

Discourse Embellishment Using a Deep Encoder-Decoder Network

Leonid Berov

Institute of Cognitive Science
University of Osnabrück, Germany
lberov@uos.de

Kai Standvoss

Donders Institute for Brain, Cognition
and Behaviour
Radboud University, The Netherlands
kstandvoss@uos.de

Abstract

We suggest a new NLG task in the context of the discourse generation pipeline of computational storytelling systems. This task, textual embellishment, is defined by taking a text as input and generating a semantically equivalent output with increased lexical and syntactic complexity. Ideally, this would allow the authors of computational storytellers to implement just lightweight NLG systems and use a domain-independent embellishment module to translate its output into more literary text. We present promising first results on this task using LSTM Encoder-Decoder networks trained on the Wiki-Large dataset. Furthermore, we introduce “Compiled Computer Tales”, a corpus of computationally generated stories, that can be used to test the capabilities of embellishment algorithms.

1 Introduction

Narratives can be analyzed to consist of at least two layers: plot—what is told—and discourse—how it is told. Usually, computational systems first generate the events of the plot, and then decide how to render these in text (Gervás, 2009). This means that a tight coupling exists between the knowledge bases that are used for plot generation, and the NLG modules for discourse rendering. The result is that it is hard to share these modules between systems, while the implementation of a custom module to generate literary discourse requires significant effort and linguistic expertise. Indeed, our subjective feeling is that often it is possible to recognize repeating textual patterns when reading several stories created by the same system.

To alleviate this problems we suggest a new

NLG task, *textual embellishment* (TE), with the goal to automatically increase the lexical and syntactic complexity of a text. Textual embellishment can be understood as the inverse task to textual simplification (TS), which has been researched for at least two decades (Shardlow, 2014). Recent TS systems are trained on vast corpora using machine learning techniques and work independent of domains or handcrafted rules. These results open the intriguing possibility for an equally general resource for TE. This could allow the authors of storytelling systems to save time by implementing only thin NLG modules and perform subsequent embellishment. We would like to strongly caution that TE most likely will prove to be a more difficult task than TS. While the latter typically results in removing information in a text, the former might lead to an automated adding of information which can introduce semantic contradictions. However, on a syntactic level TE ideally just results in periphrastic constructions with no added information.

2 Related Work

Traditionally, narrative generation systems like Meehan (1977); Lebowitz (1985) were devised to generate macro-level plot and character elements without connecting them into a comprehensive story in natural language. Later work has sought to combine plot and NLG into a unified system (Callaway and Lester, 2002). These full pipelines generate story elements and employ rule-based language models that produce naturally sounding narratives. While these approaches make use of text embellishment, the disadvantage of these rule-based approaches is that rules for text enrichment have to be conceived of in advance by the system’s architect. The work presented here, in contrast, seeks to learn a natural language representation

from human data to automatically embellish priorly generated narratives.

This approach can be located in the field of natural language processing. Within this area of research several related subfields can be identified, including text summarisation, statistical machine translation, and most notably, TS. Both, text summarisation and simplification, seek to extract the relevant information of a phrase and lose all linguistic embellishment that is deemed unnecessary for its understanding. Our work has a complementary objective, which is enriching text while maintaining its original meaning. This generative task is related to recent developments in computer vision using Generative Adversarial Networks for problems like image super resolution (Ledig et al., 2017). Therein similar challenges are faced as new information has to be “hallucinated”, requiring particular caution in content sensitive domains.

Within TS, Shardlow (2014) distinguishes between syntactic and lexical simplification. While the former approach seeks to facilitate readability by reducing the complexity of sentence structure, the latter seeks to replace words deemed difficult for a target audience by words that are easier understandable. Previous work has focused on both tasks individually (Siddharthan, 2006; Paetzold and Specia, 2017), while recent approaches addressed syntactic and lexical simplification simultaneously (Wang et al., 2016; Zhang and Lapata, 2017). These latter systems cast text simplification as a monolingual machine translation problem and borrow insights from automatic natural language generation (Wen et al., 2015). Therein, the task of simplifying text is framed as a translation problem between “complex English” and “simple English”. The recent success of Deep Learning systems in “neural machine translation” (Bahdanau et al., 2014; Cho et al., 2014) has increased the employment, specifically of the LSTM encoder-decoder architecture (Sutskever et al., 2014). These models make use of the Long Short-Term Memory neural network architecture (Hochreiter and Schmidhuber, 1997) to solve a sequence to sequence task. Therein, a mapping from a *source* sequence to a *target* sequence is learned both of which may be of variable length. To that goal, an *encoder* LSTM generates a sequence of hidden representations using the source, which is then used by a second *decoder* LSTM to generate the target sequence word by word, whereof each is conditioned on all pre-

vious outputs. Here, we follow the same approach but address the opposite problem to TS, translating from “simple” to “complex” English, building on the success of previous neural network approaches.

3 Discourse Embellishment

For the trained network, two LSTM layers per encoder and decoder were chosen, since Wang et al. (2016) demonstrated that this is sufficient for sorting, reversing and reordering operations in a textual simplification task. The network was set up and trained using Harvard’s OpenNMT-tf framework (Klein et al., 2017). Parameters were selected as suggested by Zhang and Lapata (2017): Each layer contained 512 hidden units with weights uniformly initialized to $[-0.1, 0.1]$, learning was performed with a 0.2 dropout rate (Zaremba et al., 2014) and Luong attention was used by the decoder (Luong et al., 2015). If not stated otherwise, training was always performed for 24 epochs, with a learning rate of 1 and a simple gradient descent optimization, where gradients were clipped when their norm exceeded 5 (Pascanu et al., 2013). After 18 epochs a decay rate of 0.25 was applied each epoch. Furthermore, pre-trained 300 dimensional GloVe vectors were used to initialize word embeddings (Pennington et al., 2014), and the vocabulary size was 50,000.

3.1 Story Corpus

The performance of a discourse embellishment system should ideally be judged on domain-specific texts. To the best of our knowledge no corpus of computationally generated stories has been compiled so far. Since no common resources for NLG generation have been deployed, the stories that have been generated by different systems over the decades differ markedly in their language.

To reflect this diversity we decided for a breadth-focused approach while setting up *Compiled Computer Tales* (CCT), a corpus of computationally generated stories. Since the aim of this corpus is to be used as a qualitative test set, and not as training data for machine learning algorithms, we collected at most three stories from as many systems as available to us, instead of compiling many stories from few systems. Because it is unfeasible to deploy all individual systems, we instead opted for using stories that have been reported in scientific publications, although this runs

the danger of biasing the corpus towards high-quality exemplars.¹ The resulting corpus can be found online² and we cordially invite researchers to expand this necessarily incomplete collection.

At the moment of writing, it contains 14 stories from 8 storytellers, ranging from early systems like “Novel Writer” (Klein et al., 1973) to recent ones like “Simlextrix” (Riegl and Veale, 2018). It is formatted in a way that allows separation based on system, story, paragraph (45 in total) or sentence (290 in total); depending on the task at hand. We also provide a python script that can be used to perform such splits, as well as common preprocessing steps like tokenization or named entity anonymization.

3.2 Sentence Based: Lexical Embellishment

As with the more common TS task, we trained our network on a sentence-aligned dataset. The largest simplification dataset is, to the best of our knowledge, WikiLarge, which contains 296,402 automatically aligned sentences from the ordinary and simple English Wikipedias in its training set. To reduce vocabulary, named entities in the dataset were anonymized (Zhang and Lapata, 2017).

Unlike for simplification, for the present task the network was trained to generate original (‘complex’) English based on simple English input. The performance of the system is evaluated using the common BLEU metric (Papineni et al., 2002), which measures the degree to which generated embellished text differs from the language employed in the original Wikipedia. Training is continuously evaluated on a held-out evaluation set using the Moses toolkit (Koehn et al., 2007), and BLEU stagnates after around 18 epochs. The final performance on the held-out test set is a BLEU of 56.02 on the tokenized data, which is a reasonable performance³.

To judge the performance of this model on CCT, the corpus had to be split in sentences, tokenized, and named entities had to be anonymized in a manner comparable to WikiLarge. Since no gold-

standard of embellished tales exists, quantitative evaluation of embellishment quality is not possible. Comparing the embellished version with the original ones shows that the network mostly learned to correctly reproduce its input (BLEU 92.13). If differences that are due only to out-of-vocabulary words and minor formatting differences are ignored, then roughly 83% of the sentences are simply reproductions of the input. Around 5% of the generated sentences do not bear any resemblance to the corresponding input, which we consider to happen when the network overfitted a certain type of input. 2% of the generated sentences miss individual words. The remaining 10% contain syntactically correct lexical substitutions, which can be considered cases of lexical embellishment. Some examples of felicitous embellishment are the mappings: *Because of this* → *consequently* || *tried* → *attempted* || *wanted to* → *sought to* || *bossy* → *overbearing*; while examples of failed embellishment are: *sinful* → *drinking* || *lioness* → *cat* || *ubiquitous* → *familiar* || *towards princess* → *towards LOCATION@1*.

3.3 Paragraph Based: Degradation

Sentence aligned training enabled the network to perform lexical embellishment, however, no syntactical embellishment like concatenation or parenthetical insertion was observed. It stands to reason that such sentence-level reordering can be better picked up by learning from a corpus that aligns whole paragraphs, where several sentence are simplified/embellished at a time. A promising candidate is the document aligned Wikipedia corpus compiled by Kauchak (2013). After the necessary preprocessing and splitting steps the resulting dataset contained 134,233 paragraphs in the training set.

Training a network on this dataset using the same setup was not successful. The model achieved a BLEU score of 3.55 on the held-out test set, which is a clear sign of its failure.

A manual inspection of the dataset revealed, that the employed alignment of paragraphs is rarely meaningful, since paragraphs of the same ordinal number often fail to contain semantically related information, let alone simplified/embellished versions of each other. In fact, further literature review did not reveal work on either paragraph or document aligned simplification, as well as no published results using the document

¹ A notable exception is “Simlextrix” (Riegl and Veale, 2018), which is accessible online and was used to generate novel stories for the corpus.

²<https://github.com/cartisan/CompiledComputerTales>

³ The state of the art system from 2014 achieves a worse BLEU of 48.97 on the considerably easier textual simplification task using this dataset. However, current state of the art strongly outperforms this with a BLEU of 88.85, again, on the simpler text simplification task (Zhang and Lapata, 2017).

aligned dataset. This lead us to suspect that the encountered problems are not of trivial nature, and abandon this direction of inquiry.

3.4 Pair Based: Syntactical Embellishment

An inspection of the WikiLarge dataset revealed that the sought-for syntactical reorderings can also be found there. In particular, the sentence-alignment does not necessarily result in a bijective mapping from complex to simple sentences, but sometimes in a 1:2 mapping, when complex sentences are split in two. This lead us to the hypothesis that the network described in section 3.2 should already be able to perform syntactical embellishment, when presented with two input sentences.

To test this we pre-processed the CCT corpus by concatenating, with a space, two consecutive sentences. This resulted in sentence-pairs that in most cases had the potential to be combined, because they shared subjects or were causally related. In some cases, syntactical combinations were not meaningful, especially when sentence pairs crossed paragraph or story boundaries as a result of the unsophisticated concatenation operation.

Comparing the embellished version with the original ones shows that the network is doing an acceptable job in keeping original and embellishment related (BLEU 68.20). Approximately 56% of the generated pairs were combined by the network into one sentence. In 42% of the cases the network did not combine the pairs, in which case it usually kept the first sentence and removed the second. In the remaining 2% the networks output was not related to the input.

The combination of sentences was performed by the network in different ways. Most commonly, it conjoined them using a comma and the conjunction *and*, or replaced the period sign with a comma, while changing the following letter to lower case. We also observed cases where the pronouns *who*, *which* or *where* were employed. Another mode of combining sentences employed by the network was by using participles: ...*she robbed PERSON@2 of her illusions* . *She said* : " *PERSON@3 took ...* " → ...*she robbed PERSON@2 of her illusions* , *saying* : " *PERSON@3 took ...* ". The most interesting case appeared to involve an inclusion: *PERSON@1 saw the affair* . *PERSON@1 was jealous* → *PERSON@1 was jealous of the affair*. In a previous run the same

sentence was embellished using nominalisation: *PERSON@1 saw the affair with jealousy*. For more context, a comparison between an original and a pair-embellished story can be found in the appendix.

4 Conclusion

NLG systems in computational storytellers are commonly dependent on the systems' knowledge base and require idiosyncratic, hand-coded rules if complex language output is desired. Based on this observation we propose to extend the discourse generation pipeline with a TE step that takes as input simple English sentences and performs a monolingual translation into lexically and syntactically more complex English. This approach has the benefit of allowing the authors of storytellers to save time by implementing only simple custom text generation modules, and employ a domain-independent embellishment module to translate into more sophisticated text.

The LSTM Encoder-Decoder approach explored in this paper is not ready for productive use as is, mainly because it does not always fail gracefully, i.e. produces well-formed and semantically appropriate language when embellishment fails. However, it shows promising results by demonstrating that the same trained network is capable of producing interesting lexical as well as syntactical embellishment of varying kind.

We are confident that the presented results can be improved by employing techniques that have proven beneficial in a simplification setting, like e.g. the reinforcement learning proposed by Zhang and Lapata (2017). Due to the nature of the corpus we used for training, only those entries could contribute to the performance of syntactical embellishment that contained two sentences on the simple English, and one on the normal English side. At the same time, the language employed in the Wikipedia can be expected to be markedly different from the one usually aimed for by storytelling systems, which has an impact on lexical embellishment performance. Hence, further improvements can be expected from using more—or better suited—corpora, like e.g. Newsela (Xu et al., 2015). Another promising option would be to explore the viability of embellishment/simplification datasets based on an alignment of simple English versions of novels with their original counterparts. To exceed sentence-

pair based processing, paragraph aligned versions of text would prove to be valuable resources.

Apart from the above theoretical and technological contributions we also presented a first version of Compiled Computer Tales, a corpus of computer generated stories. This corpus can be used to test the capabilities of embellishment algorithms, as well as for historiographic inquiries. We cordially invite collaboration to complete its coverage.

Acknowledgment

The first author is grateful for a Humboldt PhD fellowship.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Charles B. Callaway and James C. Lester. 2002. Narrative prose generation. *Artif. Intell.*, 139(2):213–252.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Pablo Gervás. 2009. Computational approaches to storytelling and creativity. *AI Magazine*, 30(3):49–62.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st ACL*, volume 1, pages 1537–1546.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.
- Sheldon Klein, J. Aeschlimann, and D. Balsiger. 1973. Automatic novel writing: A status report. *Technical Report 186, Computer Science Department, The University of Wisconsin, Madison, Wisconsin*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, and Richard Zens. 2007. MOSES: Open source toolkit for statistical machine translation. In *Proceedings of the 45th ACL*, pages 177–180. Association for Computational Linguistics.
- Michael Lebowitz. 1985. Story-telling as planning and learning. *Poetics*, 14(6):483–502.
- Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4.
- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- James R Meehan. 1977. Tale-spin, an interactive program that writes stories. In *IJCAI*, volume 77, pages 91–98.
- Gustavo H. Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *JAIR*, 60:549–593.
- 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 ACL*, pages 311–318. Association for Computational Linguistics.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th ICML*, pages 1310–1318.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 19th EMNLP*, pages 1532–1543.
- Stefan Riegl and Tony Veale. 2018. Live, die, evaluate, repeat: Do-over simulation in the generation of coherent episodic stories. In *Proceedings of the 9th ICCG*, pages 80–87, Salamanca, Spain. Association for Computational Creativity.
- Matthew Shardlow. 2014. A survey of automated text simplification. *IJACSA*, 4(1):58–70.
- Advait Siddharthan. 2006. Syntactic simplification and text cohesion. *Res. Lang. Comput.*, 4(1):77–109.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Adv Neural Inform Process Syst*, pages 3104–3112.
- Tong Wang, Ping Chen, Kevin Amaral, and Jipeng Qiang. 2016. An experimental study of LSTM encoder-decoder model for text simplification. *arXiv preprint arXiv:1609.03663*.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *TACL*, 3(1):283–297.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 22nd EMNLP*, pages 584–594, Copenhagen, Denmark.

Appendix

Original (Riegl and Veale, 2018) and embellished stories, using the pair-based method:

PERSON@1 needed a place to live and PERSON@2 had plenty of it . PERSON@1 found PERSON@2 at an underground lair . PERSON@1 rented accommodation from her . She paid PERSON@2 what she owed . LOCATION@1 could not achieve bossy PERSON@1 's lofty goals . She refused to honour PERSON@2 's commitments to her , so PERSON@1 ripped off rich PERSON@2 's best ideas . PERSON@1 PERSON@2 evicted PERSON@3 from LOCATION@1 's home . At a smoke-filled back room PERSON@3 met PERSON@4 . LOCATION@1 assiduously curried favor with dictatorial Oscar after cheated PERSON@1 evicted PERSON@2 from LOCATION@2 's home . PERSON@2 told eager Wilde a pack of lies . PERSON@1 said : " Dolores wrote propaganda to promote your cause . " His attitude hardened toward LOCATION@1 . He openly disrespected PERSON@1 because earlier she took everything that PERSON@2 had . PERSON@1 tried to tune out loudmouthed PERSON@3 's voice . LOCATION@1 PERSON@1 wrote PERSON@2 off as a loser , so he coldly dismissed PERSON@2 and turned away . It was at the red carpet when PERSON@2 found LOCATION@2 . PERSON@1 started a new job for influential Rina after unsatisfied PERSON@2 told PERSON@1 to get out and not come back . PERSON@3 took full advantage of her . She pulled the wool over PERSON@1 's eyes . She said : " PERSON@2 was a real suck-up to aristocratic PERSON@3 . " LOCATION@1 could not reach the bar set by bossy LOCATION@2 . She was very disappointed in her , so " Get out ! You 're fired " said PERSON@1 . It was at a recording studio when PERSON@2 found PERSON@3 . LOCATION@1 PERSON@1 re-

cruited PERSON@2 into her ranks after PERSON@3 asked her to clear out her desk and leave . PERSON@2 took the spotlight from lackadaisical Dolores . PERSON@1 withheld due payment from lazy Maura . PERSON@2 criticized sinful Dolores in public . She said : " PERSON@1 showed no shame in sucking up to influential PERSON@2 . " She broke with her and went her own way . What do you think ? Can PERSON@1 and PERSON@2 ever mend their relationship ?

PERSON@1 needed a place to live and PERSON@2 had plenty of it , and PERSON@1 found PERSON@2 at an underground lair . PERSON@1 rented accommodation from her and paid PERSON@2 what she owed . LOCATION@1 could not achieve overbearing PERSON@1 's lofty goals and refused to honor PERSON@2 's commitments to her , so PERSON@1 ripped off rich PERSON@2 's best ideas . PERSON@1 PERSON@2 evicted from LOCATION@1 's home , and at a smoke-filled back room PERSON@3 met PERSON@4 . LOCATION@1 assiduously curried favor with dictatorial Oscar after cheated PERSON@1 evicted PERSON@2 from LOCATION@2 's home , who told eager a pack of lies . PERSON@1 said : " Dolores wrote propaganda to promote your cause " . He openly disrespected PERSON@1 because earlier she took everything that PERSON@2 had , and PERSON@1 tried to tune out loudmouthed PERSON@3 's voice . LOCATION@1 PERSON@1 wrote PERSON@2 off as a loser , so he coldly dismissed PERSON@2 and turned away at the red carpet when PERSON@2 found LOCATION@2 . PERSON@1 started a new job for influential Rina after unsatisfied PERSON@2 told PERSON@1 to get out and not come back . She pulled the wool over PERSON@1 's eyes , saying : " PERSON@2 was a real suck-up to aristocratic PERSON@3 . " LOCATION@1 could not reach the bar set by overbearing LOCATION@2 , and she was very disappointed in her , so " Get out ! You 're fired " said PERSON@1 , at a recording studio when PERSON@2 discovered PERSON@3 . LOCATION@1 PERSON@1 recruited PERSON@2 into her ranks after PERSON@3 asked her to clear out her desk and leave . PERSON@1 withheld due payment from lazy Maura . She said : " PERSON@1 exhibited no shame in digestion up to influential PERSON@2 " she broke with her and went her own way . What do you think ?