DSCORER: A Fast Evaluation Metric for Discourse Representation Structure Parsing

Jiangming Liu Shay B. Cohen Mirella Lapata
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh

jiangming.liu@ed.ac.uk, {scohen,mlap}@inf.ed.ac.uk

Abstract
Discourse representation structures (DRSs) are scoped semantic representations for texts of arbitrary length. Evaluation of the accuracy of predicted DRSs plays a key role in developing semantic parsers and improving their performance. DRSs are typically visualized as nested boxes, in a way that is not straightforward to process automatically. COUNTER, an evaluation algorithm for DRSs, transforms them to clauses and measures clause overlap by searching for variable mappings between two DRSs. Unfortunately, COUNTER is computationally costly (with respect to memory and CPU time) and does not scale with longer texts. We introduce DSCORER, an efficient new metric which converts box-style DRSs to graphs and then measures the overlap of n-grams in the graphs. Experiments show that DSCORER computes accuracy scores that correlate with scores from COUNTER at a fraction of the time.

1 Introduction
Discourse Representation Theory (DRT) is a popular theory of meaning representation (Kamp, 1981; Kamp and Reyle, 2013; Asher, 1993; Asher et al., 2003) designed to account for a variety of linguistic phenomena within and across sentences. The basic meaning-carrying units in DRT are Discourse Representation Structures (DRSs). They consist of discourse referents (e.g., $x_1$, $x_2$) representing entities in the discourse and conditions (e.g., $\text{male.n.02}(x_1)$, $\text{Agent}(e_1, x_1)$) representing information about discourse referents. Every variable and condition are bounded by a box label (e.g., $b_1$) which implies that the variable or condition are interpreted in that box. DRSs are constructed recursively. An example of a DRS in box-style notation is shown in Figure 1(a).

DRS parsing differs from related parsing tasks (e.g., Banarescu et al. 2013) in that it can create representations that go beyond individual sentences. Despite the large amount of recently developed DRS parsing models (van Noord et al., 2018b; van Noord, 2019; Evang, 2019; Liu et al., 2019b; Fancellu et al., 2019; Le et al., 2019), the automatic evaluation of DRSs is not straightforward due to the non-standard DRS format shown in Figure 1(a).

It is neither a tree (although a DRS-to-tree conversion exists; see Liu et al., 2018, 2019a for details) nor a graph. Evaluation so far relied on COUNTER (van Noord et al., 2018a) which converts DRSs to clauses shown in Figure 1(b).

Given two DRSs with $n$ and $m$ ($n \geq m$) variables each, COUNTER has to consider $\frac{n!}{(n-m)!}$ possible variable mappings in order to find an optimal one for evaluation. The problem of finding this alignment is NP-complete, similar to other metrics such as SMATCH (Cai and Knight, 2013a) for Abstract Meaning Representation. COUNTER uses a greedy hill-climbing algorithm to obtain one-to-one variable mappings and then computes precision, recall, and F1 scores according to the overlap of clauses between two DRSs. To get around the problem of search errors, the hill-climbing search implementation applies several random restarts. This incurs unacceptable runtime, especially when evaluating document-level DRSs with a large number of variables.

Another problem with the current evaluation is that COUNTER only considers local clauses without taking larger window sizes into account. For example, it considers “$b_1$ sing $e_2$” and “$b_3$ NOT $b_1$” as separate semantic units. However, it would also make sense to assess “$b_3$ NOT $b_1$ sing $e_2$” as a whole without breaking it down into smaller parts. By considering higher-order chains, it is possible to observe more global differences in DRSs which are important when assessing entire documents.

In order to address the above issues, we propose DSCORER, a highly efficient metric for the evalu-
The algorithm operates over $n$-grams in a fashion similar to BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), which are metrics widely used for evaluating the output of machine translation and summarization systems. While BLEU only calculates precision with a brevity penalty (it is not straightforward to define recall given the wide range of possible translations for a given input), ROUGE is a recall-oriented metric since the summary length is typically constrained by a pre-specified budget.\footnote{See \url{https://github.com/tensorflow/tensor2tensor} for computing ROUGE F1.} However, in DRS parsing, there is a single correct semantic representation (gold-standard reference) and no limit on the maximum size of DRSs. Our proposed metric, DS\textsc{corer}, converts box-style DRSs to a graph format used for evaluation and computes F1 with high efficiency (7,000 times faster compared to C\textsc{ounter}). We release our code, implementing the metric, at \url{https://github.com/LeonCrashCode/DRSScorer}.

2 DS\textsc{corer}

The proposed metric converts two box-style DRSs into graphs, extracts $n$-grams from these graphs, and then computes precision, recall, and F1 score based on the $n$-gram overlap.

2.1 Graph Induction

Following the work of van Noord et al. (2018a), box-style DRSs can be converted to clauses as shown in Figure 1(b). For example, box $b_1$ is in a contrast relationship to box $b_4$ within box $b_0$ which corresponds to the clause $b_0$ CONTRAST $b_1$ $b_4$; variable $b_2 : x_1$ is converted to clause $b_2$ REF $x_1$, and the condition $b_1 : t_1 < \text{"now"}$ is converted to $b_1$ TPR $t_1$ \text{"now"}.\footnote{REF and TPR are operators abbreviating “referent” and “temporally precedes”, respectively; see \url{https://pmb.let.rug.nl/drs.php} for more detail.}

We now explain how we convert DRSs to graphs. There are two types of clauses depending on the number of arguments: 2-argument clauses (e.g., $b_2$ male.n.02 $x_1$) and 3-argument ones (e.g., $b_1$ Agent $e_1$ $x_1$). The two types of clauses can be formatted as node $\xrightarrow{\text{edge}}$ node and node $\xrightarrow{\text{edge}}$ node $\xrightarrow{\text{edge}}$ node, respectively. For example, clause “$b_2$ male.n.02 $x_1$” is rendered as $b_2 : \text{male.n.02}(x_1)$ $b_1 : \text{time.n.08}(t_1)$ $b_1 : t_1 < \text{"now"}$ $b_1 : \text{play.v.03}(e_1)$ $b_1 : \text{Agent}(e_1, x_1)$ $b_1 : \text{piano.n.01}(x_2)$

![Figure 1: (a) Box-style DRS for the text “He didn’t play the piano. But she sang.”; (b) Clause-style DRS for the text “He didn’t play the piano. But she sang.”; (c) Proposed graph-style DRS format for the text “He didn’t play the piano. But she sang.”](image-url)
an agent), we make edges bidirectional (red in Figure 1(c)) if they do not connect the two b nodes.

Next, we rewrite the nodes, keeping their type\(^3\) (e.g., B, X, E, S, P, and T) but not their indices and the resulting graph is shown in Figure 1(c). In addition to being typed, variables can be distinguished by their neighboring nodes and connecting edges. For example, the two E nodes are different. One is on the path \(B \xrightarrow{\text{play.v}.03} E \xrightarrow{\text{Theme-A2}} X \xrightarrow{\text{piano.n}.01} B\) showing that the Theme of the predicate play is piano, and the other is on the path \(B \xrightarrow{\text{sing.v}.01} E \xrightarrow{\text{Agent-A2}} X \xrightarrow{\text{female.n}.02} B\) showing that the Agent of the predicate sing is female. To compare two graphs, we compute the overlap between extracted paths instead of searching for best node mappings, which saves computational resources (i.e., CPU memory and time).

### 2.2 Evaluation Based on n-grams

An n-gram in our case is an Euler path\(^4\) on a graph with n edges. For example, \(B \xrightarrow{\text{Theme-A1}} E \xrightarrow{\text{Theme-A1}} E \xrightarrow{\text{Theme-A2}} E \xrightarrow{\text{Theme-A1}} E \xrightarrow{\text{Theme-A2}} X \xrightarrow{\text{piano.n}.01} B\) is a 1-gram as it contains a single edge, \(B \xrightarrow{\text{Theme-A1}} E \xrightarrow{\text{Theme-A2}} X \xrightarrow{\text{piano.n}.01} B \xrightarrow{\text{Theme-A2}} E \xrightarrow{\text{Theme-A1}} E \xrightarrow{\text{Theme-A2}} X \xrightarrow{\text{piano.n}.01} B\) is a 3-gram since it has three edges, and a single node is a 0-gram. We extract the n-grams for each node in a graph. Due to the high sparsity of graphs typical for DRSs, the number of n-grams does not explode as the size of graphs increases, \(|G| = |N| + |E|\), where \(|N|\) and \(|E|\) are the number of nodes and edges in graph \(G\), respectively. Given the n-grams of predicted and gold DRS graphs, we compute precision \(p_k\) and recall \(r_k\) as:

\[
\begin{align*}
   p_k &= \frac{|k\text{-grams}_{\text{pred}} \cap k\text{-grams}_{\text{gold}}|}{|k\text{-grams}_{\text{pred}}|} \quad (1) \\
r_k &= \frac{|k\text{-grams}_{\text{pred}} \cap k\text{-grams}_{\text{gold}}|}{|k\text{-grams}_{\text{gold}}|} \quad (2)
\end{align*}
\]

where \(k\text{-grams}_{\text{pred}}\) and \(k\text{-grams}_{\text{gold}}\) are k-grams on predicted and gold DRS graphs, respectively, and \(f_k = \frac{2p_k r_k}{p_k + r_k}\), where \(p_0 = r_0 = f_0 = \min(|N_{\text{pred}}|, |N_{\text{gold}}|)\). DSCORER calculates precision, recall, and F1 as:

\[
\text{DSCORER}_{n_F} = \exp \left( \sum_{k=1}^{n} w_k \log F_k \right) \quad (3)
\]

\(^3\) B refers to box labels, X to entities, E to events, S refers to states, P to propositions, and T to time.

\(^4\) An Euler path is a path that visits every edge of a graph exactly once (allowing for revisiting nodes).

### 3 Experiments

In our experiments, we investigate the correlation between DSCORER and COUNTER, and the efficiency of the two metrics. We present results on two datasets, namely the Groningen Meaning Bank (GMB; Bos et al. 2017) and the Parallel Meaning Bank (PMB; Abzianidze et al. 2017). We compare two published systems on the GMB: DRTS-sent which is a sentence-level parser (Liu et al., 2018) and DRTS-doc which is a document-level parser (Liu et al., 2019a). On the PMB, we compare seven systems: Boxer, a CCG-based parser (Bos, 2015), AMR2DRS, a rule-based parser that converts AMRs to DRSs, SIM-SPAR giving the DRS in the training set most similar to the current DRS, SPAR giving a fixed DRS for each sentence, seq2seq-char, a character-based sequence-to-sequence clause parser (van Noord et al., 2018b), seq2seq-word, a word-based sequence-to-sequence clause parser, and a transformer-based clause parser (Liu et al., 2019b).

### 3.1 Metric Settings

COUNTER takes 100 hill-climbing restarts to search for the best variable mappings on PMB and 10 restarts on GMB. Both DSCORER and COUNTER are computed on one CPU (2.10GHz). The weight \(w_0\) is set to 0.1 and the weights \(w_k\) (1 ≤ k ≤ n) in DSCORER are set to 0.9/n, where n = 4.

### 3.2 Analysis

We analyze the number of n-grams extracted by DSCORER; we also report the values obtained by
### Systems Evaluation

<table>
<thead>
<tr>
<th>Systems</th>
<th>COUNTER</th>
<th>DSCORER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>PMB</td>
<td>39.7</td>
<td>6.5</td>
</tr>
<tr>
<td>AMR2DRS</td>
<td>43.2</td>
<td>17.5</td>
</tr>
<tr>
<td>SIM-SPAR</td>
<td>56.8</td>
<td>41.8</td>
</tr>
<tr>
<td>Boxer</td>
<td>74.3</td>
<td>56.7</td>
</tr>
<tr>
<td>seq2seq-word</td>
<td>83.1</td>
<td>72.4</td>
</tr>
<tr>
<td>seq2seq-char</td>
<td>83.6</td>
<td>71.9</td>
</tr>
<tr>
<td>transformer</td>
<td>87.4</td>
<td>79.8</td>
</tr>
</tbody>
</table>

**Table 1:** System evaluation according to COUNTER and DSCORER which runs on 4-grams.

<table>
<thead>
<tr>
<th>System</th>
<th>PMB</th>
<th>GMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRTS-sent</td>
<td>77.9</td>
<td>66.7</td>
</tr>
<tr>
<td>DRTS-doc</td>
<td>66.7</td>
<td>60.0</td>
</tr>
</tbody>
</table>

**Table 2:** Average runtime (secs) for a pair of DRSs, where $|G|$ is the average graph size and $|N_G|$ is the average number of nodes in a graph.

### Number of n-grams

Figure 2(a) shows the number of $n$-grams across graphs in GMB where the largest size of 4-grams extracted on one graph is $1.47 \times 10^6$. Figure 2(b) shows the number of $n$-grams across graphs in PMB where the largest size of 4-grams extracted on one graph is $2.27 \times 10^3$. The number of $n$-grams will increase exponentially with $n$ or as the size of the graph increases. Nevertheless, the number of 4-grams remains manageable. We set $k = 4$ for computing our metric (see Equations (1) and (2)) as 4-grams are detailed enough to capture differences between meaning representations whilst avoiding overly strict matching (which would render the similarity between predicted and gold DRSs unnecessarily low and not very useful).

### Metric Values

Table 1 shows the various scores assigned by DSCORER and COUNTER to the different systems. We observe similar trends for both metrics; DSCORER penalizes more harshly SPAR and SIM-SPAR, which output random DRSs without any parsing algorithm. Generally speaking, the two metrics are highly correlated; across systems and datasets, Pearson’s correlation coefficient $r$ is 0.93 on 1-grams, 0.94 on 2-grams, 0.91 on 3-grams, and 0.88 on 4-grams, with 2-grams being most correlated. This is not surprising, 2-grams in DSCORER are most similar to COUNTER which only considers predicates with at most two arguments. Figure 3 shows the 4-gram correlation between COUNTER and DSCORER. We found most points are around the curve of $y = x^3$, which means that considering high-order grams renders the two metrics less similar, but nevertheless allows to more faithfully capture similarities or discrepancies between DRSs.

### Efficiency

Table 2 shows the average run-time for COUNTER and DSCORER on a pair of DRSs. Both metrics have similar run-times on PMB which mostly consists of small graphs. However, in GMB, which consists of larger graphs with many nodes, the run-time of COUNTER explodes (more than 4 hours per graph), while DSCORER evaluates DRSs within an acceptable time frame (2.35 seconds per graph). In GMB-doc, DSCORER runs seven thousand times faster than COUNTER, showing it is very efficient at comparing large graphs.

### Case Study

We further conducted a case study in order to analyze what the two metrics measure. Figure 4 shows two different sentences in their clause-style DRS format used by COUNTER and graph-style DRS format used by DSCORER. Note that the two sentences have totally different meanings (distinguished using various meaning constructs in the corresponding DRSs). Using COUNTER to compare the two sentences yields an F1 of 47.06, which drops to 16.11 when employing DSCORER on 4-grams. Note that DSCORER on 1-grams obtains an F1 of 46.42 which is close to COUNTER.

COUNTER takes matching clauses into account...
We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of the European Research Council (Lapata, Liu; grant agreement 688139) and Bloomberg (Cohen, Liu).
References


Nicholas Asher. 1993. Reference to abstract objects in english.


Appendix

Figure 5 shows the complete graph for Figure 1(c).

Figure 5: The complete DRS graph for Figure 1(c)