

Neural Text Generation from Rich Semantic Representations

Valerie Hajdik¹, Jan Buys², Michael W. Goodman¹ and Emily M. Bender¹

¹Department of Linguistics, University of Washington

²Paul G. Allen School of Computer Science and Engineering, University of Washington

vhajdik@uw.edu, jbuys@cs.washington.edu,
{goodmami, ebender}@uw.edu

Abstract

We propose neural models to generate high-quality text from structured representations based on Minimal Recursion Semantics (MRS). MRS is a rich semantic representation that encodes more precise semantic detail than other representations such as Abstract Meaning Representation (AMR). We show that a sequence-to-sequence model that maps a linearization of Dependency MRS, a graph-based representation of MRS, to English text can achieve a BLEU score of 66.11 when trained on gold data. The performance can be improved further using a high-precision, broad coverage grammar-based parser to generate a large silver training corpus, achieving a final BLEU score of 77.17 on the full test set, and 83.37 on the subset of test data most closely matching the silver data domain. Our results suggest that MRS-based representations are a good choice for applications that need both structured semantics and the ability to produce natural language text as output.

1 Introduction

Text generation systems often generate their output from an intermediate semantic representation (Yao et al., 2012; Takase et al., 2016). However many semantic representations are task- or domain-specific (He and Young, 2003; Wong and Mooney, 2007), while rule-based text generation systems often have incomplete coverage (Langkilde-Geary, 2002; Open et al., 2007).

In this work we combine the advantages of Minimal Recursion Semantics (MRS; Copestake et al., 2005) with the robustness and fluency of neural sequence-to-sequence models trained on large datasets. We hypothesize that MRS is particularly well-suited for text generation, as it is explicitly compositional, capturing the contribution to sentence meaning of all parts of the surface form (Bender et al., 2015).

In contrast, semantic representations such as Abstract Meaning Representation (AMR; Banarescu et al., 2013) seek to abstract away from the syntax of a sentence as much as possible. Therefore MRS captures meaning distinctions that AMR fails to represent (see Fig. 1).

Our approach (§2) uses neural sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2014) to map linearizations of directed acyclic graphs (DAGs) to text, similar to the approach proposed by Konstas et al. (2017) to generate text from AMR. We use Dependency MRS (DMRS; Copestake, 2009), a graph-based representation in which nodes are MRS predicates (annotated with additional attributes) and edges represent relations between predicates. MRS and DMRS are interconvertible and the graph-based representation enables more convenient linearization and manipulation than MRS’s variable-based representation (Copestake et al., 2016).

Results (§3) show that neural DMRS to English text generation can obtain up to 83.37 BLEU and 32% exact match, substantially higher than previous work. In particular, we obtain an 11.6 BLEU improvement through semi-supervised training using the output of a grammar-based parser, compared to training on gold data only. In comparison a grammar-based generator obtained 62.05 BLEU, and an approach based on DAG Transducers (Ye et al., 2018) 68.07 BLEU. Ablation experiments show that node attributes encoding fine-grained morpho-semantic information such as number and tense contribute more than 12 BLEU points. The highest reported result for AMR generation is 33.8 BLEU (Konstas et al., 2017); on the same dataset our best model obtains 75.8 BLEU. While a more detailed meaning representation is harder to produce, our results suggest that MRS could be suitable for text generation applications where precise semantic representations are required.

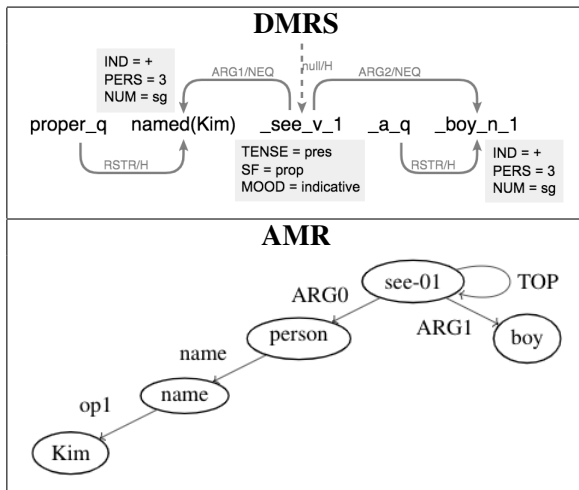


Figure 1: DMRS and AMR graphs for the sentence *Kim sees a boy*. Because DMRS includes tense and number, and has a node for the determiner, it can distinguish between, e.g. *Kim sees a boy* and *Kim saw the boys*, which AMR does not do.

2 Approach

2.1 Data

Our gold training data are parallel MRS and English text corpora, derived from the 1214 release of the Redwoods Treebank (Oepen et al., 2002).¹ MRS is implemented as the semantic layer of the English Resource Grammar (ERG; Flickinger, 2000, 2011), a broad-coverage, hand-engineered computational grammar of English. The Redwoods annotation was produced in conjunction with the ERG by parsing each sentence into a forest (discarding unparseable sentences), followed by manual disambiguation (Flickinger et al., 2017).

About half of the training data comes from the Wall Street Journal (sections 00-21), while the rest spans a range of domains, including Wikipedia, e-commerce dialogues, tourism brochures, and the Brown corpus. The data is split into training, development and test sets with 72,190, 5,288, and 10,201 sentences, respectively.

2.2 Graph linearization

We use PyDelphin² to convert MRS annotations to DMRS. In order to apply sequence-to-sequence models to graph-to-text generation, we then linearize the DMRS into PENMAN format (which is also used to represent AMR). We follow Goodman (2018, pp. 82–86) in finding normalized spanning

¹<http://svn.delph-in.net/erg/tags/1214/tsdb/gold>

²<https://github.com/delph-in/pydelphin>

PENMAN

```
(10002 / _see_v_1
 :tense PRES
 :sf PROP
 :perf -
 :mood INDICATIVE
 :ARG1-NEQ (10001 / named
  :carg "Kim"
  :pers 3
  :num SG
  :ind +)
 :ARG2-NEQ (10004 / _boy_n_1
  :pers 3
  :num SG
  :ind +
  :RSTR-H-of (10003 / _a_q))
```

Linearization

```
( _see_v_1 mood=INDICATIVE|perf=-|sf
 ↪ =PROP|tense=PRES ARG1-NEQ (
 ↪ named0 ind=+|num=SG|pers=3 )
 ↪ ARG2-NEQ ( _boy_n_1 ind=+|num=
 ↪ SG|pers=3 RSTR-H-of ( _a_q ) )
 ↪ )
```

Figure 2: The DMRS for the sentence *Kim sees a boy*. in PENMAN format (top) and the linearization used by our model (bottom).

trees through depth-first traversal over the directed acyclic DMRS graphs.³ The PENMAN format defines each node once, supports node attributes and edge labels, marks edges whose direction is reversed in the traversal, and represents edges which are not covered by the spanning tree.

The PENMAN format is processed further to obtain a linearization appropriate as input to sequence-to-sequence models, similar to the approach proposed by Konstas et al. (2017) for AMR linearization (see Fig. 2). Node variable identifiers are removed, node attributes are concatenated, and named entities are anonymized. Predicates that appear only once in the training data are treated as unknowns. Preprocessing and unknown word handling are described in greater detail in Appendices A and B.

2.3 Model

Our neural generator follows the standard encoder-decoder paradigm (Bahdanau et al., 2014). The encoder is a two-layer bidirectional LSTM. Predicates and their attributes are embedded separately; their embeddings are then concatenated (Sennrich and Haddow, 2016). The

³<https://github.com/goodmami/mrs-to-penman>

decoder uses global soft attention for alignment (Luong et al., 2015), and pointer attention to copy unknown tokens directly to the output (Gulcehre et al., 2016). The models are trained using Adam (Kingma and Ba, 2014). Dropout is applied to non-recurrent connections. Decoding uses beam search (width 5). The generator is implemented using OpenNMT-py (Klein et al., 2017). Hyperparameter details are given in Appendix C. Our code is available online.⁴

2.4 Semi-supervised training

We augment the gold training data with a *silver* dataset generated using ACE,⁵ a parser for the ERG, to parse sentences to MRS. We sample one million sentences from the Gigaword corpus (Parker et al., 2011), restricted to articles published before the year 2000, to match the domain of the Wall Street Journal data. The parser failed to parse about 10.3% of the Gigaword sentences, so these were discarded. While there are robust MRS parsers (Buys and Blunsom, 2017; Chen et al., 2018), the MRSs they produce are less accurate and not guaranteed to be well-formed. Our approach thus differs from Konstas et al. (2017), who used self-training to improve AMR to text generation by iteratively training on larger amounts of data parsed by their neural parser.⁶

3 Results

We compare the performance of our neural generator when trained on either gold, silver, or gold and silver data (Table 1). Generation quality is primarily evaluated with BLEU (Papineni et al., 2002), using SacreBLEU (Post, 2018).⁷ We evaluate the neural models on both the full Redwoods test set ('All') and the WSJ subset.

The results show that our neural generator obtains very strong performance. Semi-supervised training leveraging the ERG parser leads to an 11 BLEU point improvement on Redwoods, comparing to supervised training only. We found that the best semi-supervised results are obtained by up-

⁴<https://github.com/shlurbee/dmrs-text-generation-naacl2019>

⁵ACE version 0.9.25, with the 1214 ERG release, available at <http://sweaglesw.org/linguistics/ace>

⁶The ACE parser obtained 93.5 Smatch score on parsable sentences (Buys and Blunsom, 2017), while the neural AMR parser (Konstas et al., 2017) obtained 62.1 Smatch (on a different domain).

⁷<https://github.com/mjpost/sacreBLEU>

sampling the gold data so that the gold to silver ratio in training examples is 1:2. Interestingly, training on silver data performs only slightly worse than training on both gold and silver.

3.1 Baselines

Our baselines are the ERG's grammar-based generator (Carroll et al., 1999; Carroll and Oepen, 2005) and the DAG transducer generator of Ye et al. (2018). To compare our models against the grammar-based generator, implemented in ACE, we need to restrict the evaluation to examples from which ACE is able to generate ('All overlap').⁸ In addition to BLEU, we also report exact match accuracy on the overlapping subset.

Results show that our neural models outperform the grammar-based generator by a large margin. ACE ranks candidate generations with a discriminative ranker based on structural features over its derivations (Velldal and Oepen, 2006). However, it does not use a language model trained on large amounts of text, which would likely improve fluency substantially.

The DAG transducer was trained to generate from Elementary Dependency Structures (EDS; Oepen and Lønning, 2006), an MRS-derived representation almost equivalent to DMRS (after edge properties are removed, which Table 3 shows has an effect of less than 1 BLEU point). It was evaluated against the same WSJ test set reference generations, but trained using both less gold data (only the WSJ subsection) and less silver data (300K vs 900K sentences). Our model trained on WSJ gold data performs only slightly worse (65.78 BLEU; see Table 2) and all our semi-supervised models obtain substantially higher results.

3.2 Out of domain evaluation

We evaluate the in- and out-of-domain performance of our approach by training models on either WSJ gold data only, or both WSJ gold data and Gigaword silver data, and evaluating on different domains. The results in Table 2 show that while the generator performs best on test data which matches the training domain (news), semi-supervised training leads to substantial out-of-domain improvements on the Wikipedia and the Brown corpus portions of the test set.

⁸Despite all test sentences being *parsable* by the ERG, there are gaps in generation coverage, primarily because ACE is unable to generate words outside the grammar's vocabulary.

Model	BLEU (All)	BLEU (WSJ)	BLEU (All overlap)	Exact Match%	Coverage%
Neural MRS (gold)	66.11	73.12	69.27	24.09	100
Neural MRS (silver)	75.43	81.76	77.13	25.82	100
Neural MRS (gold + silver)	77.17	83.37	79.15	32.07	100
ACE (ERG)	–	–	62.05	15.08	78
DAG transducer (Ye et al., 2018)	–	68.07	–	–	100

Table 1: BLEU and exact-match scores over held-out test set

Test domain	Training Data	
	WSJ	WSJ + Giga
WSJ	65.78	83.42
Brown	45.00	76.99
Wikipedia	35.90	62.26

Table 2: BLEU scores for domain match experiments

Ablation	BLEU
All attributes	72.06
No node attributes	59.37
No node attr except num, tense	67.34
No edge features	71.27

Table 3: Results of semantic feature ablation, model trained with gold data only

3.3 Attribute ablations

To understand which elements of MRS contribute most to our generator’s performance, we ablate node (predicate) and edge attributes from both the training and test DMRS graphs (Table 3). In the training data, number and tense show the most variation among node attributes, and subsequently have the largest effect on the reported BLEU score. The most common value for number is SG, but 62.36% of sentences contain a node with PL. Similarly, 42.41% of sentences contain a tense value other than PRES or UNTENSED. Many other attributes are less informative: Mood has a value other than INDICATIVE in only 0.38% of sentences, and perf is + in just 9.74% of sentences. Edge features (including H, EQ and NEQ) encode constraints on scopal relationships (see Copestake 2009). Removing them, which makes the DMRS representation close to equivalent to EDS, has only a small impact on performance.

Representation	Train on Gold	Train on Gold+Silver
AMR	22.0	33.8
DMRS - no attributes	40.1	63.6
DMRS - all attributes	56.9	75.8

Table 4: BLEU scores for evaluating AMR and DMRS generators on an AMR test set

3.4 Comparison with AMR generation

We compare our approach to AMR-to-text generation by evaluating our generator on a standard AMR test set (LDC2015E86). As we do not have manually verified MRSes available on this test set, we use ACE to parse the reference sentences to silver MRSes. We then evaluate the outputs that our generator produces from those MRSes. About 20% of the examples could not be parsed by ACE, and are discarded for the MRS evaluation. We compare our generator to the neural AMR generator of Konstas et al. (2017) for models trained on gold as well as gold plus silver data.⁹

We evaluate DMRS models both with and without predicate and edge attributes, as these attributes contain information that is absent from AMR.¹⁰ The results in Table 4 show that our MRS generator performs better than the AMR generator by a large margin, even when the additional MRS attributes are excluded. Our system results are reported on the subset for which we obtained MRS parses. AMR results are as given by Konstas et al. (2017) and cover the entire test set.

⁹The AMR and DMRS systems have different gold training data, but the same source of silver data.

¹⁰Recently, Donatelli et al. (2018) proposed adding tense and aspect to AMR, but this annotation is not yet available in a large AMR corpus.

Type	B80-89	B60-69	B40-49	All
Unproblematic	56.4	39.55	48.8	47.1
Slightly problematic	18.0	9.2	3.3	7.6
Moderately problematic	12.8	25.0	18.7	19.8
Ungrammatical	5.1	7.9	8.1	7.6
Other serious error	7.7	18.4	21.1	18.1
Number of errors	39	76	123	238
Errors per item	1.18	2.30	3.73	7.21

Table 5: Percentage of errors of each type, across 99 sampled items, grouped by BLEU score

3.5 Error analysis

We sampled 99 items for error analysis from the dev set, 33 each from among sentences with sentence-level BLEU scores of 80-89, 60-69, and 40-49.¹¹ We identified all differences between these strings and the reference strings and then labeled each difference with a fine-grained error type.¹² We classified the differences into 238 errors, distributed across five levels of severity (Table 5).

Almost half of the differences (47.1%) were unproblematic, including spelling variants, meaning-preserving punctuation variation and grammatical alternations (such as optional *that* or auxiliary contraction as in (1)). The slightly problematic category includes close synonyms (e.g. *sometime v. someday*), spelled out number names where Arabic numerals are preferred, and differences in formatting. The next more serious category (moderately problematic) includes meaning-changing differences in punctuation, tense or aspect, and minor grammatical errors such as swapping *who* and *which* in relative clauses or *a v. an*.

- (1) a. I think I would like a Sony. [sys.]
b. I think I'd like a Sony. [ref.]

Finally, among the most serious errors, we find cases where the generator provided ungrammatical output or grammatical output not conveying the correct semantics. The former include spurious additional tokens, ungrammatical word orders, and ungrammatical inflection. Serious errors that nonetheless resulted in grammatical strings include meaning-changing dropped or swapped

¹¹Items with BLEU scores lower than 40 tend to be very short and primarily involve formatting differences.

¹²This was done by a single annotator only. The labels were generated bottom up, with new labels added as needed in the course of annotation.

tokens (as in (2)), spurious additional tokens, and word order changes that alter the semantics or pragmatics of the string.

- (2) a. For such cases, machine learning techniques emulate human linguistics and learn from training examples to predict future events. [sys.]
b. For such cases, machine learning techniques emulate human cognition and learn from training examples to predict future events. [ref.]

In summary, we find that the BLEU scores underestimate the quality of system outputs, due to unproblematic differences (N=112) and differences, like formatting markup (N=6), not reflected in the input semantic representations. Among the 108 moderate to serious differences, about a third (35) involve punctuation, suggesting that meaning signalled by punctuation could be better reflected in the semantic representations. About half (52) involve added, dropped, or swapped tokens, showing room for improvement in the generator's ability to learn appropriate connections between semantic predicates and surface forms. The remainder (21) involve inflection, grammatical alternations (such as *who/which*) and word order constraints, showing room for improvement in mimicking grammatical processes.

4 Conclusion

We have shown that neural sequence-to-sequence models can be used to generate high quality natural language text from Minimal Recursion Semantics representations, in contrast to both existing MRS-based generators and neural generators based on other broad-coverage semantic representations. Furthermore, we have demonstrated that a large hand-crafted grammar can be leveraged to produce large training sets, which improves performance of neural generators substantially. Therefore we argue that the ability to generate high quality text from MRS makes it a good choice of representation for text generation applications that require semantic structure. For future work, we are interested in applying graph-to-sequence neural networks (Beck et al., 2018; Song et al., 2018) to MRS-to-text generation.

Acknowledgements

Thanks to Yannis Konstas for sharing preliminary results on DMRS generation, and Swabha Swayamdipta for discussions. This research was supported in part by NSF (IIS-1524371) and Samsung AI Research.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- 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.
- Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283.
- Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann A. Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th international conference on Computational Semantics*, pages 239–249.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. O’Reilly Media, Inc.
- Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1215–1226.
- John Carroll, Ann Copestake, and Dan Flickinger. 1999. An efficient chart generator for (semi-) lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 86–95.
- John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *International Conference on Natural Language Processing*, pages 165–176. Springer.
- Yufei Chen, Weiwei Sun, and Xiaojun Wan. 2018. [Accurate shrg-based semantic parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 408–418, Melbourne, Australia.
- Ann Copestake, Guy Emerson, Michael Wayne Goodman, Matic Horvat, Alexander Kuhnle, and Ewa Muszyska. 2016. Resources for building applications with dependency minimal recursion semantics. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portoro, Slovenia.
- Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. [Minimal recursion semantics: An introduction](#). *Research on Language and Computation*, 3(2):281–332.
- Ann A. Copestake. 2009. Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 1–9.
- Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential amr. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions*, pages 96–108.
- Dan Flickinger. 2000. [On building a more efficient grammar by exploiting types](#). *Nat. Lang. Eng.*, 6(1):15–28.
- Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. In E.M. Bender and J.E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage, and Processing*, pages 31–50. CSLI Publications, Stanford.
- Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. [Sustainable development and refinement of complex linguistic annotations at scale](#). In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 353–377. Springer Netherlands, Dordrecht.
- Michael Wayne Goodman. 2018. *Semantic Operations for Transfer-based Machine Translation*. Ph.D. thesis, University of Washington.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.
- Yulan He and Steve Young. 2003. A data-driven spoken language understanding system. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 583–588. IEEE.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1701.02810*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. [Moses: Open source toolkit for statistical machine translation](#). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion*

- Volume: *Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.
- 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.
- Irene Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the International Natural Language Generation Conference*, pages 17–24.
- 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.
- Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based mrs banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, Genoa, Italy.
- Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The lingo redwoods treebank: Motivation and preliminary applications. In *The 17th International Conference on Computational Linguistics: Project Notes*.
- Stephan Oepen, Erik Velldal, Jan Tore Lønning, Paul Meurer, Victoria Rosén, and Dan Flickinger. 2007. Towards hybrid quality-oriented machine translation. on linguistics and probabilities in mt. In *In Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation*.
- 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, Philadelphia, Pennsylvania.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition ldc2011t07. DVD. Philadelphia: Linguistic Data Consortium.
- M. Post. 2018. *A Call for Clarity in Reporting BLEU Scores*. *arXiv preprint arXiv:1804.08771*.
- Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91.
- 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. In *Advances in neural information processing systems*, pages 3104–3112.
- Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1054–1059, Austin, Texas.
- Erik Velldal and Stephan Oepen. 2006. Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 517–525, Sydney, Australia.
- Yuk Wah Wong and Raymond Mooney. 2007. Generation by inverting a semantic parser that uses statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–179.
- Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based question generation and implementation. *Dialogue & Discourse*, 3(2):11–42.
- Yajie Ye, Weiwei Sun, and Xiaojun Wan. 2018. Language generation via dag transduction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1928–1937.

A Preprocessing Details

We preprocess the text by removing special formatting characters such as HTML tags, converting all double quotations to the same format (i.e., ” instead of ' ') and removing double brackets ‘[[’ and ‘]]’ that represent hyperlinks in Wikipedia data. These formatting characters are somewhat arbitrary and represent content (markup) that is orthogonal to the meaning of the text, so would be difficult for a model (or a human) to predict from a semantic representation for the text alone (such as MRS). After preprocessing and normalization, sentences in the gold and silver training data identical to sentences in the test set were removed.

Named entities (as annotated in MRS) are anonymized according to the conventions described in Konstas et al. (2017) prior to training. The purpose of anonymization is to reduce the sparsity of tokens such as dates and named entities that, while having a different surface form almost every time they appear in the text, should be

treated similarly by the model. These tokens are replaced, both in the DMRS graph and in the raw sentence text, with numbered placeholders like *named0*, *named1*, and *month*.

Sentences are tokenized using the NLTK's implementation (Bird et al., 2009) of Moses-style tokenization (Koehn et al., 2007). During post-processing, generated sentences are de-anonymized (by replacing the placeholder with the original surface form) and de-tokenized, reversing the NLTK tokenization.

B Unknown Word Handling

We use a combination of two methods to deal with unknown words: anonymizing words to a special *UNK0* token and using the pointer attention to copy the token with highest attention value directly from the source to the target (Gulcehre et al., 2016). There are two types of unknown words in our setup: words that are not recognized by the model because they do not appear in the training data and words that may or may not appear in the training data that are not recognized by the ERG because they are not part of its lexicon. Inspection shows that words missing from the ERG lexicon are often misspellings, uncommon proper names, or domain-specific terminology.

For words that are not part of the ERG lexicon, if the word appears only once in the training data, we replace it with *UNK0*. Otherwise, if the word appears multiple times in the training set, we include its surface form (instead of a predicate) as the node token to enable the model to learn from it. In early experiments, we also tried either anonymizing all unknowns or including all unknowns (relying on the pointer copy mechanism), and found that anonymizing singletons gave best performance while also keeping the vocabulary size reasonable.

C Model Details

The encoder and decoder each have two 800-dimension Bidirectional LSTM layers. The main input embedding layer has 500 dimensions and the size of the attribute embedding layer varies between 22 and 25 depending on the number of attribute combinations in the training data. We train using negative log likelihood loss, optimizing using Adam with initial learning rate = 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For regularization, we apply dropout of 0.3 to the non-recurrent connections of

the LSTM layers and after the attention layer. The number of layers, per-layer hidden state size, and dropout rates were all chosen empirically to minimize validation perplexity.