Evaluation Framework for Layered Meaning Representation

Rémi de Vergnette Maxime Amblard Bruno Guillaume
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{remi.de-vergnette, maxime.amblard, bruno.guillaume}@inria.fr

Abstract

We propose different modular evaluation metrics for Layered Meaning Representation, defined as YARN, a semantic formalism encoded using rich structures that generalize AMR graphs. While existing metrics like SMATCH evaluate graph-based semantic representations such as AMR, they cannot directly handle YARN's more complex structures. We make full use of the modular nature of YARN to propose two families of metrics, depending on the linguistic features and type of semantic phenomenon targeted. The first one, SMATCHY, extends the AMR SMATCH metric. We also propose YARNBLEU, based on the SEMBLEU metric for AMR. We evaluate both families on a small dataset of human annotated YARN structures, adding random modifications simulating annotation mistakes and show that SMATCHY provides a more consistent and reliable approach with respect to the type of modifications considered.

1 Introduction

Evaluating the similarity between two graphs is a non-trivial task, as different approaches emphasize different aspects of structural variation. On the specific topic of graph based semantic formalisms, the most popular metric, SMATCH (Cai and Knight, 2013) compares AMR graphs (Banarescu et al., 2013) by matching nodes from a candidate graph to a reference graph, and treating the task as prediction, evaluating on the popular f-score metric. Alternative metrics based on SMATCH have been proposed like S²MATCH (Opitz et al., 2020) who allows soft matching by incorporating a distance function on concepts. Another popular metric for AMR evaluation is SEMBLEU (Song and Gildea, 2019), which is based on the classical Bleu metric for machine translation, and compares k-grams in the candidate and reference graphs. SMATCH and SEMBLEU have been introduced to take into account the specificities of AMR graphs, and they cannot be applied directly to other kinds of semantic formalism that are not graph-based.

We focus on layered meaning representations such as the recently introduced YARN formalism (Pavlova et al., 2024). YARN is based on AMR, but extends this formalism by adding typed edges and vertices, and enabling certain edges to go from or toward other edges. By allowing one to choose the features they would like to target (like quantification, modalities, aspect), YARN provides a modular framework for partial annotations: it is more expressive than AMR, can represent first-order logic and quantification phenomenon, as well as scope. Meaning representation-based similarity measures have been widely applied to natural language processing tasks, ranging from Natural Language Inference (Opitz et al., 2023) to text generation evaluation (Manning and Schneider, 2021) and compositional semantic similarity measurement (Fodor et al., 2025). Since YARN provides a more complete and accurate representation than AMR, similarity measures on YARN structures have the potential to yield more precise results on such tasks, provided parser accuracy. We propose decomposing the YARN structures as a set of clauses. This allows us to extend the steps presented in the original SMATCH paper to YARN structures. Furthermore, by keeping the information related to edge and vertices types in the clause decomposition, we are able to evaluate the performance of a given parser on various type of phenomenon. We extend SEMBLEU in a similar way, by proposing a way to represent YARN structures as graphs and using the same k-grams extraction method as in SEMBLEU.

We first review the classical AMR metrics SMATCH and SEMBLEU and present YARN. Then, we introduce two metrics families based on SMATCHY and YARNBLEU, and evaluate them on a small dataset of annotated YARN. Finally, we discuss the results and propose future work.

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 38–48
August 4, 2025.

2 AMR metrics

Smatch (Cai and Knight, 2013) uses a semantically motivated approach, by decomposing the candidate AMR and the reference graph as conjunctions of triples , and computing precision, recall and f-score based on predicting correct triples. Since triples involves variables, the score depends on variable matching of both graphs, and the SMATCH score is calculated as the best f-score over all possible partial one-to-one mapping between the set of variables of the two AMRs. A complete example is given in Appendix A.

SMATCH is an interpretable and semantics-driven metric: each triple represents a predicate in the event structure described by the AMR graph. Thus, it accurately captures the overlap between the two meaning associated to AMRs, in terms of asserted elementary relations between entities or variables. In particular, SMATCH does not heavily penalize incorrect labels: two AMR graphs with similar structure but different vertex labels can still score high if the number of edges outweighs the labels differences. However, using a semantically grounded metric has a cost: finding the optimal variable matching between two AMRs is NP-hard, and SMATCH relies on heuristic, non-deterministic solvers with repeated random initialization.

SemBLEU (Song and Gildea, 2019) on the other hand, does away with variable matching by taking inspiration from the classical BLEU (Papineni et al., 2002) metric and comparing k-grams predicted by the candidate graph to k-grams present in a reference graph. Since BLEU is used to evaluate machine translation, it is motivated by casting AMR parsing as translating from english to AMR. However BLEU cannot be used as is since an AMR graph is not a text sequence. Nevertheless, since BLEU relies on k-grams matching, a straightforward extension of BLEU for graphs has been proposed by (Song and Gildea, 2019) by considering k-grams as sequences of connected k-nodes. More precisely, for a reference graph z and a candidate graph c, SEMBLEU enumerates 1-grams (vertices), 2-grams (labeled edges), ..., n-grams by a traversing both graphs with a breadth-first algorithm, and then applies the standard BLEU equation:

$$\text{BLEU} = e^{min\left(1-\frac{|z|}{|c|},0\right)} \times e^{\sum_{k=1}^{n} w_k \log p_k}$$

$$p_k = \frac{|k\text{-gram}(z) \cap k\text{-gram}(c)|}{|k\text{-gram}(c)|}$$

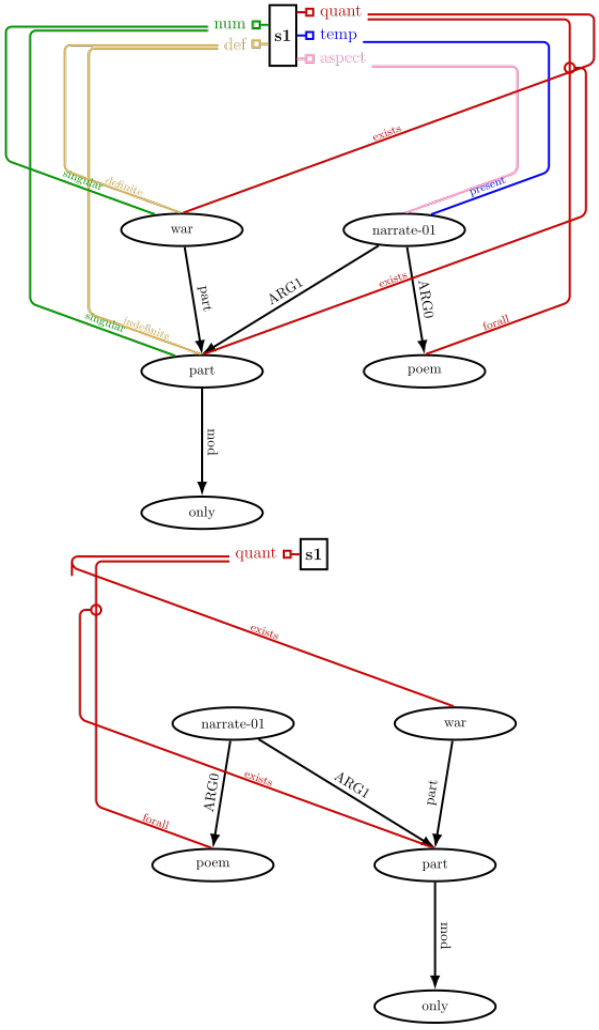


Figure 1: YARN structures representing for "Each poem narrates only a part of the war.". The second structure focuses only on the quantifier feature and the PA part.

Where w is a sequence of n positive parameters summing to 1. The authors of the original SEMBLEU paper use $n = 3$ and $w_1 = w_2 = w_2 = 1/3$.

SEMBLEU has the property of being deterministic and computable in linear time for trees. Although AMR graphs are not necessarily trees, they are generally sparse, and (Song and Gildea, 2019) empirically verified that the property still holds.

3 YARN

In this section, we give a brief overview of the YARN formalism, and how it can be represented as a set of clauses. We refer to (Pavlova et al., 2024) for a more detailed description of the formalism.

The features of YARN that we need to take into account when proposing a metric are the following: The base of a YARN structure is a graph representing the basic predicate argument (PA) structure. (i)

YARN has typed vertices. (ii) YARN has typed edge connecting different types of vertices. (iii) YARN has typed edges[1] connecting vertices or edges to other vertices or edges. (iv) YARN is modular: we might remove all vertices and edges connected to the structure only through feature nodes representing certain features we do not wish to focus on. This allows to get another simplified YARN structure. Figure 1 gives an example of this process.

We use the definition by Pavlova (2025) which, compared to Pavlova et al. (2024), provides a slightly simplified and more expressive version of the YARN formalism. We explicitly define the changes between the former and the later in the following paragraph.

A YARN structure is defined as a 9-tuple:

$$\mathcal{Y} = (S, V, F, D, E, C, L, H, I)$$

Each term denotes a set of labeled edges or vertices. The base of the representation follows AMR: V elements are vertices representing concepts, individuals or attributes, while E edges express relations between V elements. S elements are nodes corresponding to elementary events with F elements, features associated to them. D elements are edges representing discourse relations between elementary events (D is called E_s in Pavlova et al. (2024)). L elements are edges connecting F and V nodes (L is called E_{FV} in Pavlova et al. (2024)). For details on their interpretation and use to model various phenomena, see again Pavlova et al. (2024). The remaining elements are not present in Pavlova et al. (2024): C elements are edges linking V and S nodes to model clauses. H elements are edges going either from elements of F towards other elements of H or L, or from L or other H ones towards V or E. This expresses how features interact and modulate semantic relations between entities. Finally, I are undirected edges between V vertices.

4 SMATCHY

4.1 SMATCHY-BASE

SMATCH uses variables associated to nodes to handle reference towards them, encoding the structure of a graph as a collection of triples. YARN structures can be considered as classical directed graphs that have nodes of different types, with the addition of specific L or H edges that either go from another edge to a node or from a node to an edge. The only missing element in order to use SMATCH on YARN structures would be the ability to encode such edges. This can be done by adding variables corresponding to edges, as illustrated in Figure 2. With this encoding, due to the additional variables assignations, we encode YARN structures as sets of quadruples[2] (corresponding to edges) and triples (corresponding to labels of vertices), or only quadruples by adding dummy variables. An easy extension of SMATCH can then be proposed for YARN, as the best f-score that can be achieved through partial one-to-one variable matching on the clauses (triples and quadruples) defining the given SMATCH structures. We now show how to compute such a matching using integer linear programming (ILP).

ILP formulation let Y_1 and Y_2 be two graph structures, we define V_1 as the set of variables in Y_1, V_2 as the set of variables in Y_2, C_1 the set of clauses appearing in Y_1, C_2 the set of clauses appearing in Y_2.

We say that two clauses are comparable if they correspond to the same type of edge or vertex in the YARN structure, and they are labeled with the same relation, concept or feature type.

We can frame the problem of finding optimal variable alignment as an integer linear programming problem, with the given binary matrixes:

$$v : V_1 \times V_2 \to \{0; 1\} \quad t : C_1 \times C_2 \to \{0; 1\}$$

Where v_{ij} is 1 if and only if variable i is assigned to variable j, and t_{cd} is 1 if and only if the clauses c and d are comparable and match given the variable assignment.

The constraints for v to represent a partial one to one alignment are:

$$\sum_{i=1}^{n} v_{ij} \leq 1, \quad \forall j \in \{1, 2, \ldots, m\}$$

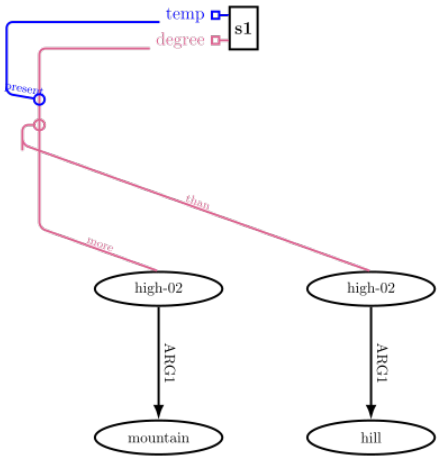
$$\sum_{j=1}^{m} v_{ij} \leq 1, \quad \forall i \in \{1, 2, \ldots, n\}$$

Additionally clauses $c_i \in C_1$ and $c_j \in C_2$ match if they are comparable and their variables match, we can formalize this in the following way: if c_i and c_j are comparable and have respective variables (x, y, z) and (a, b, c) we write:

$$t_{c_i c_j} \leq v_{xa} \quad t_{c_i c_j} \leq v_{yb} \quad t_{c_i c_j} \leq v_{zc}$$

[1] This is a slight abuse of terminology.

[2] We follow SMATCH formulation: a triple correspond to a relation together with two variables, and a quadruple consists of a relation together with three variables.



$\texttt{instance\_f}(\texttt{degree}, d)$
$\texttt{instance\_f}(\texttt{temp}, t)$
$\texttt{instance\_s}(\texttt{event}, s_1)$
$\texttt{instance\_v}(\texttt{high-02}, h)$
$\texttt{instance\_v}(\texttt{high-02}, h_2)$
$\texttt{instance\_v}(\texttt{hill}, h_3)$
$\texttt{instance\_v}(\texttt{mountain}, m)$
$\texttt{feature\_f}(d, s_1)$
$\texttt{feature\_f}(t, s_1)$

$e_1 := \texttt{ARG1\_e}(h, m)$
$e_2 := \texttt{ARG1\_e}(h_2, h_3)$
$l_1 := \texttt{more\_l}(d, h)$
$h_1 := \texttt{present\_h}(t, l_1)$
$h_2 := \texttt{than\_h}(l_1, h_2)$

Figure 2: Expression of a YARN structure representing "Mountains are higher than hills" as triples and quadruples.

$$t_{c_i c_j} \leq v_{xa} \quad t_{c_i c_j} \leq v_{yb}$$

Up to this point we follow closely the formulation of (Cai and Knight, 2013), accounting for additional variables. Most of the edges in a YARN graph are directed or between nodes of different types. The only exception to this rule in YARN structures are I edges that link V vertices and that are undirected. If c_i and c_j correspond to such vertices, linking nodes corresponding to variables x, y and a, b respectively then we may write:

$$t_{c_i c_j} \leq v_{xa} + v_{xb} \quad t_{c_i c_j} \leq v_{ya} + v_{yb}$$

Where the constraints on v insure that both right hand side are less than or equal to 1, and that if they are both 1, then $\{x, y\} = \{a, b\}$.

When clauses c_i and c_j correspond to relations that may not be compared, we write

$$t_{c_i c_j} = 0$$

Naming the set of pairs of matrixes that follow those constraints Λ, finding the best alignment is equivalent to solving the ILP problem:

$$\max_{(t,v)\in\Lambda} \sum_{c_i \in C_1, c_j \in C_2} t_{c_i c_j}$$

Since Λ is not empty (setting v and t equal to 0 satisfies all the constraints) and the function to maximize is bounded by the number of comparable clauses, the problem is well defined and can be solved in reasonable time[3] by ILP solvers.

[3]To give a rough estimate, computing the optimal alignement for a given pair of YARN structures takes about 20 ms on a personal laptop using the CBC solver (Forrest et al., 2024) through the python PuLP(Mitchell et al.) ILP modeling library.

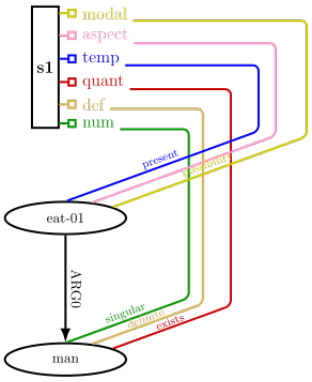


Figure 3: A simple YARN structure achieving high average base SMATCHY score (0.55 average f-score) against human annotated samples for unrelated sentences

Once an optimal mapping is found, we consider clauses that match between the candidate graph and the reference graph as true positives (TP), clauses that are present in the candidate graph and not in the reference graph as false positives (FP), and clauses that are not present in the candidate graph but are present in the reference graph as false negatives (FN): we then compute recall, precision and f-score using the usual formulas (Davis and Goadrich, 2006). Continuing with (Cai and Knight, 2013), we use the f1-score as the final metric.

4.2 Feature Aware SMATCHY

Using the previously introduced metric to compare YARN structures is unsatisfactory. It leads to considering every element of the YARN structure as equally important, either during alignment or phase. For instance, instance clauses predicting the very presence of a feature count as much as clauses relating to how this feature acts on other elements

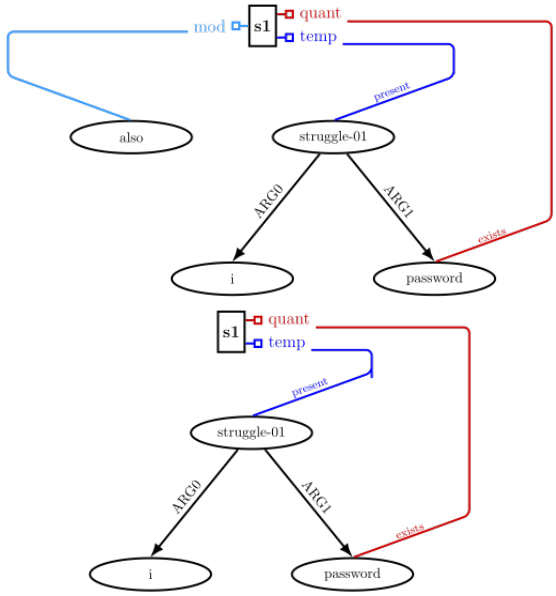


Figure 4: YARN structures for "I also struggle with passwords", before and after filtering *quant* and *temp*

of the structure, which is not ideal. Using an annotated subset of 100 sentences from the English PUD dataset (Zeman et al., 2017), the average score of random pairs of graph was 0.45, which is unsatisfactory, as one would expect this number to be closer to zero. In fact, comparing every annotated graph with a nearly empty YARN graph composed of only two V nodes and several common features gives an average score of 0.55 (see Figure 3).

Additionally, YARN has the advantage of allowing easily one to "switch" features (see Figure 1), depending on what kind of semantic phenomenon they would like to focus on. This should be reflected in any metric evaluating similarity of YARN structures: we would like to have not only a score reflecting how well two structures globally match, but also a family of derived metrics reflecting how they match on certain restricted set of features.

To tackle both challenges, we propose to retain the alignment method of the SMATCHY-BASE metric, but modify the scoring function, in order to ignore certain easy or irrelevant matches. Concretely, once an optimal variable matching is found between the variables corresponding to the two structures, we filter out the set of clauses considered for the precision and recall calculation.

Clause filtering algorithm Given a set of types $\mathbb{T} \subset \{S, V, F, D, E, C, L, H, I\}$, and a set of feature labels $\mathbb{F}$, we filter clauses by: (1) removing instance clauses defining features not in $\mathbb{F}$; (2) recursively removing clauses referencing variables from removed clauses; (3) recursively removing clauses whose variables appear only in removed clauses; and (4) removing clauses of types not in $\mathbb{T}$. Steps 1-3 "switch off" layers, see Figure 1 and Figure 4, while Step 4 filters the clauses considered according to type in order to ignore easy matches.

We now give an example of this filtering process. Let $\mathbb{T} = \{V, E, H, L\}$ and $\mathbb{F} = \{quant, temp\}$. We might choose this setting to evaluate how well a parser can extract first-order logical formulas as well as temporal features.

The YARN structure shown at the top of Figure 4 is split into the following clauses:

(1) $e_1 :=$ `ARG0_e`(s, i)
(2) $e_2 :=$ `ARG1_e`(s, p)
(3) `feature_f`(s_1, m)
(4) `feature_f`(s_1, q)
(5) `feature_f`(s_1, t)
(6) `instance_f`$(\texttt{mod}, m)$
(7) `instance_f`$(\texttt{quant}, q)$
(8) `instance_f`$(\texttt{temp}, t)$
(9) `instance_s`$(\texttt{event}, s_1)$
(10) `instance_v`$(\texttt{also}, a)$
(11) `instance_v`$(\texttt{i}, i)$
(12) `instance_v`$(\texttt{password}, p)$
(13) `instance_v`$(\texttt{struggle-01}, s_1)$
(14) $l_1 :=$ `edge_l`(m, a)
(15) $l_2 :=$ `exists_l`(q, p)
(16) $l_3 :=$ `present_l`(t, s_1)

Let's apply the four steps of the filtering process.

Step 1 Remove the instance clauses that define feature variables corresponding to features that are not in $\mathbb{F}$: Remove clause (6).

Step 2 Recursively remove the clauses referencing variables defined in clauses that have been removed: Remove clause (3), remove clause (14).

Step 3 Recursively remove clauses whose variables are referenced only in clauses that have been removed: Remove clause (10): thus the set of clause that match the second structure in Figure 4.

Step 4 Remove clauses of types that are not in $\mathbb{T}$: Remove clause (4), (5), (7), (8) and (9).

The final set of clauses is:

(1) $e_1 :=$ `ARG0_e`(s, i)
(2) $e_2 :=$ `ARG1_e`(s, p)
(11) `instance_v`$(\texttt{i}, i)$
(12) `instance_v`$(\texttt{password}, p)$
(13) `instance_v`$(\texttt{struggle-01}, s_1)$

(15) $l_2 := \texttt{exists\_l}(q, p)$

(16) $l_3 := \texttt{present\_l}(t, s_1)$

To be able to compare the general proximity of two YARN structures, we propose using our metric with $\mathbb{T} = \{S, V, D, E, C, L, H, I\}$, that is, removing only clauses of type F, with no filtering on features. With this setting, the average proximity score of pairs of structures taken randomly from our dataset drops to 0.20, while the average score between YARN structures and the structure presented in Figure 3 drops to 0.23. This is on par with results obtained using SMATCH on AMR graphs (Cai and Knight, 2013). We call the metric obtained in this setting SMATCHY-GENERAL. To have a metric focused on the PA substructure of YARN structures, we propose setting $\mathbb{T}$ to $\{V, E\}$. This is very similar to SMATCH, only using the additional more complex YARN elements to guide the variable alignment phase. We call this metric SMATCHY-PA. To evaluate on the fragment of YARN corresponding to first-order logic, we define SMATCHY-FOL by setting $\mathbb{T}$ to $\{S, V, E, H, L\}$ and $\mathbb{F}$ to $\{quant, neg\}$. We may also set $\mathbb{T}$ to $\{S, D\}$ in order to evaluate discourse relations parsing, or to $\{V\}$ for concept and entity recognition.

5 YarnBLEU

We also extend the definition of SEMBLEU to YARN structures. We leverage a graph translation of YARN structures, as seen in Figure 5: every element x of the structure is converted to a typed node $n(x)$, with type in $\{S, V, F, D, E, C, L, H, I\}$. Additionally, for every edge e in the YARN structure connecting two elements x_1 and x_2, we create two unlabeled edges $(n(x_1), n(e))$ and $(n(e), n(x_2))$. Like we did previously with SMATCHY, we propose a family of metrics, depending on the nodes considered. For a set of types $\mathbb{T}$ and features $\mathbb{F}$ applying the same process as in Figure 4.2, we extract a YARN substructure based on $\mathbb{F}$ (step 1 to 3) then select only nodes corresponding to the types in $\mathbb{T}$ before k-grams extraction. We then apply the same formula as SEMBLEU. We build in this fashion the YARNBLEU-GENERAL and YARNBLEU-PA metrics, as well as the YARNBLEU-FOL metrics that are analogous to SMATCHY-GENERAL, SMATCHY-PA and SMATCHY-FOL respectively. Since SEMBLEU additionally depends on n (the maximal size of k-grams considered) and w, we also need to set those parameters. The value proposed by (Song and Gildea, 2019) is w to $(1/3, 1/3, 1/3)$ and n to 3. In order to handle the same range of global dependencies on the PA structure while accounting for additional nodes coming from edges, we set n to 5 and w to $(1/5, ..., 1/5)$.

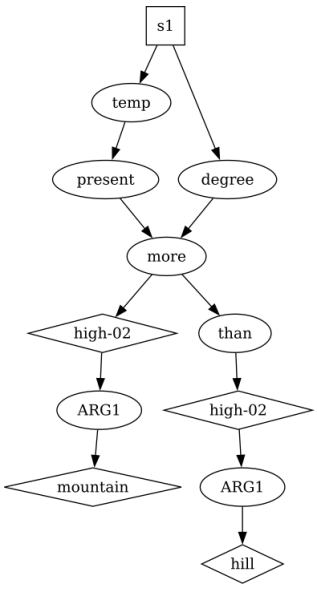


Figure 5: The graph of the YARN structure in Figure 2.

6 Experiments

6.1 Elementary modifications

We first propose a simple evaluation scheme in order to evaluate the general properties of SMATCHY- and YARNBLEU- type metrics with respect to random modifications that simulate annotator errors. We do not cover every type of mistake and the wide range of possible annotation errors. Our main focus is sensitivity and bias, as we want to measure how errors of different forms are differently penalized by such metrics. In particular, SMATCHY-GENERAL should not penalize overly one type of errors, while SMATCHY-PA should mostly penalize errors in the PA substructure of a YARN structure.

We evaluate our more fine-grained first-order oriented metrics SMATCHY-FOL and YARNBLEU-FOL on the same dataset. When changing the labels of V or E elements, we carefully introduce a new distinction. As can be seen with the node labeled "also" in Figure 4, some elements of the YARN structure will not be present in the first-order formula that can be extracted from a given YARN structure: this is typically the case for modifiers acting on elementary events. We tag such elements as "first order irrelevant" (FOI). Other elements are tagged "first order relevant" (FOR). As the con-

version of YARN structures to logical formulas is not the focus of this paper, we do not elaborate on the specific procedure one would use to build such formulas. We then compare the scores obtained when changing the label of FOR elements or FOI elements. Furthermore, modification of L and H edges are also separated between those that act on edges spanning from quantification and negation features (considered FOR) and the others (considered FOI), as only the former will have an influence on the final logical formula. The results are shown in Figure 6. As we can see, SMATCHY-FOL and YARNBLEU-FOL are able to distinguish between those two types of modifications, and only penalize acting on FOR elements. However, the general trend of fuzzier and more biased distributions for YARNBLEU metrics is still present, with YARNBLEU-FOL penalizing more modification of E edges than H or L edges.

6.2 Random chain of modifications

As a way to simulate the influence of more substantial annotation errors, we now apply sequences of random transformations to the YARN structures. This setup complements the first analysis by evaluating how metric scores degrade across cumulative and structured perturbations, rather than isolated changes. Our transformations consist in changing labels, and adding or removing random elements of types E, V, F, L, H. Those transformations are not elementary as we keep valid YARN structures at each transformation step: if a feature F is removed, we remove elements that are attached to the main structure only through this feature.[4] In the same spirit, adding a new V element, also adds an E edge linking it to the main structure. We thus keep track of the number of elementary modification (insertion, deletion of an element or change of a label) performed. We check that restricting the type of modifications to FOI elements doesn't imply a drop in YARNBLEU-FOL and SMATCHY-FOL.[5] We compute the score of the modified structures with respect to the original ones, and plot the scores as a function of the number of elementary modifications performed for SMATCHY-GENERAL, SMATCHY-PA, YARNBLEU-GENERAL and YARNBLEU-PA. The results for a small number of trajectories are shown in Figure 7. SMATCHY metrics degradation follow the editing distance more regularly than YARNBLEU. In particular, we observe mostly non increasing trajectories for SMATCHY, while this is not the case for YARNBLEU.

We note that the occasional increases observed in YARNBLEU scores is still present when changing the value of the n and w parameters. This seems to come from the precision oriented approach of SEMBLEU and YARNBLEU: removing valid elements from a modified structure might increase scores if those elements are linked to wrong ones, as it might reduce drastically the amount of wrong predicted k-grams. It is the role of the brevity penalty factor to counter this kind of effects, but it is not always sufficient: the formula proposed by (Song and Gildea, 2019) seems to rely on the assumption that AMR graphs are sparse enough that the number of k-grams extracted from them grows linearly with size of the graph: while this has been heuristically verified by the same authors on existing AMR datasets, it is not the case for YARN structures.

7 Discussion

The observed behavior of SMATCHY and YARNBLEU in our evaluation protocol leads us to favor SMATCHY for its more predictable and controlled response to parsing or annotation errors. SEMBLEU is a biased measure that penalizes mistakes differently across various regions of a graph, depending on local connectivity patterns. This bias is even more pronounced for YARN than for AMR, as complex YARN structures exhibit very different topological properties in the (H, L) substructure compared to the rest of the structure, due to specific constraints on these elements. Additionally, as noted earlier, the brevity penalty proves insufficient to address these issues.

Are there still reasons to favor SEMBLEU family metrics like YARNBLEU? The main argument appears to be computational complexity, as YARNBLEU can be computed without requiring a variable alignment phase. However, alternative solutions exist that arguably provide better approaches to assessing graph similarity (Kachwala et al., 2024; Sun and Xue, 2024; Shou and Lin, 2023). By focusing on elementary modifications, we evaluate semantic similarity on architectural grounds. YARNBLEU exhibits bias toward penalizing errors more heavily in highly connected regions of the graph, which may occasionally be desirable: in the same way AMR top elements correspond to main verbs and their core arguments, highly connected regions

[4] As a consequence, removing the quantification feature will also remove every H or L edge expressing quantification.

[5] Not obvious as FOI elements still influence the alignment.

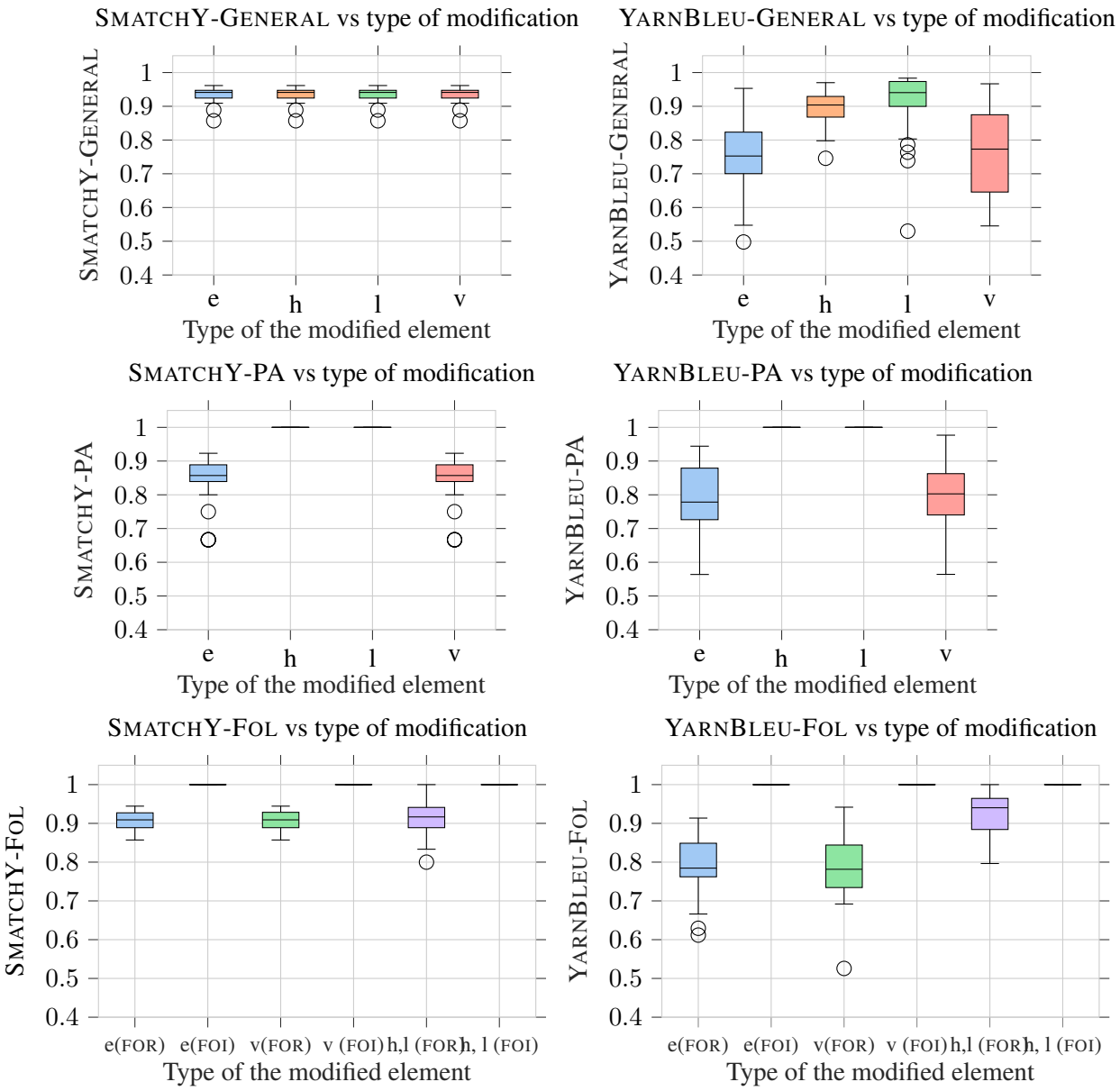


Figure 6: Distribution of scores for the metrics SMATCHY-GENERAL, YARNBLEU-GENERAL, SMATCHY-PA, YARNBLEU-PA, SMATCHY-FOL and YARNBLEU-FOL

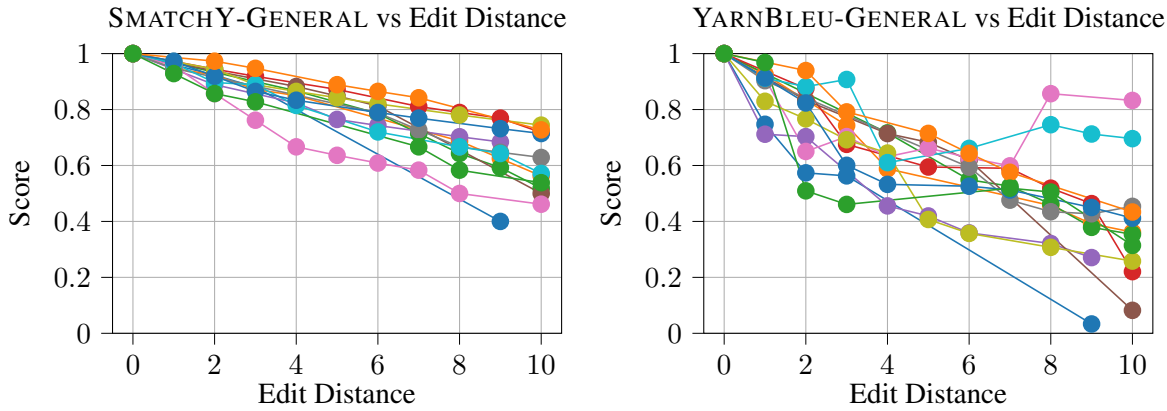


Figure 7: SMATCHY-GENERAL and YARNBLEU-GENERAL scores as functions of the number of modifications performed on the original YARN structure. Colors correspond to different sequences of modifications or original structures.

in YARN structures might correspond to elements that are more important for sentence interpretation.

To apply SEMBLEU to YARN structures, we leverage a graph translation approach. It would also be possible to apply SMATCH directly to YARN structures using this graph translation; however, we argue this is undesirable as it would represent a quadruple $a := \text{rel}(b, c)$ as three triples: instance(a, rel), $\phi(b, a)$, and $\phi(a, c)$. While this formulation suffices for checking isomorphy, it is problematic for fine-grained similarity evaluation. By tripling the number of clauses related to edges, it breaks the symmetry between nodes and edges. Additionally, compared to the quadruple formulation, this approach allows partial matching of the original edge (matching only source or target), which is unsatisfactory.

8 Conclusion

Providing evaluation metrics is a necessary first step toward the development of semantic parsers. In this context, we have introduced a new family of metrics tailored to the evaluation of parsing over YARN structures, derived from SMATCH and SEMBLEU. We have shown how to extend those original metrics to handle the specificities of YARN structures, and how to use it to evaluate parsing on different structural aspects. Those include the core predicative kernel of the structure with SMATCHY-PA and YARNBLEU-PA, the general relatedness with SMATCHY-GENERAL and YARNBLEU-GENERAL, or the first-order logic aspect with SMATCHY-FOL and YARNBLEU-FOL. We have shown that our metrics are able to distinguish and penalize different types of modifications on YARN structures. Our results suggest that using alignment based methods similar to SMATCH provide a more robust way of evaluating parsing on formalisms such YARN structures, as they seem to be less biased and more predictable than graph traversal methods such as SEMBLEU. We emphasize on the fact that many other metrics can be derived from the SMATCHY and YARNBLEU framework, allowing to focus on very specific aspects of semantic parsing, and to evaluate the overall performance or abilities of different type of models on those aspects. This results from the structural richness of YARN structures, which can be used to model a broad variety of phenomena. Furthermore, the extreme modularity of YARN allows for many applications: A single YARN annotated dataset is enough to evaluate capacities of parsers and language models across many tasks, from named entity recognition and word sense disambiguation to parsing of AMR like structures, first-order logic formulas, discourse relations and more simply by switching the evaluation metrics.

9 Limitations

The metrics we present inherit the same limitations as the ones they are based on. We can hypothesise that SMATCHY scoring systems neglect small but semantically relevant structural differences, leading to high scores for unacceptable parses, as was observed with SMATCH in (Opitz and Frank, 2022). A direction for future research is to align with human judgment by learning to aggregate different SMATCHY or YARNBLEU scores, using various choices of $\mathbb{F}$ and $\mathbb{T}$, with optimized weighting coefficients. In addition, the absence of soft concept matching penalizes structures that contain closely related but not identical concepts, overlooking nuanced semantic similarities. This limitation has been criticized and addressed in previous work on SMATCH and SEMBLEU (Opitz et al. (2020), Opitz et al. (2021)). Future work could explore incorporating soft matching in order to provide more permissive metrics evaluating semantic relatedness of YARN structures.

Furthermore, the evaluation protocol presented in this paper is biased in favor of SMATCHY because it focuses on a restricted set of modifications that induce a high variability on high level structural features of the structure as captured by YARNBLEU k-grams while leaving the underlying SMATCHY variable alignment largely unaffected.

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.

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.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine*

Learning, ICML '06, page 233–240, New York, NY, USA. Association for Computing Machinery.

Jennifer Fodor, Simon De Deyne, and Shohei Suzuki. 2025. Compositionality and sentence meaning: Comparing semantic parsing and transformers on a challenging sentence similarity dataset. *Computational Linguistics*.

John Forrest, Ted Ralphs, Stefan Vigerske, Haroldo Gambini Santos, John Forrest, Lou Hafer, Bjarni Kristjansson, jpfasano, EdwinStraver, Jan-Willem, Miles Lubin, rlougee, a andre, jpgoncal1, Samuel Brito, h-i gassmann, Cristina, Matthew Saltzman, tosttost, Bruno Pitrus, Fumiaki MATSUSHIMA, Patrick Vossler, Ron @ SWGY, and to st. 2024. coin-or/cbc: Release releases/2.10.12.

Zohair Kachwala, Jisun An, Haewoon Kwak, and Filippo Menczer. 2024. REMATCH: Robust and Efficient Matching of Local Knowledge Graphs to Improve Structural and Semantic Similarity. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Emma Manning and Nathan Schneider. 2021. Referenceless parsing-based evaluation of AMR-to-English generation. In *Eval4NLP Workshop*.

Stuart Mitchell, Anita Kean, Andrew Mason, Michael O'Sullivan, Antony Phillips, and Franco Peschiera and. Optimization with pulp.

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.

Juri Opitz, Sebastian Wein, Julia Steen, Anette Frank, and Nathan Schneider. 2023. AMR4NLI: Interpretable and robust NLI measures from semantic graphs. In *Proceedings of the International Conference on Computational Semantics (IWCS)*.

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*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Siyana Pavlova. 2025. *Tools and methods for semantically annotated corpora*. Ph.D. thesis, Université de Lorraine.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2024. YARN is All You Knit: Encoding Multiple Semantic Phenomena with Layers. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 66–76, Torino, Italia. ELRA and ICCL.

Zeyu Shou and Fangzhao Lin. 2023. Evaluate AMR graph similarity via self-supervised learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

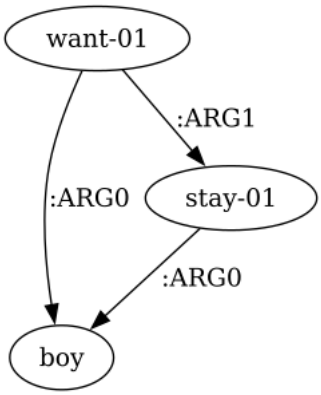
Linfeng Song and Daniel Gildea. 2019. SemBleu: A Robust Metric for AMR Parsing Evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the International Conference on Language Resources and Evaluation and the Conference on Computational Linguistics (LREC-COLING)*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

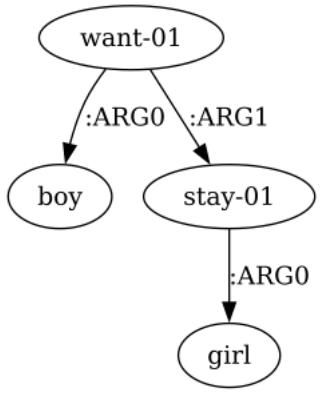
AMR graph for "the boy wants to stay"



and its triplet decomposition:

$$\texttt{instance}(w, \texttt{want-01}) \wedge$$
$$\texttt{instance}(b, \texttt{boy}) \wedge$$
$$\texttt{instance}(s, \texttt{stay-01}) \wedge$$
$$\texttt{ARG0}(w, b) \wedge$$
$$\texttt{ARG0}(s, b) \wedge$$
$$\texttt{ARG1}(w, s)$$

The same for the sentence: "the boy wants the girl to stay"



$$\texttt{instance}(x, \texttt{want-01}) \wedge$$
$$\texttt{instance}(y, \texttt{boy}) \wedge$$
$$\texttt{instance}(z, \texttt{stay-01}) \wedge$$
$$\texttt{instance}(a, \texttt{girl}) \wedge$$
$$\texttt{ARG0}(x, y) \wedge$$
$$\texttt{ARG0}(z, a) \wedge$$
$$\texttt{ARG1}(x, z)$$

Highlighted triples reflect variable alignment: blue for matching, red for non-matching. SMATCH score between the two AMR is 0.77.

Variable (First AMR)	Matching Variable (Second AMR)
w	y
b	y
s	z
(None)	a