work presented here adds to our understanding of the validity of this type of metric as a proxy for human evaluation.

As a baseline, we also compare the results of parsing-based evaluation with several reference-based metrics, including those that have traditionally been used to evaluate AMR generation as well as newer metrics that have shown promising results for NLG.

4 Data

We use human judgment data from Manning et al. (2020), consisting of judgments on a total of 600 sentences: 100 human-authored reference sentences and their corresponding AMRs, and 500 sentences automatically generated from these AMRs by 5 different systems. When comparing to reference-based automatic metrics, we do not score the reference sentences themselves since, with only one reference per AMR, these would trivially receive perfect scores on such metrics.

Each sentence has a score on a scale of 0–100, for each of fluency and adequacy, averaged over two annotators. We compare automatic metrics to these judgments, and particularly to the adequacy judgments, since we are primarily interested in parsing-based evaluation as a proxy for adequacy evaluation.

Annotators additionally provided binary judgments on whether information was added or omitted, and whether the sentence was incomprehensible; we use the latter of these judgments in §6 to determine which generated sentences to manually edit the parses of.

5 Experiment 1: Automatic Metrics

This section describes experiments with variations on the automatic version of the parsing based metric; that is, the use of similarity metrics comparing the automatic parse p to the gold AMR a. We experiment with different AMR parsers (§5.1) and variations on the Smatch similarity metric (§5.2) and measure the correlation to human judgments of adequacy (§5.3).

5.1 Parsers

We compare gold AMRs to AMR parses of the generated sentences. This includes using three different automatic English-to-AMR parsers, described below.

JAMR. The JAMR parser[1] (Flanigan et al., 2014, 2016) is an early AMR parser; we use it as a baseline to compare against the more recent, higher-accuracy parsers. The JAMR parser uses a semi-Markov model to identify concepts, followed by a graph variant on Maximum Spanning Tree algorithms to identify the relations between concepts. We used the 2016 version, which achieved a Smatch score of 67 on the LDC2015E86 dataset.

LYU-TITOV. While most AMR parsers first train an aligner to align AMR nodes with words in a sentence prior to training the parser itself, Lyu and Titov (2018)[2] treat alignments as latent variables in a joint probabilistic model for identifying concepts, relations, and alignments. This parser achieved a Smatch score of 73.7 on LDC2015E86 and 74.4 on LDC2016E25, which at the time was state-of-the-art.

CAI-LAM. Cai and Lam (2020)[3] was the state of the art in AMR parsing as of 2020, with a Smatch score of 80.2 on LDC2017T10. This transformer-based parser uses iterative inference to determine which part of the input sentence to parse and where to add it to the output graph, without requiring explicit alignments.

Parser performance. We evaluate each parser's accuracy on our sample of 100 sentences by computing $Smatch(a, p(r))$, i.e., the similarity between the gold AMRs in the sample and their corresponding parsed references. We find that CAI-LAM performs the best with a Smatch score of 84.9, followed by 76.3 for LYU-TITOV and 71.1 for JAMR.

[1]Code: https://github.com/jflanigan/jamr

[2]Code: https://github.com/ChunchuanLv/AMR_AS_GRAPH_PREDICTION

[3]Code: https://github.com/jcyk/AMR-gs

5.2 Similarity Metrics

The second piece needed for parsing-based evaluation is a way to quantify the similarity of an AMR parse to the original AMR.

The standard metric for comparing two AMRs—such as to evaluate the quality of an AMR parser or inter-annotator-agreement between human parses—is Smatch (Cai and Knight, 2013). The Smatch score compares triples between two AMR graphs, where each triple is an edge of the graph (a semantic relationship) combined with each of the nodes it connects. For a given pair of AMRs, the Smatch score is the maximum F1-score of triples which can be obtained with a one-to-one mapping of variables between the two graphs.[4]

We also experimented with a small variation on the original Smatch. Smatch computes the similarity between two different AMR graphs based on inferred alignments between the two graphs' concepts. Since checking all possible mappings is computationally intractable, it starts with one 'smart' initialization, then retries with random initializations; the default is four random restarts. This means that Smatch scores are nondeterministic; when running twice on the same pair of AMRs, we sometimes got different scores. To mitigate this effect, we made two changes: First, we increased the number of restarts to 100 to increase the chance that the best mapping would be found, while still maintaining a reasonable runtime. Second, we seeded the random function in the Smatch script to make the results reproducible. In table 3, we refer to the default Smatch as 'Smatch_4', while the variation with a seed and 100 restarts is 'Smatch_{100}+seed'.

More recently, Opitz et al. (2020) analyzed both Smatch and an alternative metric, SemBleu (Song and Gildea, 2019), and proposed a new variant of Smatch, S^2match, which conforms to desirable principles better than either previous metric. In particular, S^2match introduces the concept of embedding-based semantic gradable semantic similarity by allowing for soft matches between concepts. While the primary advantage of this variant is for tasks with more variation in wording, such as measuring the similarity of paraphrases, it could also be advantageous in our setting—for example, to penalize AMR generation systems that represent a concept with the wrong word less if it is a se-

[4]We compute Smatch using the `smatch.py` script found at https://github.com/jflanigan/jamr/tree/Semeval-2016/scripts/smatch_2.0.2.

	Fluency	Adequacy
BLEU$_\uparrow$	0.40	0.52
METEOR$_\uparrow$	0.41	0.57
TER$_\downarrow$	−0.33	−0.43
CHRF++$_\uparrow$	0.32	0.47
BERTScore$_\uparrow$	0.47	0.60
BLEURT$_\uparrow$	**0.60**	**0.69**

Table 2: Sentence-level correlations with human judgments for **reference-based** metrics.

mantically related one, or to mitigate the effects of certain parser errors. Thus, we also experiment with computing the S^{2}match similarity of parsed sentences to the original AMRs.[5]

5.3 Results

The primary statistic of interest for this study is the sentence-level correlation between a proposed metric and human judgments, particularly those for adequacy. We measure this with Spearman's Rho correlation. Following Manning et al. (2020), we compare several popular reference-based metrics; table 2 reports the correlations for the 5 metrics they used: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006), ChrF++ (Popović, 2017), and BERTScore (Zhang et al., 2020). We add the results of one newer metric, BLEURT (Sellam et al., 2020). Of these, BLEURT performs the best by this measure with a correlation of 0.69. BLEU, the most popular metric for this task, has a correlation of 0.52.

Table 3 shows the correlation with adequacy for each variant of the parser-based metric, combining the three AMR parsers and three similarity metrics used. Notably, even the highest correlations here underperform those achieved by BLEU, METEOR, BERTScore, and BLEURT.

As expected, the correlation increases with parser quality, indicating that parsers that have higher accuracy on human-authored sentences also do better with generated sentences.

For each parser, there is very little difference between the different similarity metrics. The similarity between Smatch$_4$ and Smatch$_{100}$+seed is expected, since these are separated only by minor implementation differences. The lack of substantial improvement when using S^{2}match is probably because it is rare for the generated sentences to contain concepts that are different but semantically similar to those in the gold AMR.

[5]We calculate S^{2}match using https://github.com/Heidelberg-NLP/amr-metric-suite.

	Smatch$_4$	Smatch$_{100}$+seed	S^{2}match
JAMR	0.358	0.356	0.362
LYU-TITOV	0.462	0.460	0.465
CAI-LAM	0.495	0.492	0.494

Table 3: Sentence-level correlations with human judgments for **parsing-based** metrics, with different choices of parser and similarity metric.

Since none of the similarity metrics are clearly stronger than the others based on correlations, we choose Smatch$_{100}$+seed as the best for more conceptual reasons: it is more reproducible, and unlike S^{2}match, does not rely on embeddings. The use of additional resources seems unjustified in this case if it does not improve performance, especially given concerns that using embeddings in an evaluation metric makes it less transparent and more arbitrary (results can vary depending on specific choice of language model) than a simpler method.

Thus, for the following experiments, we use the CAI-LAM parser combined with Smatch$_{100}$+seed.

6 Experiment 2: Manually-Edited Parses

Even a state-of-the-art AMR parser is of course not perfect, and may struggle more with parsing automatically-generated sentences than the human-authored ones it is designed for. The potential for parser error is a major limitation of the proposed approach; evaluating the parse p against gold AMR a can only be a good measure of g's relationship to a if p is a sufficiently accurate parse of g. Thus, to get a better sense of the effect that parsing errors can have on this metric even when using a SOTA parser, and of a rough upper bound for how well the metric could work in the future as parsing improves, we also manually edited a sample of the parses p to create alternate parses, p', which better reflect the meaning expressed in the generated sentences g, and use Smatch to compare p' to a.

6.1 Methods

Since the CAI-LAM parser is the strongest automatic parser, we used its parses as a starting point. For a given generated sentence g or reference sentence r, we compared the sentence to the automatic parse $p(g)$ or $p(r)$, and edited the parse to represent, as accurately as possible, the meaning expressed in the sentence. This sometimes included referring to the gold AMR a to ensure consistency between our annotations and the canonical representation of the same meanings. All edits were performed by the first author.

System	Edited Parses
Guo	65
Konstas	83
Manning	25
Ribeiro	70
Zhu	73
Reference	90
Total	406

Table 4: Number of sentences (out of 100) from each generation system that were not marked as incomprehensible by either annotator, and whose AMRs were manually edited.

However, this approach is limited by an assumption that the generated sentences *have* meanings in the same way that human-authored sentences do. In fact, many of the generated sentences in this dataset do not clearly and unambiguously express a particular meaning. Since it is essentially impossible to 'accurately' parse an incoherent sentence, we only edited the parses of sentences which were not marked as incomprehensible by either annotator in the human evaluation. Table 4 shows how many sentences from each system fit this criterion. Overall, we edited parses for 406 sentences, or 67.7% of the total sample of 600 sentences used in the human evaluation. Excluding references, we edited parses for 316 of the 500 automatically-generated sentences, or 63.2%. For the remaining sentences, we use the unedited automatic parse.

Even after filtering out those marked incomprehensible, we encountered many sentences that we found unclear or highly ambiguous; perhaps there were so many unclear sentences in the data that annotators reserved the annotation only for the most egregious cases. We did our best to interpret these sentences as well as we could, erring on the side of preserving the automatic parse's interpretation when it seemed as reasonable as an alternative. Nevertheless, this required some subjective judgment calls. An example of a difficult-to-annotate case is shown in table 5. The generated sentence, "ukraine and ukraine in kenya stated –", would probably never be produced by a human author, and it is difficult to assign a precise meaning to it. In this case, we decided to preserve the automatic parser's interpretation that it describes a statement being made by two entities: the country Ukraine, and a separate location, also known as Ukraine, that is in Kenya.

r	Ukrainian diplomat in Kenya oleh belokolos stated –
g	ukraine and ukraine in kenya stated –

p	`(c0 / state-01`
	`  :ARG0 (c1 / and`
	`  :op1 (c2 / ukraine)`
	`  :op2 (c3 / ukraine`
	`  :location (c4 / country`
	`  :name (c5 / name`
	`  :op1 "Kenya")`
	`  :wiki "Kenya"))))`

p'	`(c0 / state-01`
	`  :ARG0 (c1 / and`
	`  :op1 (c2 / country`
	`  :wiki "Ukraine"`
	`  :name (n2 / name`
	`  :op1 "Ukraine"))`
	`  :op2 (c3 / location`
	`  :name (n3 / name`
	`  :op1 "Ukraine")`
	`  :location (c4 / country`
	`  :name (c5 / name`
	`  :op1 "Kenya")`
	`  :wiki "Kenya"))))`

Table 5: An example of a generated sentence with unclear meaning.

	Full Sample	INC0
Automatic Parses	0.49	0.35
Edited Parses	**0.66**	**0.46**

Table 6: Sentence-level Spearman's correlation of Smatch with human adequacy scores, when using edited parses vs. automatic ones. INC0 indicates the subset of AMRs that were edited.

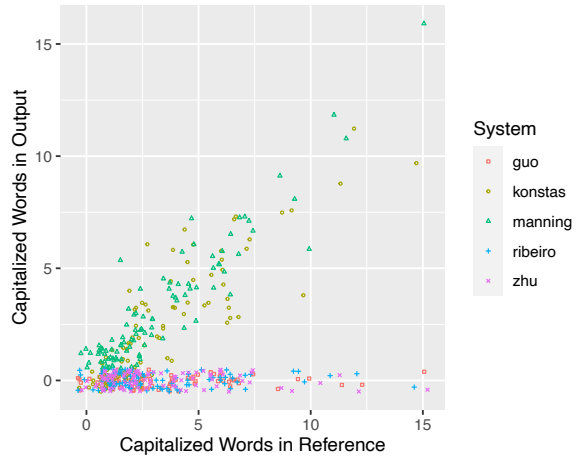

Figure 2: Scatterplot of number of capitalized words in the output compared to the reference for each system (jitter=0.5).

6.2 Results

As table 6 shows, the correlation of Smatch with adequacy improves substantially when using the edited parses, as opposed to the purely automatic ones. With edits, the correlation over all data increases to 0.66, better than most of the automatic metrics—the only exception is BLEURT, with a correlation of 0.69 (see table 2). It seemed possible that this improvement occurred simply because the edited sample of sentences, which generally received stronger human scores, largely had their Smatch scores improved by the editing process. Thus we also include the correlations on only the edited sample (INC0); the fact that the correlation improves within this sample demonstrates that editing does help distinguish better and worse sentences.

Table 7 shows an example where editing helped substantially. The generated sentence fully expresses the information in the gold AMR, and received fluency and adequacy scores of 100—in fact, it differs from the reference only in capitalization—but the automatic parse differs greatly from the gold AMR, resulting in a low Smatch score of 0.222. Parser errors in this case include a failure to recognize the two named entities in the sentence, as well as misidentifying the root concept as `be-located-at-91` rather than `organization`. While the edited parse doesn't perfectly match the gold AMR, it corrects these major errors, resulting in a much higher Smatch score of 0.875.

r	The Institute for Science and International Security is a private research organization located in Washington.
g	the institute for science and international security is a private research organization located in washington .

a	`(o / organization`
	`:mod (r / research-01)`
	`:ARG1-of (p / private-03)`
	`:domain (o2 / organization`
	`:wiki "Institute_for_Science_and_`
	`International_Security"`
	`:name (n / name`
	`:op1 "Institute" :op2 "for"`
	`:op3 "Science" :op4 "and"`
	`:op5 "International"`
	`:op6 "Security"))`
	`:ARG1-of (l / locate-01`
	`:location (c / city`
	`:wiki "Washington,_D.C."`
	`:name (n2 / name`
	`:op1 "Washington"))))`

p	`(c0 / be-located-at-91`
	`:ARG1 (c1 / institute`
	`:mod (c3 / organization`
	`:ARG1-of (c6 / private-03)`
	`:mod (c5 / research-01))`
	`:topic (c4 / and`
	`:op1 (c7 / science)`
	`:op2 (c8 / security`
	`:mod (c9 / international))))`
	`:ARG2 (c2 / washington))`

p'	`(c0 / organization`
	`:domain (c4 / organization`
	`:name (c6 / name`
	`:op1 "Institute" :op2 "for"`
	`:op3 "Science" :op4 "and"`
	`:op5 "International"`
	`:op6 "Security")`
	`:wiki "Institute_for_Science_and_`
	`International_Security")`
	`:location (c3 / city`
	`:name (c5 / name`
	`:op1 "Washington")`
	`:wiki "Washington,_D.C.")`
	`:mod (c1 / private-03)`
	`:mod (c2 / research-01))`

Table 7: An example where parser error led to a low Smatch score on a high-adequacy sentence, which is improved substantially in the edited parse.

r	A US-endorsed package of incentives to cease enriched uranium production
g	the us endorsed package of incentives to cease enriched uranium production .
a	(p / package :consist-of (t / thing :ARG0-of (i / incentivize-01 :ARG2 (c2 / cease-01 :ARG1 (p2 / produce-01 :ARG1 (u / uranium :ARG1-of (e2 / enrich-01)))))) :ARG1-of (e / endorse-01 :ARG0 (c / country :wiki "United_States" :name (n / name :op1 "US"))))
p	(c0 / endorse-01 :ARG0 (c2 / we) :ARG1 (c1 / package-01 :ARG1 (c3 / incentivize-01 :ARG2 (c4 / cease-01 :ARG1 (c5 / produce-01 :ARG1 (c6 / uranium) :ARG1-of (c7 / enrich-01))))))
p'	(c0 / endorse-01 :ARG0 (c2 / country :name (c5 / name :op1 "US") :wiki "United_States") :ARG1 (c1 / package-01 :ARG1 (c3 / incentivize-01 :ARG2 (c4 / cease-01 :ARG1 (p1 / produce-01 :ARG1 (c6 / uranium :ARG1-of (c7 / enrich-01)))))))

Table 8: An example of a parser error due to lack of capitalization in the generated sentence. 'US', written as 'us' by the system, is treated as a form of the pronoun 'we' by the parser.

System	Unedited ρ	Edited ρ	Improvement
Konstas	0.53	0.59	0.05
Manning	0.44	0.57	0.12
Guo	0.44	0.59	0.14
Ribeiro	0.53	0.72	0.19
Zhu	0.35	0.59	0.24
Overall	0.49	0.66	0.17

Table 9: Spearman's correlation of adequacy scores with Smatch scores based on unedited and edited parses. The two systems that produce capitalization are shown above the line; the three below output only lowercase.

7 Analysis

A common parser error was failure to recognize named entities when they were not capitalized; examples of this are given in tables 7 and 8. As figure 2 shows, three of the systems never produce capitals in their output, while those of Konstas and Manning typically produce about as many capitals as are present in the reference. Thus, it seems likely that the systems that never produce capitals may be unfairly penalized by a parsing-based metric.

Table 9 shows that when separating the data by system, there is no clear difference in the degree to which Smatch correlates with adequacy for systems that capitalize compared to those that do not. However, the *difference* between $Smatch(a, p')$ and $Smatch(a, p)$ is greater for the systems that do not produce capitals; that is, manual editing had a greater effect on the reliability of the parser-based metric on the systems which do not produce capitals than those that do.

It may be possible to overcome this particular limitation of the automatic parser by adding a preprocessing step that recognizes and capitalizes named entities, or by training the parser on more all-lowercase examples.

8 Conclusion

In this paper, we have explored the idea of evaluating AMR generation via AMR parsing and similarity metrics, using the human judgments of adequacy collected by Manning et al. (2020) to test the validity of possible variants of the parsing-based metric approach and compare them to existing reference-based metrics. We found that parser quality is a major factor affecting the performance of this evaluation approach: the better the AMR parser, the better the evaluation; however, even a state-of-the-art parser with an accuracy of 80+% on standard human-authored data has significant

limitations for evaluating generated sentences, including a failure to recognize named entities in the absence of capitalization. We showed that when automatic AMR parses are manually edited to better reflect the meaning in generated sentences, this referenceless metric outperforms most popular automatic reference-based metrics, including BLEU and BERTScore (but not BLEURT).

While the current reliance on manual editing for more reliable results may not be practical for evaluation, the results of this experiment indicate that fully-automatic parser-based metrics are likely to prove more reliable in the future as the state of the art in AMR parsing continues to improve, especially if newer AMR generation systems also more closely replicate human-authored data, such as by producing more human-like capitalization than the majority of systems tested here did.

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.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

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.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. CMU at SemEval-2016 task 8: Graph-based AMR parsing with infinite ramp loss. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1202–1206, San Diego, California. Association for Computational Linguistics.

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)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407, Melbourne, Australia. Association for Computational Linguistics.

Emma Manning, Shira Wein, and Nathan Schneider. 2020. A human evaluation of AMR-to-English generation systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4773–4786, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *ICLR*.