HPSG/MRS-Based Sentence Generation with Transformer

Gyu-min Lee
Korea University
gyuminlee@korea.ac.kr

Sanghoun Song
Korea University
sanghoun@korea.ac.kr

1 Introduction

In recent years, Natural Language Processing (NLP) has achieved significant improvements thanks to the introduction of neural networks. For instance, BERT (Devlin et al., 2018) achieved state-of-the-art performance in many tasks. However, BERT is not linguistically motivated but developed with a concept of Cloze Task (Taylor, 1953).

Some people claim that natural language texts can be processed and generated successfully only with the text information by the success of NLP systems without linguistic motivations. For instance, a deep learning textbook introduces NLP as a field that does not require feature engineering (Subramanian, 2018, p.75). In doing so, it has often been postulated that the raw texts are enough to build up the NLP systems with.

Meanwhile, Konstas et al. (2017) redefined natural language generation (NLG) as a machine translation (MT) task by training a model to parse to and generate from Abstract Meaning Representation (AMR). In other words, they built an MT model that translates AMR to natural language text using a sequence-to-sequence model with bidirectional LSTMs and Attention mechanisms.

While AMR formalism abstracts away the surface form to represent human language sentences as a directed graph, other formalisms contain richer information. Minimal Recursion Semantics (MRS) formalism (Copestake et al., 2005), for instance, contains rich information about the surface form, including number, tense, and case information (Konstas et al., 2017). Using this richer representation, Hajdik et al. (2019) reproduced the work of Konstas et al. (2017) and showed significant improvement in translation when measured by the BLEU metric (Papineni et al., 2002).

Konstas et al. (2017) and Hajdik et al. (2019) have their own significance in that they demonstrate using the linguistically motivated features works for a neural network-based NLP task. The fact the linguistically richer representation significantly improved the performance also hints that linguistic features are still relevant when applied in a proper format. In other words, a proper use of grammars like Head-Driven Phrase Structure Grammar (HPSG), as in Hajdik et al. (2019), can improve NLP even in this age of neural networks.

However, both studies used a bi-LSTM based sequence-to-sequence model for the MT task. RNNs "learn" from sequential data, such as natural language text divided by token, through the back-propagation of the error to the connected cells. In this process, the gradient information is used to minimize the loss of the model. As this gradient is acquired by differentiation, the gradient of the old cells vanishes, which impedes the RNNs from processing longer sentences.

Later algorithms such as LSTM and GRU tried to bypass this problem by strategically "forgetting" some information. Meanwhile, Attention Mechanism helps a sequence-to-sequence model to process longer sentences by indicating which cells to "pay attention to". Built solely upon this mechanism, Transformer performed better at MT tasks. Since MRS representation, linearized as in Hajdik et al. (2019), tends to be long, it is expected that applying Transformer would further improve the performance.

In short, the current research aims to reproduce the work of Hajdik et al. (2019) using Transformer to substantiate that computational grammar inspired by rich syntactic and semantic formalism does improve neural NLG. Along the line of the previous studies, the present study draws more attention to the use of grammar-based representation. In so doing, the present study aims to demonstrate that Transformer-based NLG fits into the MRS representation.

2 Method

2.1 Data

The suggested model uses the data from Hajdik et al. (2019), which consists of gold and silver datasets and are created with the HPSG motivation. The gold dataset is the Redwoods Treebank (Oepen et al., 2004) release 1214. The Redwoods Treebank is a parallel corpus of natural language sentences and their MRS representations. The latter was predicted using English Resource Grammar (ERG; Flickinger, 2000), then manually checked by human reviewers.

To accompany the gold dataset, one million sentences from the Gigaword Corpus were prepared. Hajdik et al. (2019) used ERG with ACE processor.[1] ERG is a computational grammar based on Pollard and Sag (1994) but implements MRS without implementing binding theory

[1] moin.delph-in.net/AceTop

(Flickinger, 2000). It guarantees the well-formedness of the MRS representation. However, it is still capable of producing incorrect MRS representation. Therefore, the parser failed to parse 10.3% of the sentences, in which case the sentence was discarded. In total, 87,679 sentences were prepared as the gold and silver datasets.

In Hajdik et al. (2019), the MRS representations of the sentences assumed with HPSG-based grammar of ERG (#1 in Figure 1) were converted into Dependency MRS (DMRS), which is interchangeable with MRS. The DMRS representation is converted into PENMAN format following Goodman (2018) (#2 in Figure 1), then linearized as Konstas et al. (2017) (#3 in Figure 1) so that it can be fed to a sequence-to-sequence model. They then anonymized what is considered a named entity according to the MRS representation by replacing the token on the raw text to reduce data sparsity. The NLTK implementation of Moses tokenization (Bird et al., 2009) was used. This entire data preparation process was done locally using the code provided by Hajdik et al. (2019).

2.2 Model

As aforementioned Konstas et al. (2017) and Hajdik et al. (2019) employed the bidirectional LSTM (Hochreiter and Schmidhuber, 1997) for their sequence-to-sequence model. By contrast, the current work applies Transformer. While the sequential nature of RNNs inherently brings computational disadvantages, Transformer utilizes Attention mechanisms, which configure dependencies regardless of distance for more parallelization capability and performance (Vaswani et al., 2017).

Transformer also performs relatively better than traditional RNNs with longer texts, offering a breakthrough to machine translation systems. Thus, Transformer is expected to solve the problem of long-range issue and translation performance encountered in linearized MRS representation.

2.3 Implementation and Evaluation

The present study utilizes OpenNMT-py (Klein et al., 2017) in order to implement Transformer.[2] Validation was carried out every 5,000 steps. The model was saved upon each validation. The training was carried out using Google Colab, which provides access to a robust GPU environment but limits the execution up to 12 hours. Therefore, training was conducted using the OpenNMT feature to train from a saved model.

The test dataset was then translated with the trained models, detokenized, and deanonymized as Hajdik et al. (2019). Automatic evaluation of BLEU (Papineni et al., 2002) was carried out using SacreBLEU (Post, 2018), following Hajdik et al. (2019). During the translation, beam search with the beam width of 5 was used. The beam size and the BLEU calculation method we used here were determined in accordance with Hajdik et al. (2019) to directly compare the result with their research with minimal difference.

[2]Specifically, release v.2.0.0rc2 of OpenNMT-py was used.

Model	BLEU
Konstas et al. (2017)	33.8
Hajdik et al. (2019)	77.17
Ours	64.2

Table 1: Comparison of the results.

3 Results

3.1 The BLEU Score

BLEU is a metric for the automatic evaluation of machine translation. The metric itself is not designed for NLG. However, as Konstas et al. (2017) and Hajdik et al. (2019) realized NLG systems based on MT approach, they used this automatic metric to evaluate their NLG systems. The current research also utilizes BLEU to evaluate the objective performance of the model.

Konstas et al. (2017) achieved up to 33.8 BLEU. Hajdik et al. (2019) significantly improved the result to 77.17 for the entire dataset, and 83.37 for partial dataset.

While Transformer was expected to further improve the result by resolving the vanishing gradients issue, the result turned out to be the opposite. We ran the Transformer for up to 70,000 steps. Then, the BLEU score was measured for every 5,000 steps. The score peaked at 30,000 steps with 64.2 BLEU. The score decreased afterward with the accuracy, perplexity, and cross entropy plateauing, hinting the overfitting of the model after 30,000 steps.

3.2 Translation Samples

(1) a. **PREDICTION**: If I am correct, they will help you understand exactly what it is saying the Linux community of good software - and perhaps they will help you become more productive yourself.

b. **ANSWER**: If I'm correct, they'll help you understand exactly what it is that makes the Linux community such a fountain of good software—and, perhaps, they will help you become more productive yourself.

The actual translation samples reveal the problem with using Transformer. For the PREDICTION, we present detokenized and deanonymized prediction by the model. This corresponds to the model's translation from the MRS representation. The ANSWER is the original text the model is supposed to translate to. Overall, the Transformer model seems to perform relatively better with longer MRS representations, as in (1).[3]

(2) a. **PREDICTION**: The myth and the sword.

[3]We used the 30,000 step model for the predictions here.

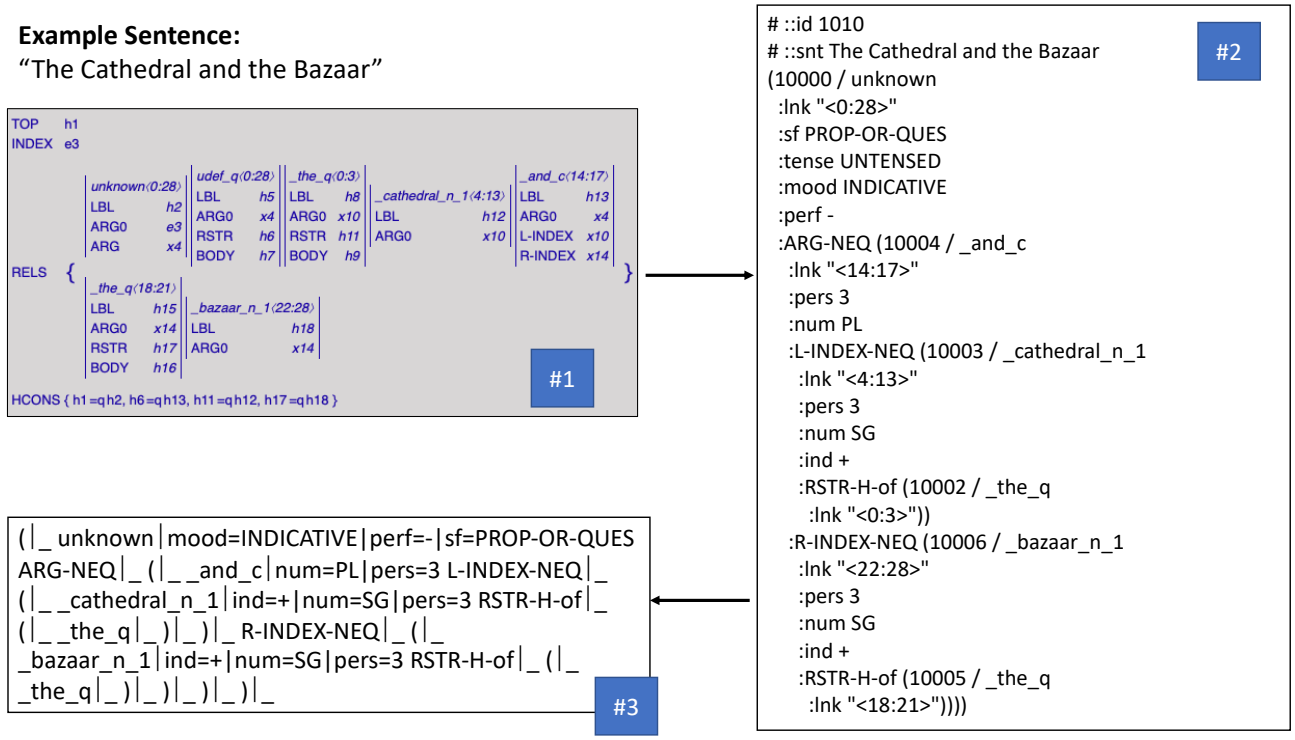

Figure 1: MRS linearization process.

b. **ANSWER**: The Cathedral and the Bazaar

```
(|_ unknown|mood=INDICATIVE|perf=-|sf
   =PROP-OR-QUES ARG-NEQ|_ (|_ _and_c
   |num=PL|pers=3 L-INDEX-NEQ|_ (|_
   _cathedral_n_1|ind=+|num=SG|pers=3
    RSTR-H-of|_ (|_ _the_q|_ )|_ )|_
   R-INDEX-NEQ|_ (|_ _bazaar_n_1|ind
   =+|num=SG|pers=3 RSTR-H-of|_ (|_
   _the_q|_ )|_ )|_ )|_ )|_
```

However, it turns out, the model struggles with the seemingly simple task of lexical choices. For instance, the linearized MRS given above seems straightforward. The lexical items of the answer (2b) are all given in the MRS representation. However, as can be seen with (2a), it appears the model failed to catch the lexical item and instead chose different items.

(3) a. **PREDICTION**: = = = Objectives = = =

b. **ANSWER**: Abstract

```
(|_ unknown|mood=indicative|perf=-|sf
   =prop-or-ques arg-neq|_ (|_
   _abstract_n_1|ind=+|num=sg|pers=3
   )|_ )|_
```

The trend continues even with a single word sentence (3). While the linearized MRS contains only a single word, *abstract*, as given above, the model answered with *Objectives*. This trend persists throughout the prediction: while the syntactic structure appears to be translated relatively well, it appears Transformer model failed to make correct lexical decisions.

3.3 Error Analysis

(4) a. **PREDICTION**: do you want to travel around what time ?

b. **TARGET**: around what time do you want to travel ?

(5) a. **PREDICTION**: What am I doing now ?

b. **TARGET**: What do I do now " ?

In order to understand the reasons for the lower performance of this model, we manually inspected 100 randomly selected translation samples. In detail, we compare the anonymized and undetokenized predictions from the 30,000 step model. Each translation was categorized as: no error, lexical choice error, syntactic error, punctuation error, and missing elements error. Since some of the grammatical information can be abstracted away, some differences were not counted as errors. Those differences are the location of adverbial phrases that do not alter the meaning (see (4)), the use of aspect (present on behalf of present progressive, or vice versa, like (5)), use of clitics (*will* on behalf of *'ll*), and unreasonable punctuations (sentences that end with a quotation mark without opening quotation mark, like (5)).

As Table 2 summarizes, around half of the translation presented no error. This trend coincides with Hajdik et al. (2019), in which manual inspection showed that BLEU metric was underestimating the model due to the issues like formatting. It appears then that, while the model was generally able to generate acceptable sentences from

Error	Number	Sample Prediction
No Error	47	*Okay , we have card0 options .*
Lexical	31	*I assume there is a full* ***salon*** *on the shipping costs .*
Punctuation	8	***: **** *named0*
Lexical & Missing Argument	5	***Don 't Linger***
Lexical & Syntactic	4	*When ad dollars is tight , the high page cost is* ***generally*** *a major* ***UNKcontributor0*** *for UNKadvertisers0 who want to appear regularly in a publication or not at all .*
Missing Argument	3	*Requesting immediately .*
Syntactic	2	***polite0 refund .***
SUM	100	

Table 2: Number of errors from the 100 translation samples. The errors in the sample prediction are marked in bold face.

linearized MRS representation, the details that are not reflected in the representation prevented the model from achieving a high score.

Among the erroneous 53 cases, 40 cases involved lexical choice problems like (2) and (3). This supports our assumption that the model learned to translate syntactic aspect of MRS representation fairly well, but failed at making correct lexical choices from it. We assume this issue stems from Attention mechanism, on which Transformer is built. An MRS representation contains many functional keywords and symbols while containing few lexical tokens inside. Thus, it appears that the Attention mechanism pays attention to functional keywords instead of lexical items, thus failing to make a correct lexical decision.

To further investigate this assumption, we retrieved Attention weights from the translation of (2) using the `attn_debug` option of OpenNMT-Py. The Attention weight was then visualized in a heatmap using the Seaborn package for Python. Here, the original sentence is presented on the horizontal axis, and the prediction tokens on the vertical axis. A brighter color indicates stronger attention. The result (Figure 2) shows that when the tokens of *myth* and *sword* were both expected to pay the strongest attention to the lexical items of *cathedral* and *bazaar*, they were instead paying attention to the functional items that indicate the syntactic positions of the items.

4 Discussion

The current research reproduced Hajdik et al. (2019) with Transformer. They showed that HPSG-based computation grammar, such as ERG, can improve neural NLG. This research takes that insight and applied a newer mechanism, expecting the it would perform better at this specific task by better processing longer sentences.

However, the result shows that Transformer model struggles with this task. It appears that the model faile extract lexical items from linearized MRS appropriately. On the other hand, it could retrieve syntactic structure from it. The full probe of this is beyond the scope of our research. However, we suspect that relying on Attention mechanism alone is harmful for the model to interpret the linearized MRS representation.

MRS is a concise representation of the syntactic and semantic information of the sentence. In other words, each items of linearized MRS contain essential information that determines the correct surface form. By using Transformer it appears that our model paid attention to the syntactic information of the linearized MRS, but not to the individual lexical items. When the model pays attention to a part of the input sentence, it can be interpreted that the other parts are neglected. This ignorance of the lexical items thus caused the poorer performance with Transformer even when the task was MT, where Transformer generally shows better result.

The present study gives us two findings. First, it is borne out that HPSG-based computational grammars indeed help neural NLG. Following Konstas et al. (2017) and Hajdik et al. (2019), this research still shows that a neural model can faithfully generate sentences in terms of syntax from rich syntactic and semantic representations such as MRS. Second, representations analyzed with a grammar should be handled with care, particularly with modern Attention-based approaches. Unlike natural language texts, grammar-represented texts come with many annotation symbols and few lexical items. This can cause the Attention mechanism to pay less attention to the lexical items embedded in the grammar representation, as in this study.

For future research, we plan to probe the model further by using other test sets. Also, considering Attention mechanism's internal disadvantage of ignoring lexical items, other advanced RNNs that model hierarchical information explicitly can be applied. Finally, one may adjust the Attention mechanism so that it can pay more attention to the lexical items even when they are surrounded by grammar information symbols.

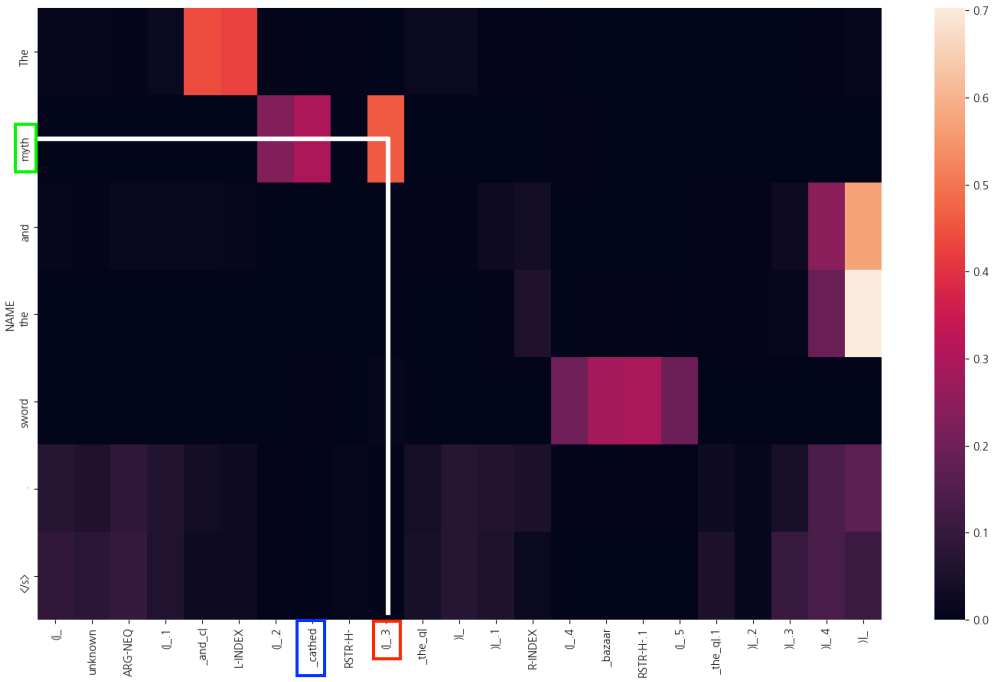

Figure 2: The Attention weight for (2) visualized as a heat map. *myth* payed most of its attention to a functional item (the red box), rather than the lexical word of *cathedral* (the blue box). Brighter color means higher attention weight. Lines are added by the authors for illustration.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Flickinger, D. (2000). On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Goodman, M. W. (2018). *Semantic operations for transfer-based machine translation*. PhD thesis.

Hajdik, V., Buys, J., Goodman, M. W., and Bender, E. M. (2019). Neural text generation from rich semantic representations. *arXiv preprint arXiv:1904.11564*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Konstas, I., Iyer, S., Yatskar, M., Choi, Y., and Zettlemoyer, L. (2017). Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint arXiv:1704.08381*.

Oepen, S., Flickinger, D., Toutanova, K., and Manning, C. D. (2004). Lingo redwoods. *Research on Language and Computation*, 2(4):575–596.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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. Association for Computational Linguistics.

Pollard, C. and Sag, I. A. (1994). *Head-driven phrase structure grammar*. University of Chicago Press.

Post, M. (2018). A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Subramanian, V. (2018). *Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch*. Packt Publishing Ltd.

Taylor, W. L. (1953). "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.