AuTS: The Autocompletion for Text Simplification

Anonymous EMNLP submission

Abstract

The goal of text simplification is to transform difficult text into a version that is easier to understand and more broadly accessible. In some domains, such as healthcare and medical, fully automated approaches cannot be used since the information must be accurately preserved. In this paper, we introduce the autocompletion task for text simplification, which aims to assist human simplification by suggesting the next word to type when manually simplifying a text. We compare two pre-trained neural language models (BERT and GPT-2) and show how the additional context of the sentence to be simplified can be incorporated to achieve significantly better results (20.5% and 28.8% absolute improvement, respectively). The best model, GPT-2 with context, achieves a word prediction rate of 52% on Wikipedia data.

1 Introduction

Text simplification (TS) is the process of modifying the words and structure of a text while preserving the content to make the information in the text more broadly accessible (Shardlow, 2014). Most research in text simplification has focused on fully automated (Zhu et al., 2010; Coster and Kauchak, 2011; Xu et al., 2016; Zhang and Lapata, 2017; Nishihara et al., 2019). In some domains, e.g., medicine or healthcare, using fully-automated text simplifications is not appropriate since it is critical that the information gets preserved correctly during the simplification process. Instead of fully-automated approaches, support tools such as editors are better suited to generate simplifications more efficiently and with higher quality (Kloehn et al., 2018).

Autocompletion tools suggest one or more words as the user types that could follow what has been typed so far. Autocompletion has been used in a range of applications including web queries (Cai et al., 2016), database queries (Khoussainova et al., 2010), texting (Dunlop and Crossan, 2000), and e-mail composition (Dai et al., 2019). In this paper, we explore autocompletion for text simplification. In contrast to most autocomplete applications, for text simplification, in addition to the text that is being typed, we also have the additional context of the content being simplified. Our work is most similar to interactive machine translation tools where a user translating a foreign sentence is given guidance as they type (Green et al., 2014).

Difficult sentence	The Chapel is actively used as a place of worship and also for some concerts and college events.
Typed	Concerts and college events ______

Figure 1: An example text simplification autocompletion task. The user is simplifying the difficult sentence on top and has typed the words on the bottom so far.

In this paper, we examine the autocompletion task for sentence-level text simplification: given a difficult sentence that a user is trying to simplify and the simplification typed so far, the goal is to suggest the next word to follow what has been typed. Figure 1 shows an example difficult sentence along with the simplification that the user has typed so far. The task is to predict the next word to assist in finishing the simplification, in this case a verb like "take", which might be continued to a partial simplification of "take place at the Chapel".

We make two main contributions. First, we introduce the autocompletion task for sentence simplification and provide an initial analysis based on a number of recent models. Second, we show how the additional context of the difficult sentence can be integrated into these models to improve the quality of the suggestions made. Using the context of the difficult sentence significantly improves the prediction quality of the autocomplete methods.

2 Text Simplification Autocomplete

Given a difficult sentence that a user is trying to simplify, $d_1 d_2 ... d_m$, and the simplification typed so far, $s_1 s_2 ... s_i$, the autocompletion task is to suggest word s_{i+1}. We examined two recent neural models that utilize the Transformer network (Vaswani et al., 2017): BERT (Devlin et al., 2018) and GPT-2 (Radrof and Wu, 2018). The two models are state-of-the-art neural language representation models that have performed well in a range of applications.

To understand the benefit of the context of the difficult sentence, we compare models that do not use context, i.e., predict only based on $s_1 s_2 ... s_i$, and context-aware versions that incorporate the difficult sentence into the prediction.

2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a method for learning language representations using bidirectional training. The main advantage of BERT is that it uses a masked approach to train the model where some of the words in the training data are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words based on the context provided by the other, non-masked words in the sequence. Unlike left-to-right or right-to-left sequential models, BERT can use context both before and after the word to be predicted. BERT has been shown to produce state-of-the-art results in a wide range of generation and classification applications (Devlin et al., 2018).

We use the original BERT pre-trained model, which was trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. To apply the model without context, we predict the masked word for the input "$s_1 s_2 ... s_i$[MASK].". For the context-aware version, we add the context of the difficult sentence "$d_1 d_2 ... d_m . s_1 s_2 ... s_i$[MASK].". This biases the prediction to words related to those found in the encoded context from difficult sentences.

BERT is a pre-trained model designed to be fine-tuned for particular applications. For the text simplification autocompletion task, we fine-tuned BERT on a corpus of sentence-aligned difficult concatenated with the corresponding simple sentences. We used Transformer Neural Networks to fine-tune the pre-trained BERT language model[1] on this data. Since, our task is to predict the next word in the simple sentence, we mask out each word in the simple sentence portion and then predict that word.

[1] https://github.com/huggingface/transformers/tree/master/examples/

2.2 GPT-2

Like BERT, GPT-2 is also based on the Transformer network, but uses left-to-right training and prediction. In each layer, GPT-2 has 12 independent attention mechanisms, called "heads", and the overall model contains 12 layers which can capture up to 144 different attention patterns. We use the publicly released model[2], which has 1.5B model parameters and is trained on web text. Since GPT-2 is a traditional left-to-right model, for the context unaware version we simply predict s_{i+1} based on $s_1 s_2 ... s_i$. Like BERT, to incorporate the context of the difficult sentence, we prepend it to the simplified text typed so far and then predict s_{i+1}. We did not do any fine-tuning for GPT-2.

3 Experiments

We introduce a new task, text simplification autocomplete, which relies on a parallel corpus of difficult and simple sentences. Given a difficult sentence and the the first i words of the simple sentence, the goal is to predict the $i+1^{th}$ simple word. We provide the first results on this task using BERT and GPT-2, with and without context, as well as an trigram language model baseline.

3.1 Experimental setup

To evaluate the quality of the different models, we used the Simple Wikipedia parallel corpus (Kauchak, 2013), which contains 167K pairs of sentences, with one sentence from English Wikipedia and a corresponding sentence from Simple English Wikipedia. We used 70% of the sentence for training, 15% for development, and 15% for testing.

As an additional baseline that does not use context, we trained a trigram language model with Kneser-Ney smoothing using the SRILM toolkit (Stolcke, 2002). The model was trainined on the simple sentences from the training portion of the dataset and predicts s_{i+1} as the word with the highest probability given the previous two words, i.e., $argmax_{s_{i+1}}\ p(s_{i+1}|s_i s_{i-1})$.

The BERT fine-tuning was done with a batch-size of 8, 8 epochs, and a learning rate of $5e^{-5}$. Early stopping was used based on the second time a decrease in the accuracy was seen.

[2] https://github.com/openai/gpt-2

Difficult sentence	The Saxons built Banbury on the west bank of the River Cherwell.
Simple sentence	Banbury is part of the Cherwell district.

Figure 2: An example sentence pair from the English Wikipedia corpus.

Typed so far	Predict
Banbury	is
Banbury is	part
Banbury is part	of
Banbury is part of	the
Banbury is part of the	Cherwell
Banbury is part of the Cherwell	district

Figure 3: The resulting prediction tasks that are generated from the example in Figure 2.

To evaluate the models, we calculated how well the models predicted the next word in a test sentence, given the previous words. A simple test sentence of length n, $s_1 s_2 ... s_n$, would result in $n-1$ predictions, i.e., predict s_2 given s_1, predict s_3 given $s_1 s_2$, etc. For example, Figure 2 shows a difficult sentence from English Wikipedia and the corresponding simplification from Simple English Wikipedia. Given this test example, we generate six prediction tasks, one for each word in the simple sentence after the first word. Figure 3 shows these six test prediction tasks. For the context-aware approaches, they also incorporated the difficult sentence. We measured the performance of a system using accuracy based on the number of predictions that exactly matched the next word in the corpus. The test corpus contained 25K sentence pairs resulting in a total of 696K individual word predictions.

3.2 Prediction performance

Table 1 shows the results for the five different variants (trigram model, BERT and GPT-2 with and without context). Both neural models significantly outperform the trigram language model; they have been trained on larger corpora and have access to more context allowing for better predictions. Without context, GPT-2 performs slightly better than BERT, with an absolute improvement of 1.7%. To put these accuracy numbers in perspective, in an actual autocomplete task, without context, both BERT and GPT-2 would get about every 4th or 5th word/suggestion correct.

With context, the results improve drastically and

Model	No Context	Context-Aware
trigram	13%	–
BERT	21.5%	42%
GPT-2	23.2%	52%

Table 1: Accuracy for the different models on the Wikipedia test corpus of 25K sentence pairs. Context-aware approaches included the context of the difficult sentence when predicting.

the accuracy rates double for both models. The GPT-2 model benefits the most from the additional information with an absolute improvement of 28.8% over the model without context, resulting in the best performing model with 52% accuracy. On the actual autocompletion task, this equates to predicting every other word correctly. Note that this metric is pessimistic in that the predicted word must match exactly the word seen in the simple sentence and does not account for other possible words that could be correctly used in the context.

Table 2 shows the output of the GPT-2 model with and without context for simplifying the difficult sentence:

> *Each pseudostem can produce a single bunch of bananas.*

The context-aware version is able to take advantage of the strong overlap between the difficult sentence and the simplified version that is being "typed". The model without context makes reasonable predictions grammatically, but without the content priming the suggestions are poor overall.

3.3 Understanding model performance

To better understand how the models are performing and how the predictions of the models differ, we broke down the performances of the neural models by part of speech (POS), difficult sentence length, and the number of words typed so far.

POS Table 3 shows the accuracies broken down by part of speech, where the POS was automatically determined using Stanford CoreNLP (Manning et al., 2014). All of the models perform best on non-content bearing words (i.e., "Other"). Of the content-bearing words, the models did the best on verbs and the worst on adverbs. Overall, GPT-2 with context was the best model at predicting content words with all of the accuracies above 40% except for adverbs, which was 37%.

		GPT-2	
Typed so far	**No Context**	**Context-Aware**	**Actual**
A	particle	**pseudostem**	**pseudostem**
A pseudostem	was	**is**	**is**
A pseudostem is	a	**able**	**able**
A pseudostem is able	**to**	**to**	**to**
A pseudostem is able to	create	**produce**	**produce**
A pseudostem is able to produce	**a**	**a**	**a**
A pseudostem is able to produce a	new	**single**	**single**
A pseudostem is able to produce a single	photon	**bunch**	**bunch**
A pseudostem is able to produce a single bunch	**of**	**of**	**of**
A pseudostem is able to produce a single bunch of	particles	**bananas**	**bananas**

Table 2: Sample output for simplifying the difficult sentence "*Each pseudostem can produce a single bunch of bananas.*" using GPT-2 with and without context. "Actual" indicates the word that should be predicted.

		No Context(%)		**Context-Aware(%)**	
	N	BERT	GPT-2	BERT	GPT-2
All words	696,683	22	23	42	52
Nouns	301,363	13	15	33	43
Verbs	80,180	16	17	39	47
Adverbs	23,600	10	9	25	37
Adjectives	41,780	10	11	30	46
Other	249,760	32	31	71	60

Table 3: Accuracy of the BERT and GPT-2 with and without context by part-of-speech on the test data.

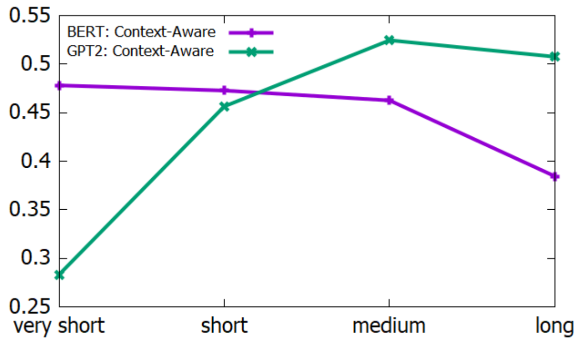

Figure 4: Accuracy for the two context-aware models based on the length of the difficult sentence: very short (≤ 5 tokens), short ($6-15$), medium ($16-19$), and long (≥ 20).

Difficult sentence length Figure 4 shows the performance of the context-aware models based on the length of the difficult sentence. BERT is fairly consistent regardless of the length of the difficult sentence. Only for very long sentences does the performance drop. GPT-2 performs poorly on very short sentences, but well for other lengths. We hypothesize that the training data for GPT-2 (web text) may require more context for the more technical Wikipedia task.

Number of words typed Figure 5 shows the performance of the two context-aware models based on how many words of the simplification the model

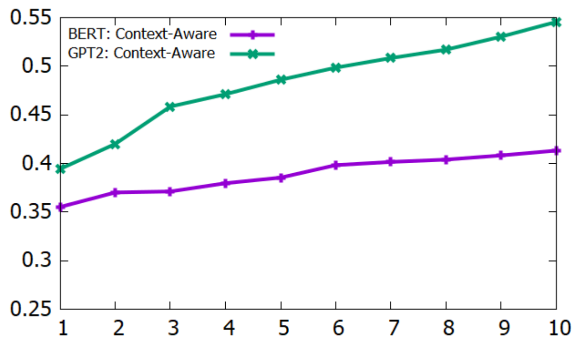

Figure 5: Accuracy for the two context-aware models based on the number of words typed so far (i).

has access to, i.e., i. Early on when the sentence is first being developed, both models struggle. As more and more words are typed and more context is provided, the accuracy of both models increase, however, GPT-2 improves more rapidly as additional context is provided. Like the difficult sentence length analysis, GPT-2 performs better with more context information.

4 Conclusions

In this paper we have introduced a new task, text simplification autocompletion. Unlike most autocomplete tasks, for text simplification, models can be guided by the sentence that the user is simplifying. We compared two recent neural models, BERT and GPT-2, and showed how the difficult sentence could be incorporated into the prediction process. Using context resulted in significant increases in performance with the best model, GPT-2 with context, achieving a prediction accuracy of 52%, getting every other word right. We hope that this new task will allow for other interesting model adaptations to be explored.

References

Fei Cai, Maarten De Rijke, et al. 2016. A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval*.

William Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, pages 1–9. Association for Computational Linguistics.

Andrew Dai, Benjamin Lee, Gagan Bansal, Jackie Tsay, Justin Lu, Mia Chen, Shuyuan Zhang, Tim Sohn, Yinan Wang, Yonghui Wu, et al. 2019. Gmail smart compose: Real-time assisted writing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mark D Dunlop and Andrew Crossan. 2000. Predictive text entry methods for mobile phones. *Personal Technologies*, 4(2-3):134–143.

Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236, Doha, Qatar. Association for Computational Linguistics.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546.

Nodira Khoussainova, YongChul Kwon, Magdalena Balazinska, and Dan Suciu. 2010. Snipsuggest: Context-aware autocompletion for sql. *Proceedings of the VLDB Endowment*.

Nicholas Kloehn, Gondy Leroy, David Kauchak, Yang Gu, Sonia Colina, Nicole P. Yuan, and Debra Revere. 2018. Improving consumer understanding of medical text: Development and validation of a new subsimplify algorithm to automatically generate term explanations in english and spanish. *Journal of Medical Internet Research (JMIR)*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266.

Alec Radrof and Jeffrey Wu. 2018. language model and unsupervised multitask learning. *OpenAI*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of ICCL*.