Legal Knowledge and Information Systems
M. Araszkiewicz and V. Rodríguez-Doncel (Eds.)

doi:10.3233/FAIA190330

Legal Text Generation from Abstract Meaning Representation

Sinh Trong VU [a] Minh Le NGUYEN [a] and Ken SATOH [b]
[a] *School of Information Science, Japan Advanced Institute of Science and Technology*
[b] *National Institute of Informatics, Japan*

Abstract.
Generating from Abstract Meaning Representation (AMR) is a non-trivial problem, as many syntactic decisions are not constrained by the semantic graph. Current deep learning approaches in AMR generation almost depend on a large amount of "silver data" in general domains. While the text in the legal domain is often structurally complicated, and contain specific terminologies that are rarely seen in training data, making text generated from those deep learning models usually become awkward with lots of "out of vocabulary" tokens. In our paper, we propose some modifications in the training and decoding phase of the state of the art AMR generation model to have a better text realization. Our model is tested using a human-annotated legal dataset, showing an improvement compared to the baseline model.

Keywords. AMR Generation, Deep Learning, Legal

1. Introduction

Abstract Meaning Representation, or AMR in short, is a semantic annotation scheme that encodes a natural language sentence as a rooted, directed graph. Every vertex and edge of the graph is labeled according to the sense of the words in a sentence [1]. We give an example of AMR annotation in Figure 1, where the nodes (e.g. *"enjoy-01", "right-05"*, ...) represent concepts, and the edges (e.g. *":arg0", ":condition"*, ...) represent relations between those concepts. Recently, AMR gains a lot of attention in the NLP research community, as it is widely used as an intermediate meaning representation for NLP tasks, e.g. machine translation [2], summarization [3].

To obtain success in those tasks, the problem of AMR-to-text generation has to be solved effectively. Several deep learning approaches have been proposed to tackle this problem by leveraging a large amount of silver data [4], [5]. Despite acceptable performance on general domain text, those generating models struggle in dealing with the legal domain, where the sentences are complicated structure and contain domain-specific terms. We figure out that lots of out-of-vocabulary words are generated, and almost the negation and conditional sentences are generated incorrectly.

In our paper, we propose a modification in the training phase and decoding phase of the baseline graph to sequence model to improve the generation quality. Specifically, in the training process, we constrain the encoder-decoder model by a controllable variable to avoid the repetitive token generating as well as guiding the model to recognize the

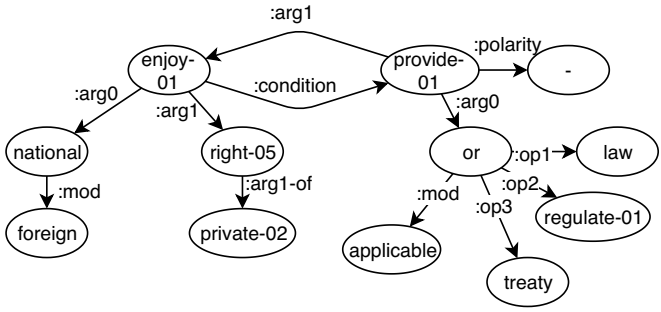

Figure 1. AMR graph for the sentence *"Unless otherwise provided by applicable laws, regulations or treaties, foreign nationals shall enjoy private rights"*.

negation and conditional sentences more appropriately. After training, the model is fine-tune with a silver dataset generated from a civil code in the English version. Moreover, we adopt weighted decoding [6] with a modified beam-search algorithm to avoid out-of-vocabulary words. The model is tested using a human-annotated legal dataset, showing improvement over the baseline model.

2. Background

2.1. Deep learning approaches in AMR-to-text Generation

Given an AMR graph $G = (V;E)$, where V and E denote the sets of nodes and edges, respectively, the goal is to generate a sentence $W = (w_1, w_2, ..., w_n)$ where w_i are words in the vocabulary. Since first introduced as a shared task at SemEval-2017 [7], several approaches have been proposed to tackle this generation problem, with a dominance of deep learning models. Konstas et al. [5] linearized AMR graphs, then adopt an encoder-decoder model to translate these string-like objects into natural language (NeuralAMR). Song et al. [4] modified the encoder side architecture to capture the graph structure data more properly. This resulted in a graph-to-sequence model (Graph2Seq) capable of generating well-written text, obtaining the state of the art BLEU score in this generation problem in 2018. However, these models still struggle when dealing with legal text, i.e. Graph2Seq obtains 9.86 BLEU score on JCivilCode [8], comparing to the score of 32.0 on LDC2017 test set. In our paper, we rely on Graph2Seq to build our baseline model.

2.2. The baseline model

As mentioned before, we adopt the graph-to-sequence model in [4] as our baseline. With a given AMR graph $G = (V;E)$, each node v_i is represented by a hidden state vector h_i, initializing by the word embedding of that node. The graph state g is defined as the set of h_i. Information exchange between a current node v_i and all incoming nodes and outgoing nodes connected to it are captured through a sequence of state transitions $g_0, g_1, ..., g_k$. The encoder side used a long short term memory (LSTM) network to perform this graph state transition. With this state transition mechanism, information of each node is propagated to all its neighboring nodes after each step. After k transition steps, each node

state contains the information of a large context, including its ancestors, descendants, and siblings, where k is the maximum graph diameter in the dataset (we choose $k = 9$ in our experiments). The decoder side is also a LSTM network incorporated with a copy mechanism [9] to deal with decoding objects like name entities, numbers, and date. The detail computation in each step can be found in the original paper.

3. Legal AMR generation

3.1. Conditional training

Conditional Training (CT) [10] is a method to learn an encoder-decoder model $P(y|g,z)$, where z is a discrete control variable and g is the AMR graph. We design z by annotating every (g,y) pair in the training set with the attribute we wish to control, e.g. the length of the linearized graph, or whether g contains negation or not. This attribute value will be determined during training, depend on each training sample. We use an embedding value with size 10 to represent the control variable z. This value will be concatenated to the decoder's input at each step. The objective function of training is given by the cross-entropy loss: $loss_{CT} = -\frac{1}{T}\sum_{t=1}^{T} logP(y|g,z,y_1,...,y_{t-1})$. Parameters of the model are initialized when training with the benchmark general domain dataset, then finetuning with the silver legal dataset to optimize $loss_{CT}$.

3.2. Decoding in legal style

To enhance the probability of generating words with certain features, we adopt Weighted Decoding (WD) that was introduced by Ghazvininejad et al. [11]. On the t_{th} step of decoding, the generated hypothesis $y_{<t} = y_1, ..., y_{t-1}$ is expanded by computing the score for each possible next word w in the vocabulary by the formula:

$$score(w, y_{<t}; g) = score(y_{<t}; g) + logP_{LSTM}(w|y_{<t}, g) + \sum_i w_i * f_i(w; y_{<t}, g).$$

In which $\log P_{LSTM}(w|y_{<t}, g)$ is the log probability of the word w calculated by the bi-LSTM network, score$(y_{<t}; g)$ is the accumulated score of the generated words in the hypothesis $y_{<t}$ and $f_i(w; y_{<t}, g)$ are decoding features with the corresponding weights w_i. There can be multiple features f_i to control multiple attributes, and the weights w_i are hyperparameters. A decoding feature $f_i(w; y_{<t}, g)$ assigns a real value to the word w. The feature can be continuous (e.g. the unigram probability of w) or discrete (e.g. the length of w in characters). A positive weight w_i increases the probability of words w that scores highly with respect to f_i and vice versa.

Another problem of generating text from legal AMR is the out of vocabulary tokens, where lots of words in the legal domain are not included in well-known word embedding, e.g. Word2Vec or Glove. We collect the vocabulary of three datasets: a benchmark dataset in general domains and two datasets obtained from Vietnamese and Japanese civil code. We observe that more than 30% of the words in these vocabulary do not appear in Glove [12]. To deal with this problem, we modified the beam search decoding algorithm. Specifically, after collecting an extra-vocabulary from the legal finetune set, we assign a binary feature to each word w in the test set representing whether w is in the legal vocab or not. This increases the probability of words in the legal vocabulary to be selected to the *top-k* generation, where k is the beam size.

4. Experiments and Results

4.1. Dataset Preparation

In our experiments, we use three datasets: (i) the benchmark dataset LDC2017T10 for training the baseline model, (ii) silver data generated from a Vietnamese Civil Code for fine-tuning the model, and (iii) the JCivilCode dataset [1] [8] for testing the performance. Because of lacking hardware resources, we do not conduct our experiments on silver data sampled from external corpora (like NeuralAMR and Graph2Seq using Gigaword).

Table 1. Statistics of the three dataset used in our experiments

Dataset	LDC2017T10	VN Civil Code	JCivilCode
Number of samples	36.521	3,073	128
Vocabulary size	29,943	3,026	778
Number of words out of vocab	4,453	602	270
:condition edge	1,794	190	69
Negation	10,947	356	57

In dataset (i) we use the linearization and anonymization algorithm provided by Song et al. [4] and Konstas et al. [5]. For dataset (ii), the silver data is obtained by performing two best parsers for legal text: JAMR [13] and CAMR [14] as suggested by Vu et al. [8]. Each sample sentence in the corpus will provide two AMR graphs, this also enlarges the dataset for finetuning our models. The statistics of these datasets can be found in Table 1.

4.2. Results and Analysis

We evaluate our models mainly by BLEU score [15] and METEOR score [16]. We also report the number of OOV words generated from each model. From Table 2, it can be observed that our both proposed modification improve the performance of text generation. While CT increases the BLEU score and METEOR score comparing to the baseline model, Legal Decoding (LD) helps reduce the OOV rate significantly. However, combining both two techniques does not result in the best score overall, where BLEU and METEOR score decrease slightly after LD, since this algorithm sometimes eliminates non-legal words from the *top-k* space.

Our experimental results also confirm the important role of training data. After finetuning with a legal dataset, we obtain 2.81 and 0.96 improvement on BLEU and METEOR score, respectively. When comparing to the state of the art pre-trained models, with a huge amount of data, our proposed modification still got lower results by a small margin.

To have a closer look, we provide some output examples for each model in Table 3. All the models still generate low-quality sentences, with grammatical errors and repetitive words. The baseline model trained without any legal data provides an out-domain word that does not appear in the source AMR graph. After finetuning, the sentences generated become longer but not so meaningful except for the output of CT model, which includes almost correct information. LD, as mentioned earlier, could help reduce the OOV rate overall, but may cause some words or fragments missing and repetitive.

[1] https://github.com/sinhvtr/legal_amr

Table 2. Generation results in BLEU score, METEOR score and number of OOV generated. The baseline Graph2Seq is trained on benchmark dataset only. The next four lines show our proposed modifications, with and without finetuning data. The last two lines are the results of two best pretrained models with extra corpus.

Model	BLEU	METEOR	OOV
Baseline Graph2Seq	5.50	16.78	135
Graph2Seq + CT	6.82	17.42	112
Graph2Seq + Finetune data	8.31	17.74	145
Graph2Seq + Finetune data + Conditional Training	**8.56**	**18.61**	143
Graph2Seq + Finetune data + LD	8.42	17.98	57
Graph2Seq + Finetune data + CT + LD	8.43	18.04	**57**
Graph2Seq Pretrained on 2M Gigaword corpus	9.31	21.38	29
NeuralAMR Pretrained on 2M Gigaword corpus	9.07	20.55	35

Table 3. Output comparison with an example from JCivilCode dataset

Gold data Unless otherwise provided by applicable laws, regulations or treaties, foreign nationals shall enjoy private rights.
Baseline model the foreign national enjoy a private right not if the applicable law or economic treaty
Baseline model + finetune data when it is not provided for by law or the treaties to enjoy the private rights , the foreign national shall have the enjoy private rights .
Baseline model + finetune data + CT the foreign national will enjoy private rights without providing applicable regulate regulate or treaty
Baseline model + finetune data + CT + LD when a foreign national enjoys the private right , if not provided for by law or the provisions of law or the provisions of law .
Graph2Seq Pretrained on 2M Gigaword foreign nationals will enjoy private rights while there are no laws or regulations if the or or without the regulations are provided .

5. Conclusion

In this paper, we figure out the difficulties of AMR generation in the legal domain, where the logical structure is complicated and lots of domain-specific terms are not in the well-known vocabulary. We propose two modifications to the training and decoding phases of the state of the art graph to sequence model to tackle these difficulties. The experimental results prove the effectiveness of our method over the baseline model. Despite the improvement, all models in our experiments still generate low-quality text from legal AMR. The best-reported score is only 9.31 for BLEU and 21.38 for METEOR, leaving a challenge for research in this domain.

Acknowledgments. This work was supported by JST CREST Grant Number JPMJCR1513, Japan.

References

[1] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer and N. Schneider, Abstract Meaning Representation for Sembanking, in: *Proceedings of*

the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 178–186. https://www.aclweb.org/anthology/W13-2322.

[2] B. Jones, J. Andreas, D. Bauer, K.M. Hermann and K. Knight, Semantics-Based Machine Translation with Hyperedge Replacement Grammars, in: *Proceedings of COLING 2012*, The COLING 2012 Organizing Committee, Mumbai, India, 2012, pp. 1359–1376.

[3] F. Liu, J. Flanigan, S. Thomson, N. Sadeh and N.A. Smith, Toward Abstractive Summarization Using Semantic Representations, in: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 1077–1086. doi:10.3115/v1/N15-1114.

[4] L. Song, Y. Zhang, Z. Wang and D. Gildea, 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)*, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1616–1626. doi:10.18653/v1/P18-1150. https://www.aclweb.org/anthology/P18-1150.

[5] I. Konstas, S. Iyer, M. Yatskar, Y. Choi and L. Zettlemoyer, 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)*, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 146–157. doi:10.18653/v1/P17-1014.

[6] A. See, S. Roller, D. Kiela and J. Weston, What makes a good conversation? How controllable attributes affect human judgments, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1702–1723. doi:10.18653/v1/N19-1170. https://www.aclweb.org/anthology/N19-1170.

[7] J. May and J. Priyadarshi, Semeval-2017 task 9: Abstract meaning representation parsing and generation, in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 536–545.

[8] S.V. Trong and M.N. Le, An empirical evaluation of AMR parsing for legal documents, *ArXiv* **abs/1811.08078** (2018).

[9] J. Gu, Z. Lu, H. Li and V.O.K. Li, Incorporating Copying Mechanism in Sequence-to-Sequence Learning, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016, pp. 1631–1640. doi:10.18653/v1/P16-1154. http://aclweb.org/anthology/P16-1154.

[10] A. Fan, D. Grangier and M. Auli, Controllable Abstractive Summarization, in: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 45–54. doi:10.18653/v1/W18-2706. https://www.aclweb.org/anthology/W18-2706.

[11] M. Ghazvininejad, X. Shi, J. Priyadarshi and K. Knight, Hafez: an Interactive Poetry Generation System, in: *Proceedings of ACL 2017, System Demonstrations*, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 43–48. https://www.aclweb.org/anthology/P17-4008.

[12] J. Pennington, R. Socher and C.D. Manning, Glove: Global vectors for word representation, in: *In EMNLP*, 2014.

[13] J. Flanigan, S. Thomson, J. Carbonell, C. Dyer and N.A. Smith, 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)*, Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 1426–1436. doi:10.3115/v1/P14-1134.

[14] C. Wang, S. Pradhan, X. Pan, H. Ji and N. Xue, CAMR at SemEval-2016 Task 8: An Extended Transition-based AMR Parser, in: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Association for Computational Linguistics, San Diego, California, 2016, pp. 1173–1178. doi:10.18653/v1/S16-1181.

[15] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, Bleu: a Method for Automatic Evaluation of Machine Translation, in: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. doi:10.3115/1073083.1073135. https://www.aclweb.org/anthology/P02-1040.

[16] M. Denkowski and A. Lavie, Meteor Universal: Language Specific Translation Evaluation for Any Target Language, in: *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.