Avoiding Overlap in Data Augmentation for AMR-to-Text Generation

Wenchao Du
University of California, Santa Cruz
duwc2013@gmail.com

Jeffrey Flanigan
University of California, Santa Cruz
jmflanig@ucsc.edu

Abstract

Leveraging additional unlabeled data to boost model performance is common practice in machine learning and natural language processing. For generation tasks, if there is overlap between the additional data and the target text evaluation data, then training on the additional data is training on answers of the test set. This leads to overly-inflated scores with the additional data compared to real-world testing scenarios and problems when comparing models. We study the AMR dataset and Gigaword, which is popularly used for improving AMR-to-text generators, and find significant overlap between Gigaword and a subset of the AMR dataset. We propose methods for excluding parts of Gigaword to remove this overlap, and show that our approach leads to a more realistic evaluation of the task of AMR-to-text generation. Going forward, we give simple best-practice recommendations for leveraging additional data in AMR-to-text generation.[1]

1 Introduction

Deep learning has made remarkable progress in many areas of natural language processing, including language generation (Sutskever et al., 2014; Luong et al., 2015) and semantic parsing (Dong and Lapata, 2016). Nevertheless, neural models are usually data-hungry, and sophisticated use of data augmentation can often go a long way (Konstas et al., 2017; Wang et al., 2018; Du and Black, 2019; Wei and Zou, 2019). One common method of data augmentation is to leverage large amounts of out-of-domain data for semi-supervised learning. However, without proper examination of the data being used, the external data may contain significant overlap with the test set, leading to unfair gains as a result. This issue is a unique problem for natural language generation (NLG) tasks with data augmentation, because training with data that overlaps with the test set is akin to training on the answers. In this work, we study the task of AMR-to-text generation and scrutinize the datasets used for training and evaluation. Our contributions are two-fold: (1) we develop an examination procedure to confirm that there are serious overlaps between one of the AMR datasets and Gigaword (Parker et al., 2011), and conduct experiments showing that some of the performance gains are indeed "unfair"; (2) we propose several strategies to apply when collecting external data for training, and empirically show that these strategies can mitigate the aforementioned unfair gains. For best practice, we suggest future work on AMR-to-text generation exclude Gigaword articles that are written in the nearby months of those covering Proxy to be on the safer side (strategy `no-3Months` described in Section 5).

2 Related Work

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) has gained growing interest as a semantic formalism. The first AMR-to-text generator was developed using tree transducers (Flanigan et al., 2016). More recent work heavily adopted neural models, explored different architectures, and commonly employed Gigaword data to boost results (Konstas et al., 2017; Song et al., 2018; Wang et al., 2020). The most common approach is to use JAMR (Flanigan et al., 2014) to bootstrap labels for the additional data and then add them to the training data.

Prior work on AMR generation has used automatic metrics such as BLEU (Papineni et al., 2002) and human evaluations (May and Priyadarshi, 2017). Currently, there is increased research on evaluation metrics for NLG (Zhang et al., 2019;

[1] Our code for these best practices is available at https://github.com/jlab-nlp/amr-clean.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1043–1048

Dataset	# Sentences	Domain
Bolt	133	Web
Consensus	100	News
DFA	229	Web
Proxy	823	News
Xinhua	86	News

Table 1: The number of test sentences and domain of each AMR dataset. Note that LDC2015E86 and LDC2017T10 have identical test sentences.

Sellam et al., 2020, inter alia). However, we are not aware of prior work investigating the problem of test set overlap when using data-augmentation methods for generation. Closest to our work is prior practice in machine translation evaluation of excluding articles from the same time period as the test set (NIST, 2012).

3 Origin of AMR and Gigaword Overlap

In this section, we describe the reason for the overlap between the AMR dataset and Gigaword. In standard LDC releases of AMR, for example LDC2015E86 and LDC2017T10, the dev and test set consist of 5 datasets from different sources. Information about these datasets are listed in Table 1. Each sentence in the dev and test set is associated with an ID. The sentences of the Proxy dataset, in particular, have IDs that can be traced back to Gigaword articles. Upon inspection, these sentences appear to originate as close edits of sentences in Gigaword. For example, the sentence with ID "PROXY_LTW_ENG_20070831_0072.1" is originated from the Gigaword article with ID "LTW_ENG_20070831". The date on which a Gigaword news article was written is included in the ID. Since Proxy takes up more than half of the test sentences, such overlap could have a high impact on the evaluation of AMR-to-text generators. In the next section, we describe our procedure to empirically examine the effect of overlap between Proxy and Gigaword.

4 Measuring Overlap

We use the following procedure to quantitatively examine the overlap between Proxy and Gigaword dataset. For each Proxy sentence in the validation and test split, we find the Gigaword article whose ID is associated with the Proxy sentence ID. Then we tokenize and split the article into sentences. We measure the overlap between the Proxy sentence and each of the Gigaword sentences with 3 different metrics: (1) absolute count of common words, which is the number of distinct words that appear in both sentences, (2) BLEU score, and (3) ROUGE-L score.

	Mean	Median
Count 1st	13.85	13.0
Count 2nd	7.87	8.0
Count 3rd	7.16	7.0
ROUGE 1st	0.64	0.68
ROUGE 2nd	0.33	0.35
ROUGE 3rd	0.29	0.32
BLEU 1st	0.39	0.36
BLEU 2nd	0.07	0.05
BLEU 3rd	0.04	0.01

Table 2: The mean and median of the 3 highest scores for word count, BLEU, and ROUGE.

5 Exclusion Strategies

We propose and investigate 3 sampling strategies for constructing semi-supervised training datasets from Gigaword, and these strategies differ by how to exclude certain Gigaword articles: `no-ID` excludes articles whose id appeared in the proxy dataset; `no-Month` excludes articles that are written in the same month as those excluded by `no-ID` ; `no-3Months` excludes articles that are written in the same month or neighboring months from those excluded by `no-ID` . We use reservoir sampling (Vitter, 1985) to sample sentences from Gigaword. We first collect a set with 200k sentences without any exclusion as a baseline. We then filter out sentences that are from articles excluded by `no-ID` , and sample same number of sentences as those being filtered from articles that are included by `no-ID` . This yields a set of 200k sentences representing `no-ID` . We collect the sample sets for `no-Month` and `no-3Months` based on the baseline set in a similar fashion.[2]

We use the GGNN-dual-encoder model by (Ribeiro et al., 2019) as our model to study the effects of different exclusion strategies. For each exclusion strategy, we obtain 3 different samples using different random seeds and repeat experiments. We keep most of the hyperparameters from the original paper. We adjusted the learning rate schedule to accommodate larger sets of training

[2]Our code for doing this filtering is available on our GitHub repository.

	Sentence	Score
Count 1st	At least one of those bands appears to be splitting into at least two different groups.	13
Count 2nd	Even though the Bush White House has generally entrusted government agencies to officials ...	7
Count 3rd	The rentals violated U-Haul's rule requiring the tow vehicle to be at least 750 pounds heavier than the one being towed.	7
Bleu 1st	At least one of those bands appears to be splitting into at least two different groups.	0.70
Bleu 2nd	At least one of those inspections would have come at a particularly delicate time ...	0.20
Bleu 3rd	... as well as other outside organizations, at least one of which then sold tickets to its own members.	0.19
Rouge 1st	At least one of those bands appears to be splitting into at least two different groups.	0.91
Rouge 2nd	For at least a few of those percentage points, we have to thank Sheehan.	0.44
Rouge 3rd	At least one Democratic member of the group questioned Giuliani's decision to quit.	0.4

Table 3: Examples of top matches found in Gigaword with test set sentence "At least one of those bands appears to be splitting into different groups."

	No Extra Data	Top 1 (Cheat)	Top 2 to 4	Top 5 to 7	Top 8 to 10
Overall	27.58	32.71	31.67	30.82	30.85
Bolt	17.36	18.59	18.54	18.80	20.36
Consensus	20.18	21.50	22.73	21.90	22.58
Dfa	21.45	22.86	24.39	23.05	22.87
Proxy	31.56	39.12	36.85	35.75	35.68
Xinhua	25.22	24.22	26.03	27.16	25.90

Table 4: Evaluation results (BLEU) when the model is trained on cheat set and other highly overlapping sets.

	No Extra Data	Cheat	Baseline Strategy	`no-ID`	`no-Month`	`no-3Months`
Overall	24.32	30.73	32.72	32.69	31.83	32.35
Bolt	15.11	15.31	19.85	20.21	18.98	19.79
Consensus	17.04	16.47	25.02	20.80	24.55	24.00
Dfa	18.21	17.95	20.14	21.50	20.34	19.96
Proxy	29.33	38.16	38.52	38.46	37.33	37.99
Xinhua	23.01	22.52	32.21	31.82	31.08	32.65

Table 5: Results (BLEU) on LDC2015E86. Average of 3 experiments are reported.

	No Extra Data	Cheat	Baseline Strategy	`no-ID`	`no-Month`	`no-3Months`
Overall	27.58	32.71	34.46	33.53	33.44	33.16
Bolt	17.36	18.59	21.37	21.20	22.66	19.7
Consensus	20.18	21.50	25.96	27.18	26.44	25.06
Dfa	21.45	22.86	24.78	22.81	24.79	23.61
Proxy	31.56	39.12	39.81	38.84	38.09	38.39
Xinhua	25.22	24.22	32.59	31.68	31.77	32.40

Table 6: Results (BLEU) on LDC2017T10. Average of 3 experiments are reported.

data. With sample sets of 200k sentences, each experiment takes 3 days to finish on a Tesla V100.

6 Results

6.1 Overlap between Proxy and Gigaword

In this section, we measure the overlap between Proxy and Gigaword using word and n-gram overlap evaluation measures, and study the effect of the overlap on the final trained system. We list the mean and median of the 3 sentences with highest overlap scores for each overlap measure in Table 2. It is clear that sentences with the top overlap score overlap significantly more than those sentences at the 2nd and 3rd place. Examples for illustration are given in Table 3. All three metrics tend to find the same top matching sentence. Most of the time, the test sentence in Proxy is an extractive summarization or rephrase of the top match in Gigaword, indicating a concerning overlap between Proxy and Gigaword.

To investigate the impact of semi-supervised training with these Gigaword sentences that are close duplicates of the test set, we create various sets for semi-supervised training. We create a cheat set using sentences with highest matching ROUGE scores, called Top 1 (Cheat). We are also interested in the impact of sentences from the same article as these duplicates, but with less overlap. We create additional sets with those that have top 2-4 overlap scores, top 5-7 overlap scores, etc. We trained the model with these sample sets for semi-supervised training, and the results on LDC2017T10 are listed in Table 4. The cheat set helped the evaluation on Proxy by more than 7 points, but only helped other datasets by about 1 point, if not hurting. As the matching scores decrease, the improvement on Proxy also went down. This indicates that the overlap sentences between Proxy and Gigaword give a significant unfair advantage, especially for the sentences with highest overlap.

6.2 Exclusion Strategies for Gigaword

To find a good exclusion strategy for constructing semi-supervised datasets from Gigaword, we sample semi-supervised training sets as described in Section 5 and ran experiments. The results on LDC2015E86 and LDC2017T10 are presented in Table 5 and 6, respectively. The results on LDC2017T10 is generally better than LDC2015E86, since the size of training of the former is larger than that of the later. Without excluding (i.e. baseline strategy), the results on Proxy are significantly better than no additional semi-supervised data (by about 8 points on LDC2017T10 and 10 points on LDC2015E86). It is also slightly better than being trained with the cheat set. This is because training on sample sets of size 200k yields much better language model than the small cheat set. On the other hand, training on the cheat set is almost as good as training on 200k additional data, since neural models are good at memorization. For LDC2017T10, filtering out articles covering Proxy test sentences decreases performance on Proxy by 1 point; excluding articles written in the same month and nearby months further decreases results on Proxy by more than 0.5 points. For LDC2015E86, excluding articles written in the same month decreases results on proxy by more than 1 point.

	Proxy	All Other
LDC2015/no-ID	0.400	0.379
LDC2015/no-Month	**0.045**	0.200
LDC2015/no-3Months	0.387	0.192
LDC2017/no-ID	0.202	0.357
LDC2017/no-Month	**0.002**	0.129
LDC2017/no-3Months	**0.047**	0.100

Table 7: P-values from statistical tests comparing system performance against baseline sampling. Significant results at $p = .05$ are highlighted.

Finally, we perform statistical tests with a paired t-test for comparing performance of systems trained on different sample sets against the baseline (no filtering). See Table 7. For LDC2015E86, `no-Month` resulted in lower BLEU scores on proxy dataset that are statistically significant; for LDC2017T10, both `no-Month` and `no-3Months` resulted in lower BLEU scores on proxy and the differences are statistically significant. All strategies performed similarly on all other datasets. This shows that the exclusion of certain overlapping articles in Gigaword has significant impact on the evaluation on Proxy dataset, but less so on the rest.

7 Conclusion and Recommendation

In this paper, we examined Gigaword, the commonly used dataset for improving AMR-to-text generation, and found sentences that almost duplicate the test set of Proxy, one of the AMR datasets. We developed a procedure that utilizes a word overlap measure to find overlapping sentences, and

found several metrics that may be good at finding duplicating sentences. We proposed 3 different strategies for excluding overlapping data from Gigaword, and validated the idea that without filtering certain articles, the evaluation results may be unfair. For best practice, we suggest future work on AMR-to-text generation exclude Gigaword articles that are written in the nearby months of those covering Proxy to be on the safer side (`no-3Months`). Additionally, we suggest future work report results on each AMR dataset separately so that techniques favoring one dataset can be detected.

Acknowledgments

This research was supported in part by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805. The opinions expressed are those of the authors and do not represent views of the NSF.

References

NIST Open Machine Translation 2012 Evaluation Plan (OpenMT12). https://www.nist.gov/system/files/documents/itl/iad/mig/OpenMT12_EvalPlan.pdf. Accessed: 2020-02-01.

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.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43.

Wenchao Du and Alan W Black. 2019. Boosting dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 38–43.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime G Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 731–739.

Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Jonathan May and Jay Priyadarshi. 2017. Semeval-2017 task 9: Abstract meaning representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 536–545.

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.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. *Google Scholar*.

Leonardo FR Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing amr-to-text generation with dual graph representations. 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 3174–3185.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27:3104–3112.

Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. Amr-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.