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., $male.n.02(x_1)$, $Agent(e_1, x_1)$) 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_4 sing e_2" and "b_3 NOT b_4" as separate semantic units. However, it would also make sense to assess " b_3 NOT b_4 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-

ation of DRS parsing on texts of arbitrary length. DSCORER converts DRSs (predicted and gold) to graphs from which it extracts n-grams, and then computes precision, recall and F1 scores between them. 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 prespecified budget.[1] 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, DSCORER, converts box-style DRSs to a graph format used for evaluation and computes F1 with high efficiency (7,000 times faster compared to COUNTER). We release our code, implementing the metric, at https://github.com/LeonCrashCode/DRSScorer.

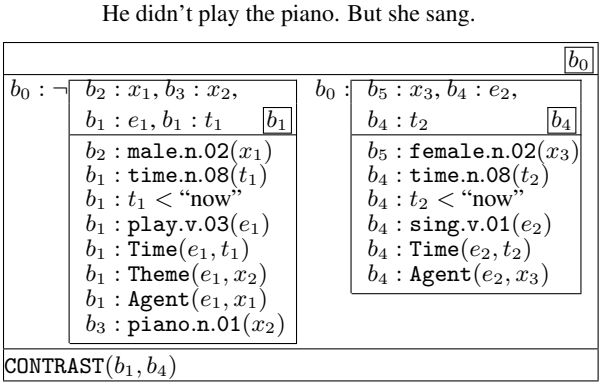
2 DSCORER

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 $b_1 : t_1 <$ "now" is converted to b_1 TPR t_1 "now".[2]

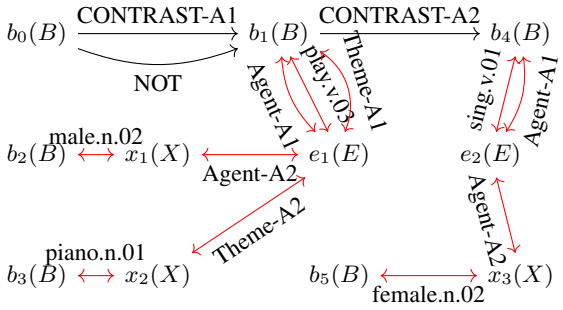
We now explain how we convert DRSs to graphs. There are two types of clauses depending on the number of arguments: 2-argument clauses (e.g., b_2 male.n.02 x_1) and 3-argument ones (e.g., b_1 Agent e_1 x_1). The two types of clauses can be formatted as $node \xrightarrow{edge} node$ and $node \xrightarrow{edge}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



(a)

b_0 CONTRAST b_1 b_4	b_3 REF x_2
b_0 NOT b_1	b_3 piano "n.01" x_2
b_2 REF x_1	b_5 REF x_3
b_2 male "n.02" x_1	b_5 female "n.02" x_3
b_1 REF e_1	b_4 REF e_2
b_1 REF t_1	b_4 REF t_2
b_1 Agent e_1 x_1	b_4 Agent e_2 x_3
b_1 TPR t_1 "now"	b_4 TPR t_2 "now"
b_1 Theme e_1 x_2	b_4 Time e_2 t_2
b_1 Time e_1 t_1	b_4 sing "v.01" e_2
b_1 play "v.03" e_1	b_4 time "n.08" t_2
b_1 time "n.08" t_1	

(b)



(c)

Figure 1: (a) Box-style DRS for the text "He didn't play the piano but she sang."; (b) Clause-style DRS format for COUNTER; (c) Proposed graph-style DRS format (abridged version shown; complete graphs can be found in the Appendix).

$b_2 \xrightarrow{male.n.02} x_1$, and clause "$b_1$ Agent e_1 x_1" as $b_1 \xrightarrow{Agent\text{-}A1} e_1 \xrightarrow{Agent\text{-}A2} x_1$. Same nodes are further merged to a single node. For example, x_1 nodes in $b_2 \xrightarrow{male.n.02} x_1$ and $e_1 \xrightarrow{Agent\text{-}A2} x_1$ are merged to a single node x_1. The induced graph is directed and yields the chain $b_1 \xrightarrow{Agent\text{-}A1} e_1 \xrightarrow{Agent\text{-}A2} x_1$. In order to capture interactions between chains, (e.g., chain $b_2 \xrightarrow{male.n.02} x_1$, assigns x_1 as a predicate "male.n.02" but x_1 is also

[1] See https://github.com/tensorflow/tensor2tensor for computing ROUGE F1.

[2] REF and TPR are operators abbreviating "referent" and "temporally precedes", respectively; see https://pmb.let.rug.nl/drs.php for more detail.

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 $B \xrightarrow{play.v.03} E \xrightarrow{Theme\text{-}A2} X \xrightarrow{piano.n.01} B$ showing that the *Theme* of the predicate *play* is *piano*, and the other is on the path $B \xrightarrow{sing.v.01} E \xrightarrow{Agent\text{-}A2} X \xrightarrow{female.n.02} B$ 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, $B \xrightarrow{Theme\text{-}A1} E$ is a 1-gram as it contains a single edge, $B \xrightarrow{Theme\text{-}A1} E \xrightarrow{Theme\text{-}A2} X \xrightarrow{piano.n.01} B$ 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:

$$p_k = \frac{|k\text{-grams}_{pred} \cap k\text{-grams}_{gold}|}{|k\text{-grams}_{pred}|} \quad (1)$$

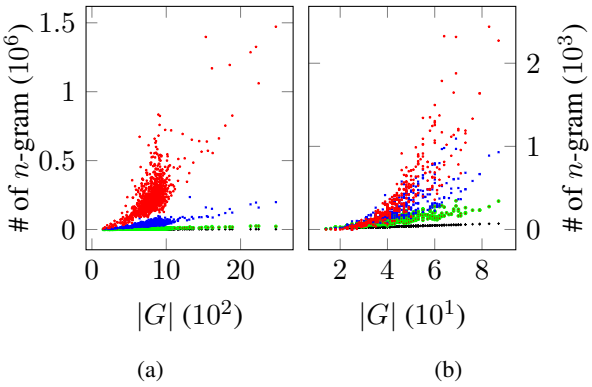
$$r_k = \frac{|k\text{-grams}_{pred} \cap k\text{-grams}_{gold}|}{|k\text{-grams}_{gold}|} \quad (2)$$

where k-grams$_{pred}$ and k-grams$_{gold}$ are k-grams on predicted and gold DRS graphs, respectively, and $f_k = \frac{2p_k r_k}{p_k + r_k}$, where $p_0 = r_0 = f_0 = \frac{min(|N_{pred}|,|N_{gold}|)}{max(|N_{pred}|,|N_{gold}|)}$. 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).



(a) (b)

Figure 2: Number of n-grams in (a) GMB and (b) PMB. Red points are 4-grams, blue points are 3-grams, green points are 2-grams and black points are 1-grams.

where w_k is a fixed weight for k-gram ($0 \leq k \leq n$) counts, and $F \in \{p, r, f\}$.

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 ($1 \leq k \leq n$) 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	COUNTER	DSCORER		
		P	R	F1
PMB				
SPAR	39.7	6.5	19.7	9.2
AMR2DRS	43.2	17.5	23.3	19.7
SIM-SPAR	56.8	41.8	39.2	40.2
Boxer	74.3	56.7	58.4	57.6
seq2seq-word	83.1	72.4	75.1	73.7
seq2seq-char	83.6	71.9	75.3	73.5
transformer	87.4	79.8	82.1	80.9
GMB				
DRTS-sent	77.9	66.7	65.3	65.9
DRTS-doc	66.7	60.0	62.9	61.4

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

dataset	$\|G\|$	$\|N_G\|$	COUNTER	DSCORER
PMB	39.93	7.83	0.006	0.004
GMB-sent	122.07	20.28	3.03	0.14
GMB-doc	801.87	120.86	14428.68	2.35

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.

DSCORER and COUNTER on the two datasets, their correlation, and efficiency.

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 1.47×10^6. 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 2.27×10^3. 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 unncessarily 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 datatasets, 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

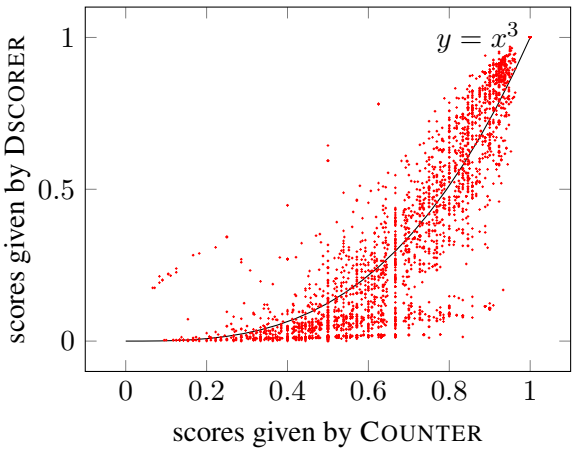


Figure 3: Pearson's r between DSCORER (on 4-grams) and COUNTER (across systems and datasets).

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.

3.3 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

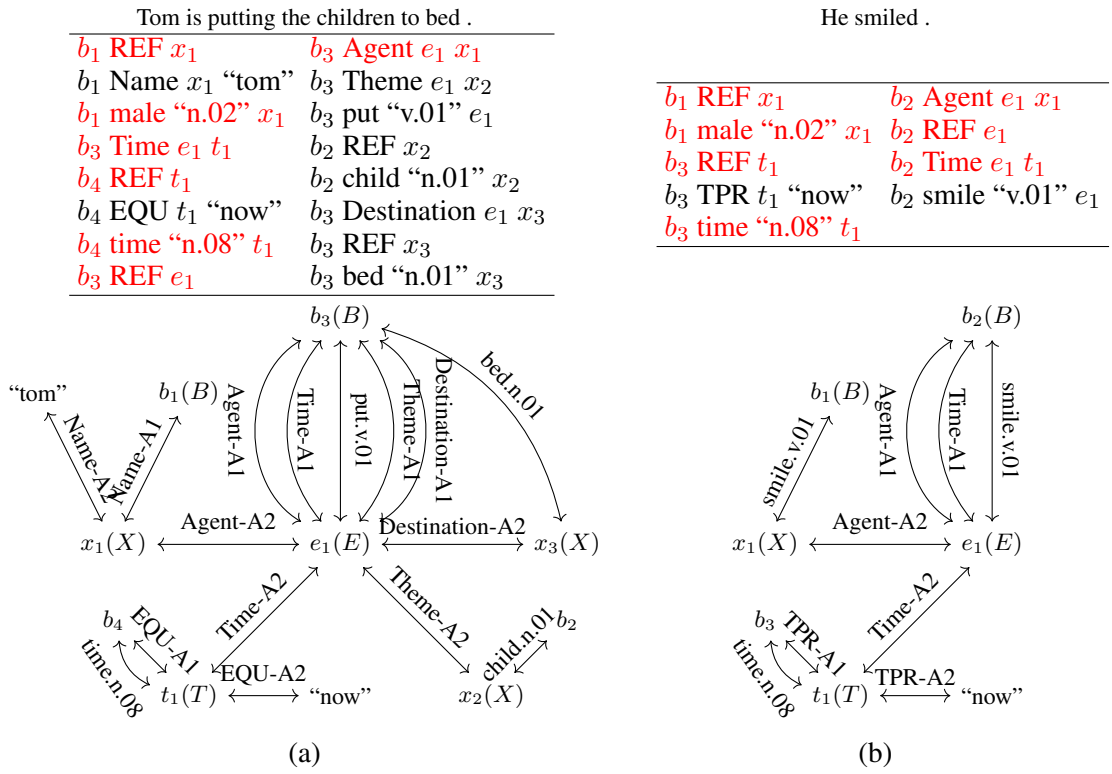


Figure 4: (a) DRS for the sentence "Tom is putting the children to bed."; (b) DRS for the sentence "He smiled."; we omit the "REF" relation from the graph for the sake of clarity.

(marked as red in Figure 4), which might inflate the similarity between two sentences without actually measuring their core meaning. For example, the common relation "b_3 Time e_1 t_1" is matched to "b_2 Time e_1 t_1" without considering what e_1 and t_1 are. Instead, DSCORER aims to find matches for paths $B \xrightarrow{Time-A1} e_1 \xrightarrow{Time-A2} t_1$ and $B \xrightarrow{smile.v.01} e_1 \xrightarrow{Time-A2} t_1$ as well. And the mismatch of the second path reduces the final score.

4 Related Work

The metric SEMBLEU (Song and Gildea, 2019) is most closely related to ours. It evaluates AMR graphs by calculating precision based on n-gram overlap. SEMBLEU yields scores more consistent with human evaluation than SMATCH (Cai and Knight, 2013b), an AMR metric which is the basis of COUNTER. SEMBLEU cannot be directly used on DRS graphs due to the large amount of indexed variables and the fact that the graphs are not explicitly given; moreover, our metric outputs F1 scores instead of precision only.

Opitz et al. (2020) propose a set of principles for AMR-related metrics, showing the advantages and drawbacks of alignment- and BLEU-based AMR metrics. However, efficiency of the metric is crucial for the development of document-level models of semantic parsing. Basile and Bos (2013) propose to represent DRSs via Discourse Representation Graphs (DRGs) which are acyclic and directed. However, DRGs are similar to flattened trees, and not able to capture clause-level information (e.g., b_1 Agent e_1 x_1) required for evaluation (van Noord et al., 2018a).

5 Conclusions

In this work we proposed DSCORER, as a DRS evaluation metric alternative to COUNTER. Our metric is significantly more efficient than COUNTER and considers high-order DRSs. DSCORER allows to speed up model selection and development removing the bottleneck of evaluation time.

Acknowledgments

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of the European Research Council (Lapata, Liu; award number 681760), the EU H2020 project SUMMA (Cohen, Liu; grant agreement 688139) and Bloomberg (Cohen, Liu).

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

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

Nicholas Asher, Nicholas Michael Asher, and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

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.

Valerio Basile and Johan Bos. 2013. Aligning formal meaning representations with surface strings for wide-coverage text generation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 1–9.

Johan Bos. 2015. Open-domain semantic parsing with boxer. In *Proceedings of the 20th nordic conference of computational linguistics (NODALIDA 2015)*, pages 301–304.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The groningen meaning bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

Shu Cai and Kevin Knight. 2013a. 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.

Shu Cai and Kevin Knight. 2013b. 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.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Federico Fancellu, Sorcha Gilroy, Adam Lopez, and Mirella Lapata. 2019. Semantic graph parsing with recurrent neural network DAG grammars. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2769–2778, Hong Kong, China. Association for Computational Linguistics.

Hans Kamp. 1981. A theory of truth and semantic representation. *Formal semantics-the essential readings*, pages 189–222.

Hans Kamp and Uwe Reyle. 2013. *From discourse to logic: Introduction to model theoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Springer Science & Business Media.

Ngoc Luyen Le, Yannis Haralambous, and Philippe Lenca. 2019. Towards a drs parsing framework for french. In *Advances in Natural Language Processing*, Grannada, Spain.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2018. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439, Melbourne, Australia. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019a. Discourse representation parsing for sentences and documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6248–6262, Florence, Italy. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019b. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord. 2019. Neural boxer at the IWCS shared task on DRS parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018a. Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018b. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. Amr similarity metrics from principles. *arXiv preprint arXiv:2001.10929*.

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. Association for Computational Linguistics.

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.

A Appendix

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

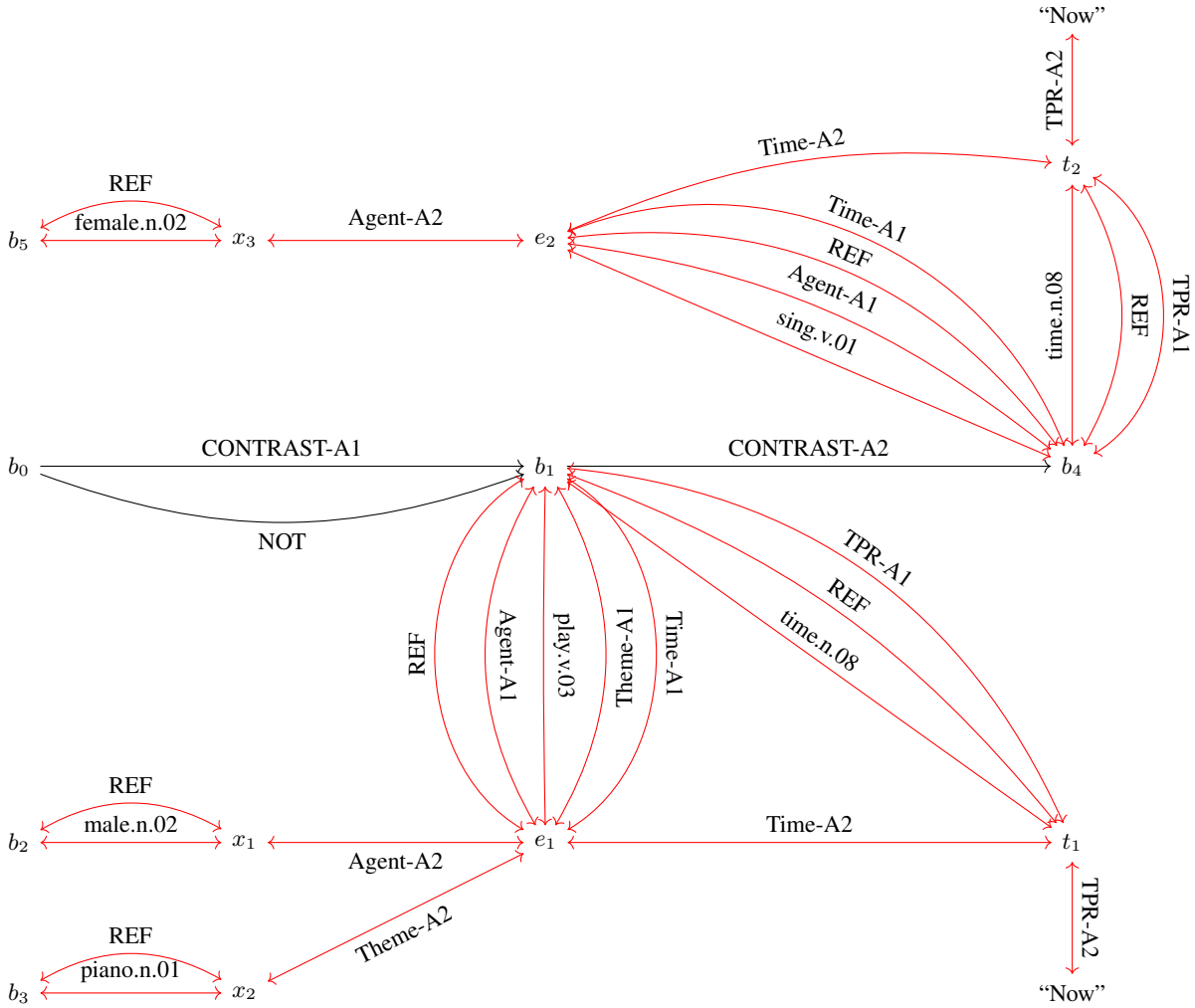


Figure 5: The complete DRS graph for Figure 1(c)