SEMBLEU: A Robust Metric for AMR Parsing Evaluation

Linfeng Song and **Daniel Gildea**
Department of Computer Science
University of Rochester
Rochester, NY 14627

Abstract

Evaluating AMR parsing accuracy involves comparing pairs of AMR graphs. The major evaluation metric, SMATCH (Cai and Knight, 2013), searches for one-to-one mappings between the nodes of two AMRs with a greedy hill-climbing algorithm, which leads to search errors. We propose SEMBLEU, a robust metric that extends BLEU (Papineni et al., 2002) to AMRs. It does not suffer from search errors and considers non-local correspondences in addition to local ones. SEMBLEU is fully content-driven and punishes situations where a system's output does not preserve most information from the input. Preliminary experiments on both sentence and corpus levels show that SEMBLEU has slightly higher consistency with human judgments than SMATCH. Our code is available at http://github.com/freesunshine0316/sembleu.

1 Introduction

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a semantic formalism where the meaning of a sentence is encoded as a rooted, directed graph. Figure 1 shows two AMR graphs in which the nodes (such as "girl" and "leave-11") represent AMR concepts and the edges (such as "ARG0" and "ARG1") represent relations between the concepts. The task of parsing sentences into AMRs has received increasing attention, due to the demand for better natural language understanding.

Despite the large amount of work on AMR parsing (Flanigan et al., 2014; Artzi et al., 2015; Pust et al., 2015; Peng et al., 2015; Buys and Blunsom, 2017; Konstas et al., 2017; Wang and Xue, 2017; Ballesteros and Al-Onaizan, 2017; Lyu and Titov, 2018; Peng et al., 2018; Groschwitz et al., 2018; Guo and Lu, 2018), little attention has been paid to evaluating the parsing results, leaving SMATCH

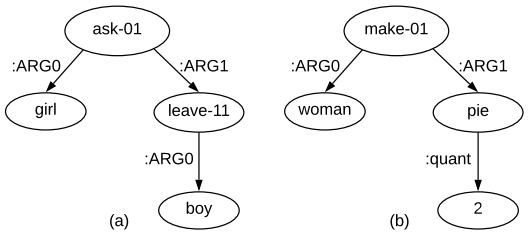

Figure 1: Two AMR examples meaning "The girl asked the boy to leave." and "The woman made two pies.", respectively. Their SMATCH score is 25 (%).

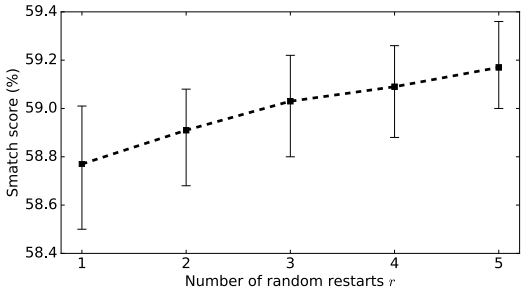

Figure 2: Average, minimal and maximal SMATCH scores over 100 runs on 100 sentences. The running time increases from 6.6 seconds (r=1) to 21.0 ($r = 4$).

(Cai and Knight, 2013) as the only overall performance metric. Damonte et al. (2017) developed a suite of fine-grained performance measures based on the node mappings of SMATCH (see below).

SMATCH suffers from two major drawbacks: first, it is based on greedy hill-climbing to find a one-to-one node mapping between two AMRs (finding the exact best mapping is NP-complete). The search errors weaken its robustness as a metric. To enhance robustness, the hill-climbing search is executed multiple times with random restarts. This decreases efficiency and, more importantly, does not eliminate search errors. Figure 2 shows the means and error bounds of SMATCH scores as a function of the number of restarts r over 100 runs on 100 sentences. We can see that the variances are still significant when r is

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4547–4552
Florence, Italy, July 28 - August 2, 2019.

large. Furthermore, by corresponding with other researchers, we have learned that previous papers on AMR parsing report SMATCH scores using differing values of r.

Another problem is that SMATCH maps one node to another regardless of their actual content, and it only considers edge labels when comparing two edges. As a result, two different edges, such as "ask-01 :ARG1 leave-11" and "make-01 :ARG1 pie" in Figure 1, can be considered identical by SMATCH. This can lead to a overly large score for two completely different AMRs. As shown in Figure 1, SMATCH gives a score of 25% for the two AMRs meaning "The girl asked the boy to leave" and "The woman made two pies", which convey obviously different meanings.[1] The situation could be worse for two different AMRs with few types of edge labels, where the score could reach 50% if all pairs of edges between them were accidentally matched.

To tackle the problems above, we introduce SEMBLEU, an accurate metric for comparing AMR graphs. SEMBLEU extends BLEU (Papineni et al., 2002), which has been shown to be effective for evaluating a wide range of text generation tasks, such as machine translation and data-to-text generation. In general, a BLEU score is a precision score calculated by comparing the n-grams (n is up to 4) of a predicted sentence to those of a reference sentence. To punish very short predictions, it is multiplied by a brevity penalty, which is less than 1.0 for a prediction shorter than its reference. To adapt BLEU for comparing AMRs, we treat each AMR node (such as "ask-01") as a *unigram*, and we take each pair of directly connected AMR nodes with their relation (such as "ask-01 :ARG0 girl") as a *bigram*. Higher-order n-grams (such as "ask-01 :ARG1 leave-11 :ARG0 boy") are defined in a similar way.

SEMBLEU has several advantages over SMATCH. First, it gives an exact score for each pair of AMRs without search errors. Second, it is very efficient to calculate. On a dataset of 1368 pairs of AMRs, SEMBLEU takes 0.5 seconds, while SMATCH takes almost 2 minutes using the same machine. Third, it captures high-order relations in addition to node-to-node and edge-to-edge correspondences. This gives complementary judgments to the standard SMATCH metric for evaluating AMR parsing quality. Last, it does not give overly large credit to AMRs that represent completely different meanings.

Our initial evaluations suggest that SEMBLEU has higher consistency with human judgments than SMATCH on both corpus-level and sentence-level evaluations. We also show that the number of n-grams extracted by SEMBLEU is roughly linear in the AMR scale. Evaluation on the outputs of several recent models show that SEMBLEU is mostly consistent with SMATCH for results ranking, but with occasional disagreements.

2 Our metric

Our method is based on BLEU, which we briefly introduce, before showing how to extend it for matching AMR graphs.

2.1 Preliminary knowledge on BLEU

As shown in Equation 1, the BLEU score for candidate $\boldsymbol{c}$ and reference $\boldsymbol{z}$ is calculated by multiplying a modified precision with a brevity penalty (BP).

$$\text{BLEU} = BP \cdot \exp\left(\sum_{k=1}^{n} w_k \log p_k\right) \quad (1)$$

BP is defined as $e^{\min\{1-\frac{|\boldsymbol{z}|}{|\boldsymbol{c}|},0\}}$, which gives a value of less than 1.0 when the candidate length ($|\boldsymbol{c}|$) is smaller than the reference length ($|\boldsymbol{z}|$). p_k and w_k are the precision and weight for matching k-grams, and p_k is defined as

$$p_k = \frac{|kgram(\boldsymbol{z}) \cap kgram(\boldsymbol{c})|}{|kgram(\boldsymbol{c})|}, \quad (2)$$

where $kgram$ is the function for extracting all k-grams from its input.

2.2 SEMBLEU

To introduce SEMBLEU, we make the following changes to adapt BLEU to AMR graphs. First, we define the size of each AMR (G) as the number of nodes plus the number of edges: $|G| = |G.nodes| + |G.edges|$. This size is used to calculate the brevity penalty (BP). Intuitively, edges carry important relational information. Also, we observed many situations where a system-generated AMR preserves most of the concepts in the reference, but misses many edges.

Another change is to the n-gram extraction function ($kgram$ in Equation 2). AMRs are directed acyclic graphs, thus we start extracting n-grams from the roots. This is analogous to starting

[1] https://amr.isi.edu/eval/smatch/compare.html gives more details.

Fg	n	Extracted n-grams
(a)	1	ask-01; girl; leave-11; boy
	2	ask-01 :ARG0 girl; ask-01 :ARG1 leave-11; leave-11 :ARG0 boy;
	3	ask-01 :ARG1 leave-11 :ARG0 boy;
(b)	1	woman; make-01; pie; 2
	2	make-01 :ARG0 woman; make-01 :ARG1 pie; pie :quant 2;
	3	make-01 :ARG1 pie :quant 2;

Table 1: n-grams (separated by ";") extracted from the AMRs in Figure 1 with our extraction algorithm. *Fg* represents the corresponding subfigure.

to extract plain n-grams from sentence left endpoints. Note that the order of each n-gram is determined only by the number of nodes within it. For instance, "ask-01 :ARG0 girl" is considered as a bigram, *not* a trigram.

Our n-gram extraction method adopts breadth-first traversal to enumerate every possible starting node for extracting n-grams. From each starting node p, it extracts all possible k-grams ($1 \leq k \leq n$) beginning from p. At each node, it first stores the current k-gram before enumerating every descendant of the node and moving on. Taking the AMR graphs in Figure 1 as examples, the n-grams extracted by our method are shown in Table 1.

Processing inverse relations One important characteristic of AMR is the inverse relations, such as "ask-01 :ARG0 girl" $\Rightarrow$ "girl :ARG0-of ask-01", for preserving the properties of being rooted and acyclic. Both the original and inverse relations carry the same semantic meaning. Following SMATCH, we unify both types of relations by reverting all inverse relations to their original ones, before calculating SEMBLEU scores.

Efficiency As an important factor, the efficiency of SEMBLEU largely depends on the number of extracted n-grams. One potential problem is that there can be a large number of extracted n-grams for very dense graphs. For a fully connected graph with N nodes, there are $O(N^n)$ possible n-grams. Luckily, AMRs are tree-like graphs (Chiang et al., 2018) that are very sparse. For a tree with N nodes, the number of n-grams is bounded by $O(n \cdot N)$, which is linear in the tree scale. As tree-like graphs, we expect the number of n-grams extracted from AMRs to be roughly linear in the graph scale. Our experiments empirically confirm this expectation.

2.3 Comparison with SMATCH

In general, SMATCH breaks down the problem of comparing two AMRs into comparing the smallest units: nodes and edges. It treats each AMR as a bag of nodes and edges, and then calculates an F1 score regarding the correctly mapped nodes and edges. Given two AMRs, SMATCH searches for one-to-one mappings between the graph nodes by maximizing the overall F1 score, and the edge-to-edge mappings are automatically determined by the node-to-node mappings. Since obtaining the optimal mapping is NP-complete (by reduction from subgraph isomorphism), it uses a greedy hill-climbing algorithm to find a mapping, which is likely to be suboptimal.

One key difference is that SEMBLEU generally considers more global features than SMATCH. The only features that both metrics have in common are the node-to-node correspondences (also called unigrams for SEMBLEU). Each bigram of SEMBLEU consists two AMR nodes and one edge that connects them, thus the bigrams already capture larger contexts than SMATCH. In addition, the higher-order n-grams of SEMBLEU capture even larger correspondences. This can be a trade-off. Generally, more high-order matches indicate better parsing performance, but sometimes we want to give partial credit for distinguishing partially correct results from the fully wrong ones. As a result, combining SMATCH with SEMBLEU may give more comprehensive judgment.

Another difference is the way to determine edge (relation) equivalence. SMATCH only checks edge labels, thus two edges with the same label but conveying different meanings can be considered as equivalent by SMATCH.[2] On the other hand, SEMBLEU considers not only the edge labels but also the content of their heads and tails, as shown by the extracted n-grams in Table 1.

Take the AMRs in Figure 1 as an example, SMATCH maps "girl", "ask-01" and "leave-11" in (a) to "woman", "make-01" and "pie" in (b). As a result, it considers that "ask-01 :ARG0 girl" and "ask-01 :ARG1 leave-11" in (a) are correctly mapped to "make-01 :ARG0 woman" and "make-

[2]One example is shown in the SMATCH tutorial https://amr.isi.edu/eval/smatch/tutorial.html.

Metric	CAMR vs JAMR	CAMR vs Gros	CAMR vs Lyu	JAMR vs Gros	JAMR vs Lyu	Gros vs Lyu
SMATCH	67.9	99.9	100.0	100.0	100.0	90.3
SEMBLEU	69.0	99.9	100.0	100.0	100.0	90.9

Table 2: Corpus-level bootstrap accuracies (%) for each system pair.

01 :ARG1 pie" in (b), which does not make sense. Conversely, SEMBLEU does not consider that these edges are correctly matched.

3 Experiments

We compare SEMBLEU with SMATCH on the outputs of 4 systems over 100 sentences from the test-set of LDC2015E86. These systems are: *CAMR*,[3] *JAMR*,[4] *Gros* (Groschwitz et al., 2018) and *Lyu* (Lyu and Titov, 2018). For each sentence, following Callison-Burch et al. (2010), annotators decide relative orders instead of a complete order over all systems. In particular, 4 system outputs are randomly grouped into 2 pairs to make 2 comparisons. For each pair, we ask 3 annotators to decide which one is better and choose the majority vote as the final judgment. All the annotators have several years experience on AMR-related research, and the judgments are based on their impression on how well a system-generated AMR retains the meaning of the reference AMR. Out of the 200 comparisons, annotators are fully agree on 142, accounting for 71%. With the judgments, we study consistencies of both metrics on sentence and corpus levels.

We consider all unigrams, bigrams and trigrams for SEMBLEU, and the weights (w_ks in Equation 1) are equivalent (1/3 for each). For sentence-level evaluation, we follow previous work to use NIST geometric smoothing (Chen and Cherry, 2014). Following SMATCH, inverse relations such as "ARG0-of", are reversed before extracting n-grams for making comparisons.

3.1 Corpus-level experiment

For corpus-level comparison, we assign each system a human score equal to the number of times that system's output was preferred.

Our four systems achieved human scores of 30, 33, 63 and 74. They achieved SEMBLEU scores of 28, 30, 38 and 41, respectively, and SMATCH scores of 56, 56, 63 and 67, respectively. SEMBLEU is generally more consistent with the human judgments. In particular, there is a tie between *CAMR* and *JAMR* for SMATCH scores, while SEMBLEU scores are more discriminating. We use the script-default 2 significant digits when calculating SMATCH scores, as their variance can be very large (Figure 2). To make fair comparison, we also use 2 significant digits for SEMBLEU.

Metric	Percent (%)
SMATCH	76.5
SEMBLEU	**81.5**
SEMBLEU (n=1)	69.5
SEMBLEU (n=2)	78.0
SEMBLEU (n=4)	80.0

Table 3: Sentence-level accuracies, where the highest n-gram order is set to 3 by default, unless specified.

Bootstrap tests To conduct more comprehensive comparisons, we use bootstrap resampling (Koehn, 2004) to obtain 1000 new datasets, each having 100 instances. Every dataset contains the references, 4 system outputs and the corresponding human scores. Using the new datasets, we check how frequently SEMBLEU and SMATCH are consistent with human judgments on the corpus level as a way to perform significant test.

Table 2 shows the accuracies of both metrics across all 6 system pairs (such as *CAMR vs Lyu*). Overall, SEMBLEU is equal to or slightly better than SMATCH across all system pairs. The advantages are not significant at $p < .05$, perhaps because of the small data size, yet human judgments on large-scale data is very time consuming. Comparatively, the precisions of both metrics on *CAMR vs JAMR* is lower than the other system pairs. It is likely because the gaps of this system pair on both human and metric scores are much smaller than the other system pairs. Still, SEMBLEU is better than SMATCH on this system pair, showing that it may be more consistent with human evaluation.

3.2 Sentence-level experiment

For sentence-level comparison, we calculate the frequency with which a metric is consistent with

[3] https://github.com/c-amr/camr
[4] https://github.com/jflanigan/jamr

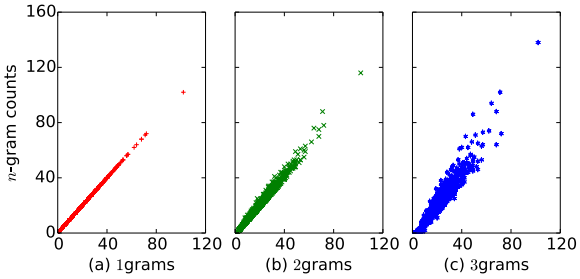

Figure 3: Extracted n-grams as a function of the number of AMR graph nodes.

human judgments on a pair of sentences. Recall that we make two pairs out of the 4 outputs for each sentence, thus there are 200 pairs in total.

As shown in the upper group of Table 3, SEMBLEU is 5.0 points better than SMATCH, meaning that it makes 10 more correct evaluations than SMATCH over the 200 instances. This indicates that SEMBLEU is more consistent with human judges than SMATCH. The lower group shows SEMBLEU accuracies with different order n. With only unigram features (node-to-node correspondences), SEMBLEU is much worse than SMATCH. When incorporating bigrams and trigrams, SEMBLEU gives consistently better numbers, demonstrating the usefulness of high-order features. Further increasing n leads to a decrease of accuracy. This is likely because humans care more about the whole-graph quality than occasional high-order matches.

3.3 Analysis on n-gram numbers

Figure 3 shows the number of extracted n-grams as a function of the number of AMR nodes on the devset of the LDC2015E86 dataset, which has 1368 instances. The number of extracted unigrams is exactly the number of AMR nodes, which is expected. The data points become less concentrated from bigrams to trigrams. This is because the number of n-grams depends on not only the graph scale, but also how dense the graph is. Overall, the amount of extracted n-grams is roughly linear in the number of nodes in the graph.

3.4 Evaluating with SEMBLEU

Table 4 shows the SEMBLEU and SMATCH scores several recent models. In particular, we asked for the outputs of *Lyu* (Lyu and Titov, 2018), *Gros* (Groschwitz et al., 2018), *van Nood* (van Noord and Bos, 2017) and *Guo* (Guo and Lu, 2018) to evaluate on our SEMBLEU. For *CAMR* and *JAMR*, we obtain their outputs by running the released systems. SEMBLEU is mostly consistent with SMATCH, except for the order between *Guo* and *Gros*. It is probably because *Guo* has more high-order correspondences with the reference.

Data	Model	SEMBLEU	SMATCH
LDC2015E86	Lyu	52.7	73.7†
	Guo	50.1	68.7†
	Gros	50.0	70.2†
	JAMR	46.8	67.0
	CAMR	37.2	62.0
LDC2016E25	Lyu	54.3	74.4†
	van Nood	49.2	71.0†
LDC2017T10	Guo	52.0	69.8†
	Gros	50.7	71.0†
	JAMR	47.0	66.0
	CAMR	36.6	61.0

Table 4: SEMBLEU and SMATCH scores for several recent models. † indicates previously reported result.

4 Conclusion

While one might expect a trade-off between speed and correlation with human judgments, SEMBLEU appears to outperform SMATCH in both dimensions. The improvement in correlation with human judgments comes from the fact that SEMBLEU considers larger fragments of the input graphs. The improvement in speed comes from avoiding the search over mappings between the two graphs. In practice, vertex mappings can be identified by simply considering the vertex labels, and the labels of their neighbors, through the n-grams in which they appear. SEMBLEU can be potentially used to compare other types of graphs, including cyclic graphs.

Acknowledgments We are very grateful to Lisa Jin and Parker Riley for making annotations. We thank Zhiguo Wang (Amazon AWS), Jinsong Su (Xiamen University) and the anonymous reviewers for their insightful comments. Research supported by NSF award IIS-1813823.

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710.

Miguel Ballesteros and Yaser Al-Onaizan. 2017. AMR parsing using stack-LSTMs. In *Proceedings of the*

2017 Conference on Empirical Methods in Natural Language Processing, pages 1269–1275.

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.

Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1215–1226.

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.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

David Chiang, Frank Drewes, Daniel Gildea, Adam Lopez, and Giorgio Satta. 2018. Weighted DAG automata for semantic graphs. *Computational Linguistics*, 44(1):119–186.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546.

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.

Jonas Groschwitz, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1831–1841.

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.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157.

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.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands (CLIN)*, 7:93–108.

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 on association for computational linguistics*, pages 311–318.

Xiaochang Peng, Linfeng Song, and Daniel Gildea. 2015. A synchronous hyperedge replacement grammar based approach for AMR parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 32–41.

Xiaochang Peng, Linfeng Song, Daniel Gildea, and Giorgio Satta. 2018. Sequence-to-sequence models for cache transition systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1842–1852.

Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Parsing English into abstract meaning representation using syntax-based machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1143–1154.

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.