Rewarding Semantic Similarity under Optimized Alignments for AMR-to-Text Generation

Lisa Jin and **Daniel Gildea**
Department of Computer Science
University of Rochester
Rochester, NY 14627

Abstract

A common way to combat exposure bias is by applying scores from evaluation metrics as rewards in reinforcement learning (RL). Metrics leveraging contextualized embeddings appear more flexible than those that match n-grams and thus ideal as training rewards. Yet metrics such as BERTSCORE greedily align candidate and reference tokens, which can give system outputs excess credit relative to a reference. Past systems using such semantic similarity rewards further suffer from repetitive outputs and overfitting. To address these issues, we propose metrics that replace the greedy alignments in BERTSCORE with optimized ones. Our model optimizing discrete alignment metrics consistently outperforms cross-entropy and BLEU reward baselines on AMR-to-text generation. Additionally, we find that this model enjoys stable training relative to a non-RL setting.

1 Introduction

Automatic evaluation metrics often score natural language generation (NLG) system outputs based on how well they lexically align to human-annotated references. In tasks such as machine translation and summarization, these metrics may unfairly penalize outputs that express the correct semantics despite a lower n-gram overlap with reference strings. As a result, models overfitting to certain token-level patterns may dominate those generating more creatively (e.g., through synonyms or varied sentence structure).

NLG systems are typically trained to maximize likelihood of a single set of references. Conditioning models on gold prefixes shields them from their own predictions during training—an issue known as exposure bias. Adding reinforcement learning (RL) objectives (Ranzato et al., 2016; Edunov et al., 2018) can aid exploration by giving a model feedback on sequences sampled from its own distribution. However, it is common practice to use automatic evaluation scores like BLEU (Papineni et al., 2002) and ROUGE (Lin and Hovy, 2002) as sequence-level rewards. This results in the same lack of semantic signal described earlier.

Instead of hinging evaluation on hard n-gram overlap, recent metrics (Zhang et al., 2019; Zhao et al., 2019) rely on vector similarity between contextualized subword embeddings to make more semantically faithful judgments. BERTSCORE, in particular F_{BERT}, computes a token-level F1 score based on greedy alignment of similar embeddings. With their strength in offline evaluation, it is natural to ask how these embeddings-based metrics can help provide more realistic training feedback.

Past approaches to train models with semantic similarity scores include both non-differentiable and differentiable objectives. Wieting et al. (2019) separately train paraphrastic sentence embeddings that provide semantic similarity rewards to a neural machine translation (NMT) system. Rewards were included in a mixed minimum risk and maximum likelihood training phase. Besides an embedding training overhead, the model needed a length penalty term to limit repetitive outputs. Li et al. (2019) adopt a similar fine-tuning approach using an RL objective with F_{BERT} for abstractive summarization. While their models were less repetitive, their news domain corpora may have been a natural match for BERT embeddings. Finally, Jauregi Unanue et al. (2021) also propose to optimize F_{BERT} but with fully differentiable training objectives in NMT. Yet their models overfit after only a few epochs and scored lower in BLEU at the cost of higher F_{BERT}. We hypothesize that metrics employing external pretrained vectors may suffer from domain mismatch with downstream data. This can hurt the accuracy of semantic similarity scores computed during training.

In this work, we focus on text generation from Abstract Meaning Representations (AMRs, Banarescu et al., 2013), sentence-level semantic

graphs that are rooted, directed, and acyclic. This task's models may especially benefit from an emphasis on semantic rather than lexical similarity. It also provides a challenging setting to evaluate overfitting given the relatively small corpus size.

In our analysis of F_{BERT} rewards, we note that F_{BERT} could worsen repetition and incomplete outputs in NLG systems. Due to its greedy token alignment, F_{BERT} precision may assign extra credit to a reference token 'retrieved' multiple times. In response, we contribute the following.

- We introduce metrics that apply discrete and continuous alignments to BERTSCORE, mitigating the pitfalls of greedy alignment.
- For text generation from AMR, we are the first to train on RL objectives with embeddings-based evaluation metrics.
- As RL rewards, we compute BERTSCORE-based metrics on a model's own token representations rather than BERT embeddings. This is more memory-efficient and does not overfit relative to pure cross-entropy training.

2 Greedy Token Alignment

The main insight behind BERTSCORE and related metrics is to align hypothesis and reference tokens using their pairwise vector similarity scores. These alignments are later used to weight the contribution of token-level similarity scores towards a final sequence-level score. Concretely, given $(\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_m)$ and $(\mathbf{y}_1, \ldots, \mathbf{y}_k)$ hypothesis and reference token embeddings, precision in F_{BERT} is

$$P_{\text{BERT}} = \frac{1}{m} \sum_{\hat{y}_i \in \hat{y}} \max_{y_j \in y} \cos(\hat{\mathbf{y}}_i, \mathbf{y}_j),$$

where $\cos(\hat{\mathbf{y}}, \mathbf{y}) = \hat{\mathbf{y}}^\top \mathbf{y} / \|\hat{\mathbf{y}}\| \, \|\mathbf{y}\|$ denotes cosine similarity. Each hypothesis token $\hat{y}_i$ is greedily aligned to the reference token y_j with the highest corresponding embedding cosine similarity. Unlike in BLEU, P_{BERT} does not clip the number of times $\hat{y}_i$ can align to a unique y_j by its count in y. As such, a hypothesis will get excess credit by repeating a reference token beyond this count. While the authors claim greedy alignments have little effect on BERTSCORE evaluation performance, they perform poorly relative to metrics based on optimized alignments in our experiments.

3 Optimized Token Alignment

Aligning tokens between hypothesis and reference can be seen as an assignment problem, where a token pair $(\hat{y}_i, y_j)$ is highly weighted if it incurs low cost (i.e., distance).

Here, we describe discrete token matching (one-to-one) and soft alignment (one-to-many). For the latter, we extract alignments from the earth mover's distance (EMD, Villani, 2009; Peyré and Cuturi, 2019) transport matrix. We weight pairwise token similarities as in F_{BERT} using each of these two alignments to provide metrics F_{DISC} and F_{CONT}.

3.1 Discrete word matching

To avoid the issues with greedy alignment in P_{BERT}, we can extract one-to-one alignments between the two sequences. Let $C \in \mathbb{R}^{m \times k}$ denote the pairwise cosine distance matrix such that $C_{ij} = 1 - \cos(\hat{\mathbf{y}}_i, \mathbf{y}_j)$. For notational clarity, let $\widetilde{C} = 1 - C$. We wish to find alignments

$$T^d = \underset{T \in \{0,1\}^{m \times k}}{\arg\min} \sum_{i=1}^{m} \sum_{j=1}^{k} T_{ij} C_{ij}, \qquad (1)$$

such that no element in $\mathbf{h} = T\mathbf{1}_k$ and $\mathbf{r} = T^\top \mathbf{1}_m$ exceeds one. In other words, each $\hat{y}_i$ can align to at most one y_j (exactly one when $m = k$), and vice versa. This linear sum assignment problem can be solved in low-order polynomial time (Crouse, 2016), making it suitable for use during training.

Metric The updated precision is found as

$$P_{\text{DISC}} = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} T^d_{ij} \widetilde{C}_{ij}. \qquad (2)$$

Recall R_{DISC} takes an analogous form and is combined with P_{DISC} to produce an F1 score, F_{DISC}.

3.2 Continuous word alignment

We also experiment with soft alignments, where weights in T are continuous. In the case of P_{BERT}, one-to-many alignments between each hypothesis token $\hat{y}_i$ and those in $\{y_j\}_{j \in [k]}$ are permitted.

Inspired by work applying EMD to semantic text similarity (Kusner et al., 2015; Clark et al., 2019), we frame alignment as minimizing the transportation cost between token embeddings from the hypothesis and reference distributions. The amount of token-level mass to transport between the two distributions is $\mathbf{h}$ and $\mathbf{r}$, respectively. Instead of

assigning IDF as the mass per token (Zhao et al., 2019), we use the norm of its embedding (i.e., $\|\mathbf{y}\|$, Yokoi et al., 2020) for simplicity.

The EMD, or optimal transport, problem is

$$T^c = \underset{T \in \mathbb{R}^{m \times k}_{\geq 0}}{\arg\min} \sum_{i=1}^{m} \sum_{j=1}^{k} T_{ij} C_{ij}, \quad (3)$$
$$\text{s.t.} \quad \mathbf{h} = T\mathbf{1}_k, \ \mathbf{r} = T^\top \mathbf{1}_m.$$

Intuitively, if we view T_{ij} as the joint probability of aligning $\hat{y}_i$ with y_j, the row and column sums are marginals (Cuturi, 2013).

Metric To compute F_{CONT}, we normalize the alignment weights such that the rows of T sum to one for precision, and the columns for recall.

$$P_{\text{CONT}} = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{h_i} \sum_{j=1}^{k} T^c_{ij} \widetilde{C}_{ij}, \quad (4)$$

$$R_{\text{CONT}} = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{r_j} \sum_{i=1}^{m} T^c_{ij} \widetilde{C}_{ij} \quad (5)$$

4 Semantic Similarity Rewards

We propose to fine-tune on our optimized F1 metrics, applying a weighted average of cross-entropy and RL objectives. Given source sequence x (e.g., a linearized AMR), the former is computed as

$$\mathcal{L}_e = -\sum_{i=1}^{k} \log p(y_i \mid y_{<i}, x).$$

To encourage close evaluation scores between sampled $\bar{y}$ and reference y, the RL objective is

$$\mathcal{L}_r = (\Delta(\bar{y}_g, y) - \Delta(\bar{y}, y)) \sum_{i=1}^{k} \log p(\bar{y}_i \mid \bar{y}_{<i}, x),$$

where Δ is the chosen evaluation metric and $\bar{y}_g$ is a greedily decoded baseline relative to $\bar{y}$. This baseline helps reduce variance in REINFORCE (Williams, 1992). The combined cross-entropy and RL loss is

$$\mathcal{L} = \lambda \mathcal{L}_r + (1 - \lambda)\mathcal{L}_e,$$

where λ is empirically set to 0.3.

5 Experiments

We examine the performance of our proposed metrics as RL rewards on AMR-to-text generation.

	BLEU	METEOR	CHRF	BLEURT
XENT	36.37	39.94	65.68	56.30
BL-R	37.06	40.30	66.19	56.08
F_{BERT}	36.06	39.85	65.23	55.45
F_{CONT}	36.91	40.34	66.07	55.96
F_{DISC}	**37.65**	**40.61**	**66.55**	**57.01**

Table 1: Results on the AMR2017T10 test set.

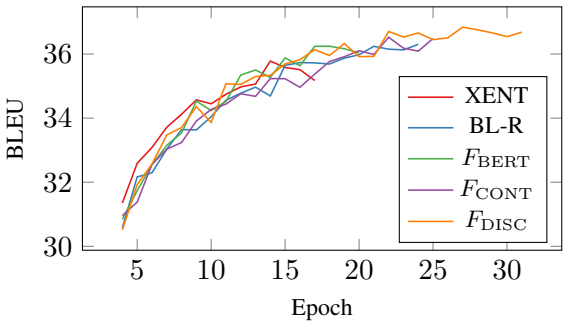

Figure 1: Development set BLEU during fine-tuning.

5.1 Setup

Dataset The LDC2017T10 dataset that we experiment on contains ~36K training and ~1.4K each of development and test AMR-sentence pairs. To leverage strong pre-trained language models, the AMRs are linearized as in Ribeiro et al. (2021).

Evaluation We report results in terms of BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CHRF (Popović, 2015), and BLEURT (Sellam et al., 2020). Only the latter metric makes use of pre-trained contextualized embeddings.

Baselines For all experiments, we fine-tune the small capacity T5 model (Raffel et al., 2020) from Ribeiro et al. (2021). The model has 60M parameters and features a Transformer-based encoder and decoder. We compare our F_{DISC} and F_{CONT} metrics for RL-based training against three baseline approaches. XENT is a pure cross-entropy objective. For RL-based approaches, we include a BLEU reward (BL-R) and one with F_{BERT}—computed on the lowest level token embeddings in T5.[1] The λ scaling factor for the RL objective is set to 0.3 across all RL-based experiments.

Implementation details *Adam* (Kingma and Ba, 2015) is used to optimize the model with an initial

[1]This also applies to F_{DISC} and F_{CONT}.

(1)	REF	There are **12 teams totally participating** in the competition.
	XENT	The competition was **part of a total of 12 teams**.
	F_{BERT}	The competition is **part of a total of 12 teams**.
	F_{DISC}	The **total of 12 teams participated** in competition.
(2)	REF	Raymond zilinskas stated that in the worst case the bacteria would be defrosted from minus 70 degrees and it would be a real mess to clean up afterward because **it would not be known for certain whether all the bacteria was dead**.
	XENT	Raymond Zilinskas stated that the bacterium was defrost in the worst case and that afterward cleaning up was a real mess because **there is certainly no known cause of death for all the bacteriums**.
	F_{BERT}	Raymond Zilinskas stated that the bacterium was defrosting in the worst case and the afterward cleaning up was a real mess because **the bacterium was certainly not known to die of all the bacteriums**.
	F_{DISC}	Raymond Zilinskas stated that the bacterium was defrost in the worst case and the afterward cleaning up was a real mess because **the bacterium was certainly not known to have all died**.

Table 2: Model-generated examples from three of the five explored systems.

learning rate of $1 \cdot 10^{-4}$ and a batch size of 16. Following Ribeiro et al. (2021), we use a linearly decreasing schedule for the learning rate and no warm-up. Since Ribeiro et al. (2021) do not release their training methodology, we train until validation BLEU does not increase for three epochs—an approach found in previous work fine-tuning T5 for AMR-to-text generation (Hoyle et al., 2021). We use SciPy[2] and the Python Optimal Transport library[3] to solve Eqs. 1 and 3.

5.2 Results

Table 1 shows that F_{DISC} achieves the highest scores on all metrics, surpassing F_{CONT} as well. It scores higher than XENT by 1.28 BLEU and 0.71 BLEURT points. Although BL-R was specially trained to optimize BLEU, F_{DISC} still outperforms it by over half a point on that metric.

There is a clear hierarchy among the approaches based on F1 score, with F_{DISC} above F_{CONT}, followed by F_{BERT} at the bottom. This dynamic suggests that the optimized alignments may provide higher quality reward signals during training.

We note that although F_{CONT} performed comparably to BL-R, it could exploit tensor operations and was far faster to compute than BLEU. On the other hand, F_{BERT} achieved significantly lower scores than BL-R. As noted in §2, perhaps the clipped precision counts in BLEU gave BL-R an advantage over the greedy nature of F_{BERT}.

5.3 Analysis

Training stability As shown in Fig. 1, F_{DISC} continues to improve on validation BLEU long after XENT overfits at epoch 18. This runs counter to the expectation of unstable RL-based training. It is also interesting that while F_{CONT} validation performance looks fairly low relative to BL-R, it achieves similar scores at test time. This may be due to irrelevant differences between the validation and test sets, however.

Manual inspection Table 2 lists a few examples of model outputs for detailed analysis. In example (1), both XENT and F_{BERT} make the error of predicting "part" instead of "participating". Only F_{DISC} approaches the meaning of the reference. This may be a side-effect of weighting lexical over semantic similarity in the former two systems. In (2), F_{BERT} repeats the word "bacterium", while XENT takes an anthropomorphic view of the bacterium. The repetition may be a result of F_{BERT} rewarding multiple instances of the same token by mistake during greedy alignment.

6 Conclusion

This paper proposes new F1 score metrics based on optimized rather than greedy alignments between predicted and reference tokens. Instead of letting hypotheses align to reference tokens without regard to their frequencies (and vice versa), we extract alignments as a constrained optimization problem. In the discrete case, we treat alignment as a matching problem between hypothesis and reference tokens. In the continuous case, we find alignments that minimize earth mover's distance between the two token embedding distributions.

We apply new metrics as rewards during RL-based training for AMR-to-text generation, with F_{DISC} outperforming both a cross-entropy baseline and one optimizing BLEU rewards. Despite being computed on a downstream model's token embeddings, the metrics still provide informative rewards during training without signs of overfitting.

[2] https://scipy.org
[3] https://pythonot.github.io

Acknowledgments Research supported by NSF awards IIS-1813823 and CCF-1934962.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

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

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760.

David F. Crouse. 2016. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26:2292–2300.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364.

Alexander Miserlis Hoyle, Ana Marasović, and Noah A Smith. 2021. Promoting graph awareness in linearized graph-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 944–956.

Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. 2021. BERTTune: Fine-tuning neural machine translation with BERTScore. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 915–924.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR-15)*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. 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 6038–6044.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51.

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

Gabriel Peyré and Marco Cuturi. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations*.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227.

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

Cédric Villani. 2009. *Optimal Transport: Old and New*. Springer, Berlin.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU: Training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word rotator's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578.