Text Embellishment using Attention Based Encoder-Decoder Model

Subhajit Naskar
University of Massachusetts
Amherst, MA, US
snaskar@cs.umass.edu

Soumya Saha
University of Massachusetts
Amherst, MA, US
soumyasaha@cs.umass.edu

Sreeparna Mukherjee
University of Massachusetts
Amherst, MA, US
sreeparnamuk@cs.umass.edu

Abstract

Text embellishment is a natural language generation problem that aims to enhance the lexical and syntactic complexity of a text. i.e., for a given sentence, the goal is to generate a sentence that is lexically and syntactically complex while retaining the same semantic information and meaning. In contrast to text simplification (Wang et al., 2016), text embellishment is considered to be a more complex problem as it requires linguistic expertise, and therefore are difficult to be shared across different platforms and domain. In this paper, we have explored this problem through the light of neural machine translation and text simplification. Instead of using a standard sequential encoder-decoder network, we propose to improve text embellishment with the Transformer model. The proposed model yields superior performance in terms of lexical and syntactic embellishment and demonstrates broad applicability and effectiveness. We also introduce a language and domain agnostic evaluation set up specifically for the task of embellishment that can be used to test different embellishment algorithms.

1 Introduction

In recent years, deep neural networks have achieved some promising results in natural language tasks such as speech recognition, text generation, and machine translation. (Kim et al., 2015); (Zaremba et al., 2014); (Mikolov et al., 2010). (Rajeswar et al., 2017). Many of these models follow a *teacher forcing* technique, where the model is trained to predict the next word in the sequence given the previous words. This is usually done using maximum-likelihood training of these models. These models are then evaluated based on sequence level metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), etc.

Narrative generation or story generation is a common natural language generation task. Narrative generation pipeline consists of two steps (Yao et al., 2018): First step is to generate a simplified story by a human or a machine learning model. The second step mainly consists of discourse generation, i.e., producing narratives that sound meaningful and appealing to the readers. Our model, which performs text embellishment, can be used in the second step mentioned above. We use the word ”Embellishment” as a method to produce engaging texts out of simplified texts.

Previous researches on similar tasks have mainly dealt with rule-based approaches for discourse, where the model designer has predefined these rules. These rule-based approaches require significant expertise of the language and, more specifically, the field of the text. Furthermore, there is no scope of model generalization, i.e., a model for one type of text may not be used for a different type. Therefore, a domain-independent model for text embellishment has significant importance in the field of natural language generation, particularly narrative generation. A domain-independent model allows the designer to build a light-weight system to generate simple text and use a domain-independent text embellishment model to make their model more diverse in terms of sentence complexity.

The embellishment characteristic of a sentence can be categorized into two types,

- Lexical embellishment: The intent behind lexical embellishment is to replace commonly occurring vocabulary words with their complex counterparts while keeping the overall meaning of the sentence undisturbed. Here complexity is judged based on a similar methodology as *Word Complexity Measure* used for phonological assessment.

- Syntactic embellishment: Syntactic embellishment targets to increase the complexity of a sentence as a whole. This includes both grammatical and structural complexity enhancement.

In this paper, we primarily focus on lexical embellishment, i.e., for a given sentence, we try to generate grammatically correct sentence that has more complex words without changing the meaning of the sentence while exploring the possibility of syntactical embellishment.

The text embellishment task is often interpreted as the inverse to text simplification (TS), which has significant work and literature. Recent works on text simplification systems are capable of simplifying text both lexically and syntactically independent of domains or any predefined rules. Thus, we were encouraged to explore the possibility of a domain-independent text embellishment. However, we do acknowledge that text embellishment is generally a significantly complicated task compared to text simplification. Text simplification often can be regarded as a careful information reduction process, whereas text embellishment requires language generation. So, defining text embellishment as merely the inverse task of text simplification can be misleading.

Instead, we would like to argue that text embellishment shares certain traits with machine translation. Since our goal is to embellish sentences by replacing simple words with more complex words, our goal can be interpreted as a translation task where the source language is **simple** English and the target language is **complex** English.

While exploring past machine learning and natural language literature, we found that domain-independent text embellishment is a relatively unexplored task and seems exciting and promising. Furthermore, there is no prior work on domain-independent text embellishment that was able to show promising results in terms of either lexical embellishment or syntactic embellishment. In this paper, we solely explore the possibility of neural encoder-decoder architecture in developing a domain-agnostic text embellishment system. We use Transformer based architecture to improve embellishment quality and compare the results with seq2seq based architecture used on the same task.

2 Related Work

Research in computational narrative traces back to the 1960s and 1970s, intending to instill narrative intelligence in machines. One of the most well-known generation systems is TALE-SPIN, which produces narratives by emphasizing problem-solving techniques (Meehan, 1977). While this work focused on the idea that events follow each other sequentially, the work of (Callaway and Lester, 2002) explicitly addressed the gap between computational narrative and Natural Language Generation. These works use rule-based language models that produce naturally sounding narratives. Even though these approaches make use of text embellishment, the main disadvantage is that these rules have to be devised before the system's architecture design, which limits the performance of the model.

Another close area of research in this domain is incorporation of linguistic style. Here, style can refer to features of lexis, grammar, and semantics, which is individual to a particular author or a specific situation. While research in this area started with rule-based methods, but from the early 2000s, the shift has been towards a data-driven approach. (Paiva and Evans, 2005) developed an algorithm that identifies a series of local decisions that maximize the desired stylistic capacity. More recently, (Ficler and Goldberg, 2017) demonstrate controlling several stylistic variations in generated text through conditioned language models. On the contrary, we are trying to make the machine learn natural language representation from human data and enhance narratives that have been generated before in a domain-independent manner.

This approach can be compared with other existing works in the field of natural language processing such as statistical machine translation, text summarization, and, most importantly, text simplification. If we look at the summarization and simplification task, we will understand that the main goal is to extract the necessary information, maintain a natural language structure and remove linguistic embellishment that is not necessary for understanding the context. Thus, we see that our work is complementary in its objective to both of these tasks. Thus, the applications of text simplification include reducing the complexity of a natural language sentence by focusing on the discourse level aspects of syntactic simplification (Siddharthan, 2002), whereas on the other hand

(Coster and Kauchak, 2011) aims to reduce the reading complexity of a sentence by incorporating more accessible vocabulary and sentence structure. Thus, here in our task is more similar to the later as we aim to increase the complexity of a simple sentence by adorning its vocabulary with more complex words. Another prominent work is this field by (Shardlow, 2014), where they distinguish between syntactic and lexical simplification. Syntactic simplification aims to reduce the complexity of sentence structure, whereas lexical simplification aims to replace difficult vocabulary with simpler words. Previously, these two types of simplification tasks have been addressed separately. ((Siddharthan, 2006), (Biran et al., 2011), (Paetzold and Specia, 2017)). More recently, we have seen that both these tasks have been addressed simultaneously ((Wang et al., 2016), (Zhang and Lapata, 2017)). These tasks address this problem as an extension of machine translation and borrow ideas from automatic natural language generation (Wen et al., 2015). So, the problem of simplification comes down to monolingual machine translation, where the goal is to translate from a **complex** English to a **simple** English. Following the recent success of neural machine translation, we see an increased use of LSTM based encoder-decoder architecture in these tasks. ((Bahdanau et al., 2014), (Cho et al., 2014)). Work by (Hochreiter and Schmidhuber, 1997) on Long Short-term Memory architecture has long been used to solve sequence to sequence tasks where both the input and output sequence can be of varied length.

Although we have seen significant progress in the domain of text simplification, little work has been done in text embellishment. What makes text embellishment promising right now are the vast corpora on which text simplification tasks have been trained for more than two decades. However, text embellishment is a much more difficult task than text simplification as adding of information might lead to an introduction of semantic contradictions.

Nevertheless, quite recently, a promising work in this field has been done by (Berov and Standvoss, 2018), motivated from researches in text simplification and machine translation. The authors have proposed a network design similar to what (Wang et al., 2016) have used in their text simplification work, which uses a Long Short-Term Memory (LSTM) Encoder-Decoder model for sentence-level text simplification as it makes minimal assumptions about word sequence. We replicate their model and use it as our baseline model for evaluation and experimentation.

In this paper, we focus on neural encoder-decoder architecture in developing a domain-agnostic text embellishment system. To our knowledge, there are no existing linguistically motivated, non-neural architecture for text embellishment. However, one can see how a word or phrase substitution method can be employed using a predefined mapping between simple word/phrase to complex word/phrase. This can be done using Wordnet (Miller, 1995) as shown by (Tambe et al., 2019) in context of text simplification. However, such substitution may lead to incorrect substitution in the context of a specific domain. As, a substitution that might be valid for a literary text(novel, short story) may be incorrect and misinforming for a different domain such as a scientific journal. That will necessitate domain-specific rule design. As discussed in (Tambe et al., 2019), Such methodology requires additional steps such as word sense disambiguation, lexical simplification, which are out of the scope of this paper. Furthermore, such substitution will limit the possibility of syntactic embellishments, such as structural and grammatical complexity enhancement. Therefore, we avoid discussing the substitution based embellishment strategy in this paper.

3 Methodology

We achieve the goal of embellishing a sentence by modeling the distribution of the embellished sentence given the simple sentence. i.e. $P(Y|X)$ where X denotes the simple sentence and the words of the simple sentence is denoted as $x_1, x_2, ..., x_n$. Similarly, Y denotes the embellished sentence, and the words of the embellished or target sentence are $y_1, y_2, ..., y_m$. We model our task similar to a machine translation task and employ an encoder-decoder architecture. The encoder consumes the input text and computes a representation of context vector c. The decoder generates one target word given the context vector c and all the previous predicted words $\bar{y_1}, ..., y_{\bar{t}-1}$.

$$p(y) = \prod_{t=1}^{T} p(y_t|\bar{y_1}, ..., y_{\bar{t}-1}, c) \qquad (1)$$

For both of our models, we used named entity masking and byte pair encoding in input sentences and beam search decoding while generating embellished sentences.

3.0.1 LSTM Architecture

From the work of (Wang et al., 2016), we can see that the LSTM Encoder-Decoder model can learn operational rules such as reversing, sorting, and replacing from sequence pairs. This shows such Encoder-Decoder model may potentially apply rules like modifying sentence structure, substituting words, and removing words for text simplification as well as text embellishment.

We chose our model as 3 LSTM layers having 300 hidden units for each encoder and decoder. All weights were uniformly initialized as [-0.1, 0.1]. We have used Harvards OpenNMT PyTorch framework (Klein et al., 2017) to set up the above network and used the model for our task. We have used a system having 8 core CPU with 2 NVIDIA P100 GPU to train our network.

3.1 Transformer architecture

The LSTM based encoder-decoder model operates sequentially using recurrence. Compared to sequence to sequence models, the Transformer processes all words or symbols in the sequence in parallel while making use of a self-attention mechanism to incorporate context from words farther away from it. By processing all words in parallel and letting each word attend to other words in the sentence over multiple processing steps, the Transformer is computationally much more efficient and gives superior performance in many natural language processing tasks. However, in our case, we will focus on the sequential nature of LSTMs and limitations. i.e., it is prone to be inferior in handling long term dependencies (even with attention). However, in our case, we require an architecture that is capturing semantic information and long term dependencies effectively. This motivated us to use the transformer model as it comes out as a promising architecture to address this problem.

The architecture we used in our implementation consists of 6 identical layers for each encoder and a decoder network with all the sublayers having 512 units and 8 parallel attention layers or heads. For our best performing model, we used Byte-pair-encoding on the input text. The model was trained on 2 NVIDIA GTX 1080 Ti GPUs.

4 Datasets

To train the model, we are using the WikiLarge dataset, constructed by (Zhang and Lapata, 2017). The dataset consists of 256252 aligned sentences for training, 854 aligned sentences for validation, and 358 sentences for test. We have chosen this particular dataset, primarily because this is the largest sentence-aligned dataset which is widely used for text simplification task. ((Xu et al., 2016) , (Vu et al., 2018) (Zhao et al., 2018))

Thus, since this task is complementary to text simplification, we have interchanged the source and target datasets, and now the goal of the model will be to produce ”complex” sentences from the ”simple” input sentences.

5 Results

The LSTM encoder-decoder model was able to achieve some basic lexical replacements (*found* → *discovered*, *stayed* → *remained*) and grammatical corrections such as character case-correction, punctuation correct. For example, *It was* ***found*** *by PERSON@1 ...* → *It was* ***discovered*** *by PERSON@1 ...*, ***the*** *former district PERSON@1 ...* → ***The*** *former district PERSON@1....*

It is to be noted, such instances of lexical embellishment were relatively limited, and in most cases, the output is identical to the input sentence. and there were no such instances of syntactic embellishment.

However, in the case of the Transformer model, there was a significantly high number of lexical embellishment. Also, along with one-one lexical replacement(replacing a single word with a more complex synonym), Transformer was able to replace POS phrases with more complex word(*very very* → *extremely*). Which was more impressive and noteworthy was, in some instances, we observed syntactic embellishment as well.

5.1 Lexical embellishment

Entrance to LOCATION@1 is ***very very*** *difficult* → *Entrance to LOCATION@1 is* ***extremely*** *difficult.*

Their culture ***is similar to*** *the culture ... The culture of LOCATION@1* ***is closely associated with*** *the culture ...*

5.2 Syntactic embellishment

It is a starting point *for people wanting ...* → ***It also serves as a starting point*** *for people wanting*

...,

*... appears as **a stretched** object . A stretched object]**was the major axis .It pointing towards Uranus** → appears as **an elongated** object,with the **major axis pointing towards Uranus***

From this example, we can see that our proposed model was able to achieve a more complex type of lexical embellishment and some impressive syntactic embellishment.

6 Evaluation

Primarily, we evaluate our models using two different evaluation setup: BLEU, readability scores. However, these standard metrics have certain limitations and may not always be sufficient for evaluating the embellishment capability of a model. Therefore, we design a human evaluation setup that is suitable for our task.

6.1 BLEU

To measure the proximity of generated output sentence's context to the original sentence, we have used BLEU score, which is an automated method to evaluate machine translation tasks. The main purpose of BLEU metric is to evaluate the *closeness* of a machine-generated translation with reference to its human translation. Now, in the context of our task, we use this metric to evaluate if the context of the sentence has been changed or not. Thus, a high BLEU score would indicate that the context is close or similar to the original sentence, whereas a low BLEU score would indicate that context has changed. One point to be noted is that this evaluation metric does not measure the level of embellishment in the hypothesis sentence compared to the reference sentence. In the following table, we present the BLEU scores of the best models we have trained.

Measure	LSTM	Transformer
BLEU	91.04	66.10

Table 1: BLEU with source sentence

From the high BLEU score of LSTM network, it can be inferred that LSTM network mainly learned to reproduce the input correctly without modifying any words in the source sentence. While the Transformer model produces sentences with a lower BLEU score, it does not provide us any significant information regarding its capability for embellishment. Furthermore, BLEU is often considered unsuitable and controversial for the language generation task, as argued by (Reiter, 2018). Thus, we have used Readability Measurements for that purpose.

6.2 Readability measures

To measure the complexity of the generated sentences, we are using three measures based on the readability of text. We have evaluated the generated sentences of both our models using these three measures, *Flesch Reading Ease score (FRES)*, *Flesch-Kincaid Grade Level (FKGL)* and *SMOG Index*. More details and corresponding equations used for each of the Readability Measures have been described in the Appendix section. In the following diagram, we have shown the readability statistics of the output of the LSTM and Transformer. We evaluate our test dataset based on the three different readability metrics and record the percentage of sentences where the readability scores increased, decreased, or stayed the same after embellishment.

From, figure 1, we can see that for all readability metrics, the percentage of data where the input and output sentence has the same readability score is significantly high in the case of LSTM. Which, further confirms the result reported by (Berov and Standvoss, 2018), that LSTM model is prone to copying the input to output without achieving any embellishment. However, if we compare the ratio of data with increased readability level, we see, Transformer model shows better embellishment performance. However, the percentage of data with readability is also high for Transformer. To inspect that phenomenon, we employ human evaluation and design our task-specific evaluation setup that will shed more light on this issue.

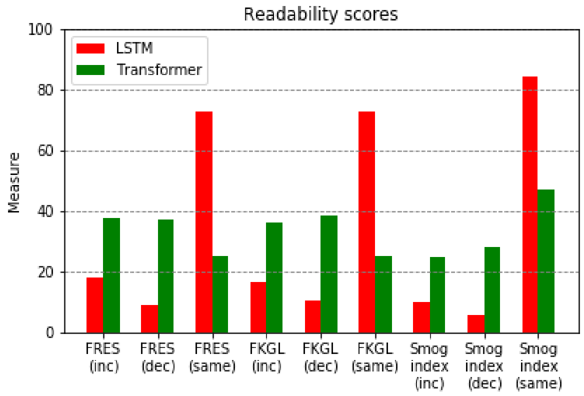

Figure 1: Readability Scores

6.3 Human evaluation

The shortcoming and limitations of the aforementioned evaluation metrics motivate a human evaluation process. For human evaluation, we designed metrics to score generated results. We employed 12 human evaluators who are proficient in English and are capable of evaluating the complexity of a text. Evaluators were randomly assigned to a group of 4, and 3 groups were formed to score model-generated results based on the scoring metrics. Then we used this scores to devise 5 model performance metrics, namely contextual capacity, generative capacity, consistency measure that measure a different aspect of how contextually coherent and meaningful the results are, and embellishment capacity and conditional embellishment capacity that measures the model's ability to embellish a given simple sentence. First, we will define the scoring metrics used by the evaluators to score the generated results. Then we will discuss the performance metrics.

6.3.1 Score metrics

The underlying goal of this task to design a system that can generate a meaningful, contextually identical, and embellished sentence for a given input sentence. Therefore, we designed scoring metrics to capture exactly that. We asked our evaluator to score each generated sentence on a categorical scale of 0,1,2 for the following three metrics.

- **Grammatical Coherence score**: If embellished sentences were grammatically correct and are a meaningful sentence.
- **Context Coherence score**: If the embellished sentence is within the same context of the simple source sentence.
- **Embellishment score**: If the generated sentence is overall more complex than the source sentence. If the model achieves generate a structurally complex sentence, that will be considered a successful embellishment. If the model manages to replace words, We asked participants to evaluate each word replacement based on whether the embellished words were more complex and if such replacement is leading to the embellished sentence becoming more complex.

6.3.2 Aggregate Performance metrics

The aforementioned scoring metrics are used to calculate the performance metrics defined below.

- **Contextual Capacity**: Model's capacity to generate contextually correct sentences.
- **Generative Capacity**: Model's capacity to generate contextually and grammatically correct sentences.
- **Embellishment Capacity**: Model's capacity to generate embellished sentences.
- **Conditional Embellishment Capacity**: Model's capacity to achieve embellishment given that the model generates contextually and grammatically correct sentences.
- **Consistency Measure**: Model's consistency of generating embellished sentence or the same sentence was given that the model always generates contextually and grammatically correct sentences.

For the sake of brevity, the categorical definition of scoring metrics and calculation procedure of performance metrics are documented in the Appendix.

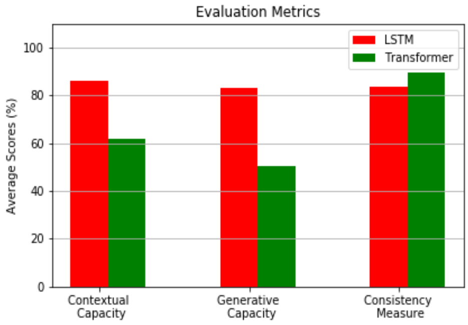

Figure 2: Evaluation metrics

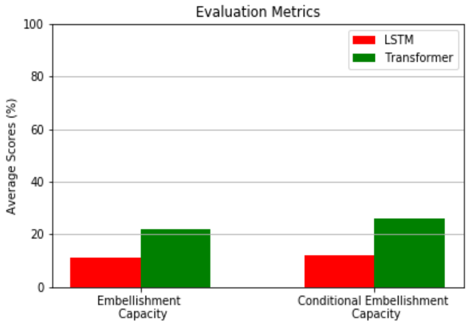

Figure 3: Evaluation metrics

From Figure 2, we can see that the LSTM model generates more contextually and grammatically consistent sentences, i.e., there are significantly fewer cases of random embellished output. The reason being, LSTM model, fails gracefully, i.e., when it fails to generate a lexically or syntactically complex sentence, it simply copies the input words to the output sentence. This same phenomenon was also recorded by (Berov and Standvoss, 2018) as well. Moreover, that is where we were able to achieve significant improvement. In figure 3, if we compare the embellishment capacity, the Transformer is significantly better (nearly double embellishment capacity) compared to LSTM encoder-decoder model. If we condition our evaluation on grammatically and contextually consistent outputs only, the transformer model outperforms LSTM encoder-decoder significantly. This may indicate that, when Transformer model can generate a contextually and grammatical consistent sentence, it has a significantly better power to achieve text embellishment.

7 Conclusions

Text embellishment is quite an unexplored track in the research field of Natural Language Generation because it would require a massive amount of data, training hours as well as various idiosyncratic, hand-coded rules to get performance which is close to human efficiency.

In this paper, the results from the LSTM encoder-decoder or the transformer network cannot be used for production purposes yet. Maybe, it is because the network is not able to learn all the nuances of human languages in a specific domain with the help of a dataset that is flawed with few grammatical and typographical errors. The availability of sentence aligned massive datasets that are more apt for this specific task is also rare.

In this paper, we show that our LSTM network achieved a BLEU score of 91.04, which is closer (92.13) to what (Berov and Standvoss, 2018) achieved with similar LSTM architecture. However, we showed why such measurements can be misleading and which motivated us to design a task-specific human evaluation setup. Based on the evaluation setup, we showed how transformer architecture is significantly better for the text embellishment task. Thus our initial assumption that in the case of generation tasks, especially text embellishment, a self-attention based architecture performs better than a seq2seq model with attention, holds. This is because these models are more capable of capturing the semantic information of a sentence.

8 Future work

In this paper, we aimed to work on the task of text embellishment for a single sentence. The same methodology and architecture can be extended to paragraphs, where we intend to generate a lexically and syntactically complicated paragraph given a simple paragraph. Such a task may require a hierarchical attention mechanism where we attend to single words as well as sentences to capture the semantics of a sentence and paragraph.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Leonid Berov and Kai Standvoss. 2018. Discourse embellishment using a deep encoder-decoder network. *arXiv preprint arXiv:1810.08076*.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: A context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 496–501, Stroudsburg, PA, USA. Association for Computational Linguistics.

Charles B. Callaway and James C. Lester. 2002. Narrative prose generation. *Artif. Intell.*, 139(2):213–252.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

William Coster and David Kauchak. 2011. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 665–669, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *CoRR*, abs/1707.02633.

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

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. *CoRR*, abs/1508.06615.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

James R. Meehan. 1977. Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pages 91–98, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 34–40.

Daniel S. Paiva and Roger Evans. 2005. Empirically-based control of natural language generation. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 58–65.

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*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Joseph Pal, and Aaron C. Courville. 2017. Adversarial generation of natural language. *CoRR*, abs/1705.10929.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Proceedings of the Language Engineering Conference (LEC'02)*, LEC '02, pages 64–, Washington, DC, USA. IEEE Computer Society.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Mrunmayee Tambe, Preeti Ballal, Vishal Dolase, Kajol Agrawal, and Yogesh Rajmane. 2019. *Lexical Text Simplification Using WordNet*, pages 114–122.

Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Sentence simplification with memory-augmented neural networks. *CoRR*, abs/1804.07445.

Tong Wang, Ping Chen, Kevin Michael Amaral, and Jipeng Qiang. 2016. An experimental study of LSTM encoder-decoder model for text simplification. *CoRR*, abs/1609.03663.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Peihao Su, David Vandyke, and Steve J. Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *CoRR*, abs/1508.01745.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2018. Plan-and-write: Towards better automatic storytelling. *CoRR*, abs/1811.05701.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. *CoRR*, abs/1810.11193.

9 Appendix

9.1 Human evaluation method

The readability measurements are apt for evaluation of human-generated embellishments, but they often fail in case of machine-generated text embellishments. Thus to provide leniency considering the capabilities of various deep learning architectures in this particular task, we have decided to evaluate our model using a small survey. We believe human judgment will be the best in understanding the level of embellishments in sentences produced using our methods. These groups of judges were asked to score the generated sentences based on lexical embellishment and grammatical and context coherence.

- Grammatical Coherence: Each participant were asked to judge if the embellished sentences were grammatically correct and is a meaning sentence.

Score	Interpretation
0	Not grammatical and meaningful
1	Grammatically partially correct
2	Grammatically correct

Table 2: Rubric for Grammatical coherence score

- Context Coherence: Each participant were asked to judge if the embellished sentence is within the same context of the simple source sentence.

Score	Interpretation
0	Deviated from context
1	Partially correct context
2	Correct context

Table 3: Rubric for Context coherence score

- Embellishment: We asked each participant to evaluate each word replacement based on whether the embellished words were more complex or the embellished sentence was more complex. Each participants assigned a score between 0 to 2 for each sentence.

Score	Interpretation
0	Not meaningful sentence
1	Same sentence or hard to decide
2	Embellishment

Table 4: Rubric for Embellishment score

For ease of evaluation, we also asked the participants to judge the source or simple sentences based on their grammatical correctness, sentence structure to decide if the sentence is a correct English sentence and has any scope of textual embellishment.

Score	Interpretation
0	Grammatically incorrect
1	Confusing or conveys no meaning
2	Proper English sentence

Table 5: Rubric for source score

This source score will help us record the number of sentences in the test data that can be embellished. However, For human evaluation based measurements we will discard all sentences with source score of 0 and 1.

Based on these scores, we recorded the following:

- Context-wise correct: Number of embellished sentences with context coherence score(C) of 2, denoted as $f(C = 2)$.
- Grammatically and context-wise correct: Number of embellished sentences with context coherence score(C) 2 and grammatical coherence score(G) 2, denoted as $f(C = 2, G = 2)$.
- Grammatically, context-wise correct and good embellishment: Number of embellished sentences with context coherence score(C) 2 and grammatical coherence score(G) 2 and embellishment score 2, denoted as $f(C = 2, G = 2, E = 2)$
- Grammatically, context-wise correct and partial embellishment: Number of embellished sentences with context coherence score(C) 2 and grammatical coherence score(G) 2 and embellishment score 2, denoted as $f(C = 2, G = 2, E = 1)$

Based on this, we will derive the following performance measures where N is the total number of test sentences used:

- **Contextual Capacity**: Model's capacity to generate contextually correct sentences.

$$Contextual\ Capacity = \frac{f(C = 2)}{N}$$

- **Generative Capacity**: Model's capacity to generate contextually and grammatically correct sentences.

$$Generative\ Capacity = \frac{f(C = 2, G = 2)}{N}$$

- **Embellishment Capacity**: Model's capacity to generate embellished sentences.

$$Embellishment\ Capacity = \frac{f(E = 2)}{N}$$

- **Conditional Embellishment Capacity**: Model's capacity to achieve given that the model always generates contextually and grammatically correct sentence.

$$Conditional\ Embellishment\ Capacity = \frac{f(C=2,G=2,E=2)}{f(C=2,G=2)}$$

- **Consistency Measure**: Model's consistency of generating embellished sentence or same sentence given that the model always generates contextually and grammatically correct sentence.

$$Consistency\ Measure = \frac{f(C=2,G=2,E=2)}{f(C=2,G=2)} + \frac{f(C=2,G=2,E=1)}{f(C=2,G=2)}$$

9.2 Readability Measures

- **Flesch Reading Ease score (FRES)**

  The Flesch Reading Ease score, developed by Rudolf Flesch, is the most commonly used readability measure. The score on the test will tell us roughly what level of education someone will need to be able to read a piece of text easily. The Reading Ease formula generates a score between 1 and 100. The formula for the Flesch reading ease score (FRES) test is

$$\text{FRES} = 206.835 - 1.015 * \frac{\text{total words}}{\text{total sentences}} - 84.6 * \frac{\text{total syllables}}{\text{total words}}$$

- **Flesch-Kincaid Grade Level (FKGL)**

  The Flesch-Kincaid Grade Level index is one way to measure and report the readability of English text. Both Flesch reading ease and Flesch-Kincaid grade level use the same core metrics: word length and sentence length. But they correlate inversely. If one receives a high score on the reading ease test, one should receive a lower grade level score. The FKGL formula presents a score as a U.S. grade level, making it easier for teachers, parents, librarians, and others to judge the readability level of various books and texts. It can also mean the number of years of education generally required to understand this text, relevant when the formula results in a number greater than 10. The grade level is calculated with the following formula:

$$\text{FKGL} = 0.39 * \frac{\text{total words}}{\text{total sentences}} + 11.8 * \frac{\text{total syllables}}{\text{total words}} - 15.59$$

- **SMOG Index**

  The SMOG Index is also a measure of readability that estimates the years of education needed to understand a piece of writing. What is different from the above two evaluation measures is it considers the number of polysyllables (words of 3 or more syllables) whereas the above two measures consider average number of syllables per sentence. The formula for calculating SMOG index is:

$$\text{SMOG} = 1.043 * \sqrt{\text{\# of polysyllables}} * \sqrt{\frac{30}{\text{total sentences}}} + 3.1291$$

9.3 Additional results

9.3.1 LSTM encoder-decoder

*It was **found** by PERSON@1 in images from the Voyager NUMBER@1 → It was **discovered** by PERSON@1 in images from the Voyager NUMBER@1*

*This stamp **stayed** the standard letter stamp for the rest of PERSON@1 's reign, and many were printed → This stamp **remained** the standard letter stamp for the rest of PERSON@1 's reign , and many were printed .*

*In the year NUMBER@1 ,the population was NUMBER@2 . → The population was NUMBER@1 at the NUMBER@2 **census** .*

*the former district PERSON@1 , also resembles the upper half of the coat of arms . → **The** former district PERSON@1 , also resembles the upper half of the coat of arms .*

In December , NUMBER@1 , PERSON@1 was honored as part of the Righteous Among the Nations by the State of LOCATION@1 . → In December NUMBER@1 , PERSON@1 was honored as part of the Righteous Among the Nations by the State of LOCATION@1 .

9.3.2 Transformer

It is a starting point *for people wanting to explore LOCATION@1 , LOCATION@2 and LOCATION@3 . →* ***It also serves as a starting point*** *for people wanting to explore LOCATION@1 , LOCATION@2 and LOCATION@3 .*

Their culture ***is similar to*** *the culture of the coastal peoples of LOCATION@1 . → The culture of LOCATION@1* ***is closely associated with*** *the culture of the coastal people of LOCATION@1*

entrance to LOCATION@1 is ***very very*** *difficult . → entrance to LOCATION@1 is* ***extremely*** *difficult .*

ORGANIZATION@1 ***named*** *him " Sportsman of the Year " in NUMBER@1 . → ORGANIZATION@1* ***crowned*** *him " Sportsman of the Year " in NUMBER@1.*

Early September NUMBER@1 , dry air wrapping around the southern area of the cyclone caused most of the heat to ***leave*** *. → Early September NUMBER@1 , dry air wrapping around the southern area of the cyclone caused most of the heat to* ***evacuate*** *.*

At the Voyager NUMBER@1 pictures PERSON@1 appears as ***a stretched object*** *. A stretched object was the major axis . It* ***pointing towards Uranus*** *. → At the Voyager NUMBER@1 pictures PERSON@1 appears as an* ***elongated object, with the major axis pointing towards Uranus*** *.*

Some ***clauses*** *are rather lengthy and rich in content while others are shorter -LRB- possibly stubs -RRB- and of lesser quality . → Some* ***language content*** *are rather lengthy in content while others are shorter -LRB- possibly stubs -RRB- and of lesser quality .*

In NUMBER@1 PERSON@1 was inducted into the Rock and ORGANIZATION@1 . → In NUMBER@1 , PERSON@1 was inducted into the Rock and ORGANIZATION@1 .