Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation

Hardy
The University of Sheffield
hhardy2@sheffield.ac.uk

Andreas Vlachos
The University of Sheffield
a.vlachos@sheffield.ac.uk

Abstract

Recent work on abstractive summarization has made progress with neural encoder-decoder architectures. However, such models are often challenged due to their lack of explicit semantic modeling of the source document and its summary. In this paper, we extend previous work on abstractive summarization using Abstract Meaning Representation (AMR) with a neural language generation stage which we guide using the source document. We demonstrate that this guidance improves summarization results by 7.4 and 10.5 points in ROUGE-2 using gold standard AMR parses and parses obtained from an off-the-shelf parser respectively. We also find that the summarization performance using the latter is 2 ROUGE-2 points higher than that of a well-established neural encoder-decoder approach trained on a larger dataset. Code is available at https://github.com/sheffieldnlp/AMR2Text-summ

1 Introduction

Abstractive summarization is the task of automatically producing the summary of a source document through the process of paraphrasing, aggregating and/or compressing information. Recent work in abstractive summarization has made progress with neural encoder-decoder architectures (See et al., 2017; Chopra et al., 2016; Rush et al., 2015). However, these models are often challenged when they are required to combine semantic information in order to generate a longer summary (Wiseman et al., 2017). To address this shortcoming, several works have explored the use of Abstract Meaning Representation (Banarescu et al., 2013, AMR). These were motivated by AMR's capability to capture the predicate-argument structure which can be utilized in information aggregation during summarization.

However, the use of AMR also has its own shortcomings. While AMR is suitable for information aggregation, it ignores aspects of language such as tense, grammatical number, etc., which are important for the natural language generation (NLG) stage that normally occurs in the end of the summarization process. Due to the lack of such information, approaches for NLG from AMR typically infer it from regularities in the training data (Pourdamghani et al., 2016; Konstas et al., 2017; Song et al., 2016; Flanigan et al., 2016), which however is not suitable in the context of summarization. Consequently, the main previous work on AMR-based abstractive summarization (Liu et al., 2015) only generated bag-of-words from the summary AMR graph.

In this paper, we propose an approach to guide the NLG stage in AMR-based abstractive summarization using information from the source document. Our objective is twofold: (1) to retrieve the information missing from AMR but needed for NLG and (2) improve the quality of the summary. We achieve this in a two-stages process: (1) estimating the probability distribution of the side information, and (2) using it to guide a Luong et al. (2015)'s seq2seq model for NLG.

Our approach is evaluated using the Proxy Report section from the AMR dataset (Knight et al., 2017, LDC2017T10) which contains manually annotated document and summary AMR graphs. Using our proposed guided AMR-to-text NLG, we improve summarization results using both gold standard AMR parses and parses obtained using the RIGA (Barzdins and Gosko, 2016) parser by 7.4 and 10.5 ROUGE-2 points respectively. Our model also outperforms a strong baseline seq2seq model (See et al., 2017) for summarization by 2 ROUGE-2 points.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 768–773
Brussels, Belgium, October 31 - November 4, 2018.

2 Related Work

Abstractive Summarization using AMR: In Liu et al. (2015) work, the source document's sentences were parsed into AMR graphs, which were then combined through merging, collapsing and graph expansion into a single AMR graph representing the source document. Following this, a summary AMR graph was extracted, from which a bag of concept words was obtained without attempting to form fluent text. Vilca and Cabezudo (2017) performed a summary AMR graph extraction augmented with discourse-level information and the PageRank (Page et al., 1998) algorithm. For text generation, Vilca and Cabezudo (2017) used a rule-based syntactic realizer (Gatt and Reiter, 2009) which requires substantial human input to perform adequately.

Seq2seq using Side Information: In Neural Machine Translation (NMT) field, recent work (Zhang et al., 2018) explored modifications to the decoder of seq2seq models to improve translation results. They used a search engine to retrieve sentences and their translation (referred to as translation pieces) that have high similarity with the source sentence. When similar n-grams from a source document were found in the translation pieces, they rewarded the presence of those n-grams during the decoding process through a scoring mechanism calculating the similarity between source sentence and the source side of the translation pieces. Zhang et al. (2018) reported improvements in translation results up to 6 BLEU points over their seq2seq NMT baseline. In this paper we use the same principle and reward n-grams that are found in the source document during the AMR-to-Text generation process. However we use a simpler approach using a probabilistic language model in the scoring mechanism.

3 Guiding NLG for AMR-based summarization

We first briefly describe the AMR-based summarization method of Liu et al. (2015) and then our guided NLG approach.

3.1 AMR-based summarization

In Liu et al. (2015)'s work, each of the sentence of the source document was parsed into an AMR graph, and combined into a source graph, $G = (V, E)$ where $v \in V$ and $e \in E$ are the unique concepts and the relations between pairs of concepts. They then extracted a summary graph, G' using the following sub-graph prediction:

$$G' = \arg\max_{\hat{G}=(\hat{V},\hat{E})} \sum_{v\in\hat{V}} \boldsymbol{\theta}^{\intercal}\mathbf{f}(v) + \sum_{e\in\hat{E}} \boldsymbol{\psi}^{\intercal}\mathbf{f}(e) \quad (1)$$

where $\mathbf{f}(v)$ and $\mathbf{f}(e)$ are the feature representations of node v and edge e respectively. The final summary produced was a bag of concept words extracted from G'. This output we will be replacing with our proposed guided NLG.

3.2 Unguided NLG from AMR

Our baseline is a standard (unguided) seq2seq model with attention (Luong et al., 2015) which consists of an encoder and a decoder. The encoder computes the hidden representation of the input, $\{z_1, z_2, \ldots, z_k\}$, which is the linearized summary AMR graph, G' from Liu et al. (2015), following Van Noord and Bos (2017)'s preprocessing steps. Following this, the decoder generates the target words, $\{y_1, y_2, \ldots, y_m\}$, using the conditional probability $P_{s2s}(y_j|y_{<j}, z)$, which is calculated using the equation

$$P_{s2s}(y_j|y_{<j}, z) = \text{softmax}(\mathbf{W}_s\tilde{\mathbf{h}}_t) \quad (2)$$

, where the attentional hidden state, $\tilde{\mathbf{h}}_t$ is calculated using the equation

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_c[\mathbf{c}_t; \mathbf{h}_t]) \quad (3)$$

, where $\mathbf{c}_t$ is the source context vector, and $\mathbf{h}_t$ is the target RNN hidden state. The source context vector is defined as the weighted average over all the source RNN hidden states, $\bar{\mathbf{h}}_s$, given the alignment vector, $\mathbf{a}_t$ where $\mathbf{a}_t$ is defined as

$$\mathbf{a}_t(s) = \frac{\exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_s))}{\sum_{s'} \exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_{s'}))} \quad (4)$$

3.3 Guided NLG from AMR

Our goal is to improve the text generated from the summary AMR graph by the probability distribution of the seq2seq model, P_{s2s} using the source document. Since not all sentences in the source document will be used in generating the summary, we prune the source document to a set of k sentences which have the highest similarity with the summary AMR graph. For graph-to-graph similarity comparison, we use the source document AMR parses and calculate the Longest Common

Subsequence (LCS) between the linearized AMR parses and the summary AMR graph. We keep the top-k sentences sorted by LCS length. To distinguish this pruned document from the source document, we refer to the former as side information.

Our aim is is to combine P_{s2s} with the probability distribution estimated using words in the side information, P_{side}, in order to score each word given its context during decoding. We estimate P_{side} as the linear interpolation of 2-gram to 4-gram probabilities in the form of

$$\begin{aligned} P_{side}(x_j|x_{j-3}^{j-1}) = \lambda_3 P_{LM}(x_j|x_{j-3}^{j-1}) \\ + \lambda_2 P_{LM}(x_j|x_{j-2}^{j-1}) \\ + \lambda_1 P_{LM}(x_j|x_{j-1}) \end{aligned} \quad (5)$$

, where x_j is a word occurring in side information document, P_{LM} is an N-gram LM estimated using Maximum Likelihood:

$$P_{LM}(x_j|x_{j-N-1}^{j-1}) = \frac{\text{count}(x_{j-N-1} \dots x_j)}{\text{count}(x_{j-N-1} \dots x_{j-1})} \quad (6)$$

and λ_i is defined as

$$\lambda_i = \theta\lambda_{i-1} \text{ where } \theta \in \mathbb{R}, \lambda_i > 0 \text{ and } \sum_i \lambda_i = 1 \quad (7)$$

where θ is a hyper-parameter that we tune using the dev dataset during the experiments.

Lastly, we combine the probability distribution of the decoder, P_{s2s} with that provided by the side information, P_{side}, as follows:

$$s(y_j|y_{<j}, z) = log\, a + \psi * log(\frac{b}{a} + 1) \quad (8)$$

where ψ is a hyper-parameter determining the influence of the side information on the decoding process, a is $P_{s2s}(y_j|y_{<j}, z)$ and b is $P_{side}(y_j|y_{j-3}^{j-1})$. $s(y_j|y_{<j}, z)$ is used during beam search replacing $P_{s2s}(y_j|y_{<j}, z)$ for all words that occur in the side information. The intuition behind Eq. 8 is that we are rewarding word y_j when it appears in similar context in the side information, i.e. the source document being summarized.

4 Experiments

We conduct experiments in order to answer the following questions about our proposed approach: (1) Is our baseline model comparable with the state-of-the-art AMR-to-text approaches? (2) Does the guidance from the source document improve the result of AMR-to-Text in the context of summarization? (3) Does the improvement in AMR-to-Text hold when we use the generator for abstractive summarization using AMR? We answer each of these in the following paragraphs.

Model	BLEU
Our model (unguided NLG)	21.1
NeuralAMR (Konstas et al., 2017)	22.0
TSP (Song et al., 2016)	22.4
TreeToStr (Flanigan et al., 2016)	23.0

Table 1: Results for AMR-to-text

AMR-to-Text baseline comparison We compare our baseline model (described in §3.2) against previous works in AMR-to-text using the data from the recent SemEval-2016 Task 8 (May, 2016, LDC2015E86). Table 1 reports BLEU scores comparing our model against previous works. Here, we see that our model achieves a BLEU score comparable with the state-of-the-art, and thus we argue that it is sufficient to be used in our subsequent experiments with guidance.

Guided NLG for AMR-to-Text In this experiment we apply our guided NLG mechanism described in §3.3 to our baseline seq2seq model. To isolate the effects of guidance we skip the actual summarization process and proceed to directly generating the summary text from the gold standard summary AMR graphs from the Proxy Report section. To determine the hyper-parameters, we perform a grid search using the dev dataset, where we found the best combination of ψ, θ and k are 0.95, 2.5 and 15 respectively. We have two different settings for this experiment: the oracle and non-oracle settings. In the oracle setting, we directly use the gold standard summary text as the guidance for our model. The intuition is that in this setting, our model knows precisely which words should appear in the summary text, thus providing an upper bound for the performance of our guided NLG approach. In the non-oracle setting, we use the mechanism described in §3.3. We also compare them against the baseline (unguided) model from §3.2. Table 2 reports performance for all models. The difference between the guided and the unguided model is 16.2 points in BLEU and 9.9 points in ROUGE-2, while there is room for improvement as evidenced by the difference between the oracle and non-oracle result.

Guided NLG for full summarization In this experiment we combine our guided NLG model

Model	BLEU	F_1 ROUGE		
		R-1	R-2	R-L
Guided NLG (Oracle)	61.3	79.4	63.7	76.4
Guided NLG	45.8	70.7	49.5	64.9
Unguided NLG	29.6	68.6	39.6	61.3

Table 2: BLEU and ROUGE results for guided and unguided models using test dataset.

with Liu et al. (2015)'s work in order to generate fluent texts from their summary AMR graphs using the hyper-parameters tuned in the previous paragraph. Liu et al. (2015) used parses from both the manual annotation of the Proxy dataset as well as those obtained using the JAMR parser (Flanigan et al., 2014). Instead of JAMR we use the RIGA parser (Barzdins and Gosko, 2016) which had the highest accuracy in the SemEval 2016 Task 8 (May, 2016). We compare our result against Liu et al. (2015)'s bag of words[1], the unguided AMR-to-text model from §3.2, and a seq2seq summarization model (OpenNMT BRNN)[2,3] which summarizes directly from the source document to summary sentence without using AMR as an interlingua and is trained on CNN/DM corpus (Hermann et al., 2015) using the same settings as See et al. (2017).

AMR parses	NLG Model	F_1 ROUGE		
		R-1	R-2	R-L
Gold	Guided	40.4	20.3	31.4
	Unguided	38.9	12.9	27.0
	Liu et al. (2015)	39.6	6.2	22.1
RIGA	Guided	42.3	21.2	33.6
	Unguided	37.8	10.7	26.9
	Liu et al. (2015)	40.9	5.5	21.4
Directly from Text	OpenNMT BRNN 2 layer, emb 256, hidden 1024	36.1	19.2	31.1

Table 3: The F_1 ROUGE scores for guided, unguided, Liu et al. (2015) (BoW) results in Gold and RIGA parses, and seq2seq summarization. All models are run using test dataset.

In Table 3, we can see that our approach results in improvements over both the unguided AMR-to-text and the standard seq2seq summarization. One interesting note is that using the RIGA parses result in higher ROUGE scores than the gold parses for the guided model in our experiment. This phenomenon was also observed in Liu et al. (2015)'s experiment where the summary graphs extracted from automatic parses had higher accuracy than those extracted from manual parses. We hypothesize this can be attributed to how the AMR dataset is annotated as there might be discrepancies in different annotator's choices of AMR concepts and relations for sentences with similar wording. In contrast, the AMR parsers introduce errors, but they are consistent in their choices of AMR concepts and relations. The discrepancies in the manual annotation could have impacted the performance of the AMR summarizer that we use more negatively than the noise introduced due to the AMR parsing errors.

NLG Model	Generated Summary
Gold	on 8 august 2008 russia conducted **airstrikes** on **georgian** targets .
Guided	on 8 august 2008 russia conducted **airstrikes** on **georgian** separatist targets .
Unguided	on 8 august 2008 russia conducted **a softening** of the **georgia 's** separatist target .
Seq2seq	the russian laboratory complex is a 90 - building campus and served as the location for russia 's secret biological weapons program in the soviet era of a moscow regional depository threaten moscow .

Table 4: Result summaries of guided, unguided and seq2seq models compared with gold summary.

In Table 4, we show sample summaries from the different models, where we can see that our guided model improves the unguided model by correcting a wrong word (*a softening*) into a correct one (*airstrikes*) and introducing a better suited word from the source document (*georgian* instead of *georgia 's*).

NLG Model	Fluency
Guided	2.66
Unguided	2.16

Table 5: Fluency scores on test dataset.

We also evaluated manually by asking human evaluators to judge sentences' fluency (grammatical and naturalness) on a scale of 1 (worst) to 6 (best) for the guided and unguided model (see Ta-

[1]We were able to obtain comparable AMR summarization subgraph prediction to their reported results using their published software but not to match their bag-of-word generation results.

[2]We use the OpenNMT-pytorch implementation https://github.com/OpenNMT/OpenNMT-py and a pre-trained model downloaded from http://opennmt.net/OpenNMT-py/Summarization.html which has higher result than See et al. (2017)'s summarizer.

[3]The pre-trained model generates multiple sentences summary, but we use only the first sentence summary for evaluation in accordance with the AMR dataset.

ble 5). While the manual evaluation shows improvement over the unguided model, on the other hand, grammatical mistakes and redundant repetition in the generated text are still major problems (see Table 6) in our AMR generation.

Guided NLG Model	Problems
the soldiers were injured when **a attempt** to defuse the bombs .	grammatical mistake
on 20 october 2002 the state - run radio nepal reported **on 20 october 2002** that at the evening **- run radio nepal reported on 20 october 2002** that the guerrillas were killed **and killed** .	redundant repetition

Table 6: Problems in guided model's summaries.

5 Conclusion and Future Works

In this paper we proposed a guided NLG approach that substantially improves the output of AMR-based summarization. Our approach uses a simple guiding process based on a probabilistic language model. In future work we aim to improve summarization performance by jointly training the guiding process with the AMR-based summarization process.

Acknowledgments

We are thankful for Gerasimos Lampouras for his help with the manual evaluation process and all volunteers who participated in it. We would also like to thank the Indonesian government that has sponsored the first author's studies through the Indonesia Endowment Fund for Education (LPDP). The second author is supported by the EU H2020 SUMMA project (grant agreement number 688139) and the EPSRC grant eNeMILP (EP/R021643/1).

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.

Guntis Barzdins and Didzis Gosko. 2016. RIGA at SemEval-2016 Task 8: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 15th Annual Conference of the NAACL HLT*, pages 93–98.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using Tree Transducers. In *Proceedings of the 2016 Conference of the NAACL*, pages 731–739.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. *Proceedings of the 52th Annual Meeting of the ACL*, pages 1426–1436.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG : A realisation engine for practical applications. *Proceedings of the 12th European Workshop on NLG*, (March):90–93.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Neural Information Processing Systems*, pages 1–14.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2017. Abstract Meaning Representation (AMR) Annotation Release 2.0 LDC2017T10.

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 2015 Conference of the NAACL HLT*, pages 1077 – 1086.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward Abstractive Summarization Using Semantic Representations. In *Proceedings of the 2015 Conference of the NAACL HLT*, pages 1077–1086.

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. The Association for Computational Linguistics.

Jonathan May. 2016. SemEval-2016 Task 8: Meaning Representation Parsing. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1063–1073.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems*, 54(1999-66):1–17.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from Abstract Meaning Representations. In *Proceedings of the 9th International Natural Language Generation*, volume 0, pages 21–25.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. *In Proceedings of the Conference on EMNLP*, (September):379–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the ACL*.

Linfeng Song, Yue Zhang, Xiaochang Peng, Zhiguo Wang, and Daniel Gildea. 2016. AMR-to-text generation as a Traveling Salesman Problem. In *Proceedings of the 2016 Conference on EMNLP*, pages 2084–2089.

Rik Van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7(2016):93–108.

Gregory Cesar Valderrama Vilca and Marco Antonio Sobrevilla Cabezudo. 2017. A Study of Abstractive Summarization Using Semantic Representations and Discourse Level Information. In *International Conference on Text, Speech, and Dialogue*, pages 482–490. Springer.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on EMNLP*, pages 2253–2263, Copenhagen, Denmark.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding Neural Machine Translation with Retrieved Translation Pieces. In *Proceedings of the 16th Annual Conference of the NAACL HLT*, New Orleans, Louisiana. The Association for Computational Linguistics.