AutoSMeT: The Autocompletion Application for Simplifying Medical Text

Anonymous COLING 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 medicine, fully automated approaches cannot be used since information must be accurately preserved. In this paper, we introduce a first-of-its-kind medical data set that pairs English Wikipedia with Simple English Wikipedia and the application of pretrained language models (LMs) in autocompletion for text simplification in medical domain. Our autocompletion model aims to assist human simplification by suggesting the next word to type when manually simplifying a text. We compare four pre-trained neural LMs (PNLMs) (BERT, RoBERTa, XLNet, and GPT-2) and show how the additional context of the sentence to be simplified can be incorporated to achieve significantly better results (in the range of 9.0% to 28.8% absolute improvement). The best model, RoBERTa with context, achieves a word prediction rate of 62.4% on medical Wikipedia data. With the conducted LM comparison, we introduce the AutoASMet ensemble model that combines the adventanges of four PNLMs and outperforms RoBERTa by 2.1%.

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 of the researches in text simplification have focused on fully automated (Zhu et al., 2010; Coster and Kauchak, 2011; Xu et al., 2016; Zhang and Lapata, 2017; Nishihara et al., 2019). However, in some domains, e.g., healthcare and medicine, using fully-automated text simplifications is not appropriate because it is critically required that information is preserved fully and correctly during the simplification process. (Shardlow and Nawaz, 2019) shows that fully-automated text simplification models only simplify 5.8% of clinical sentences while preserving critical information. These models tend to obmit information in clinical text, which is critical to both doctors and patients (approximately 30% showed by (Shardlow and Nawaz, 2019)). Therefore, instead of fully-automated approaches, support tools such as editors are better suited to generate simplifications with higher efficiently and quality (Kloehn et al., 2018).

In this paper, we explore the application of PNLMs to autocompletion models for sentence-level medical text simplification. Given a difficult sentence that a user is trying to simplify and the simplification typed so far, the goal is to correctly suggest the next word to follow what has been typed. Table 1 shows an example of a difficult sentence along with a simplification that the user has typed so far. The autocompletion models will 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". In contrast to most autocomplete applications, in addition to the text that is being typed, our models for text simplification benefit from additional context of the content being simplified. This unique characterisitic allows autocomplete models to efficiently simplify medical text with high quality while correctly preserving crucial information, which cannot be found in most fully-automated models. The contribution of our work are four-fold:

1. We introduce a first-of-its-kind medical data set that pairs English Wikipedia and Simple Wikipedia, which is automatically extracted from the Simple Wikipedia parallel corpus (Kauchak, 2013). The resulting medical corpus has 3.3k sentence pairs, of which an estimated 2.8k are genuinely medical.

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 ____

Table 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.

2. We also examine the PNLMs on autocompletion task for sentence simplification and provide an initial analysis based on a number of recent models. We show that the additional context of the difficult sentence can be integrated into these models to improve the quality of the suggestions made. RoBERTa is the best individual model with 62.4% accuracy (19.4% above our base-line).
3. We introduce the AuTS, the ensemble model that combines adventages of recent PNLMs. Our model outperforms the best single PNLM, RoBERTa, by 2.1%. Further, to our best knowledge, this ensembling approach is novel and suggests a potential improvements on PNLMs for natural language processing (NLP) downstream tasks.
4. Finally, we publish an interface of our autocomplete models, which can be used to simplify medical text in real time.

2 Related Work

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). 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).

3 Approach

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 goal of autocompletion model is to suggest word s_{i+1}. To evaluate the quality of the different models, we used the first-of-its-kind medical corpus (see section 4) that we extracted from the Simple Wikipedia parallel corpus (Kauchak, 2013), which contains 167K pairs of sentences. Each pair consists of one sentence from English Wikipedia and a corresponding sentence from Simple English Wikipedia. We used 70% of the sentence pairs for training, 15% for development, and 15% for testing.

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, Tabel 7 shows a difficult sentence from English Wikipedia and the corresponding simplification from the medical Simple English Wikipedia. Given this test example, we generate 19 TODO: AHMAD: please fix the prediction tasks table and add the final number here prediction tasks, one for each word in the simple sentence after the first word. Table 3 shows these six test prediction tasks. For the context-aware approaches, a coressponding difficult sentence is concatenated as a prefix for each prediction task. 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 495 sentence pairs resulting in a total of 7969 individual word predictions.

TODO: AHMAD: after you finished Medical Corpora part, please fix this table with a medical example.

In this work, we examined four recent PNLMs that utilize the Transformer network (Vaswani et al., 2017): BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and GPT-2 (Radrof and Wu, 2018) and the AutoASMeT ensemble models, which combines the advantages of the transformer-based models.

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

Table 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

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

3.1 Transformer-based Language Models

We examined four PNLMs based on Transformers network: BERT, RoBERTa, XLNet, GPT-2. To apply the models to our autocomplete task, we predict the next word for the input "$s_1 s_2 ... s_i$[NEXT].". For the context-aware version, we concatenate the context of the difficult sentence "$d_1 d_2 ... d_m . s_1 s_2 ... s_i$[NEXT].". This biases the prediction to words related to those found in the encoded context from difficult sentences. We also fine-tuned all four models on general parallel English Wikipedia (Kauchak, 2013) and further fine-tuned them on the separate medical training set described in section 4. This two-step fine-tuning helps the models learn the domain knowledge of the text simplification task and the specific language of medical text.

3.1.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a method for learning language representations using bidirectional training.BERT has been shown to produce state-of-the-art results in a wide range of generation and classification applications (Devlin et al., 2018). In this work, we use the base original BERT model pre-trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. We finetuned the pytorch BERT implemented by the huggingface[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.

3.1.2 RoBERTa

In Progress: HOANG

RoBERTa is A Robustly Optimized BERT Pretraining Approach. The RoBERTa uses the same model architecture as BERT. However, the differences between RoBERTa and BERT are that RoBERTa does not use Next Sentence Prediction during pre-trainining and RoBERTa uses larger mini-batch size. We used the publicly released base RoBERTa with 125M parameters model [2]. The RoBERTa fine-tuning was done with a batch-size of 8, 8 epochs, and a learning rate of $5e^{-5}$. Early stopping policy was also similar to BERT fine-tuning.

3.1.3 XLNet

XLNet is a generalized autoregressive pretraining method. Like BERT, XLNet benefits from bidirectional contexts. However, XLNet does not suffer limitations of BERT because of its autoregressive formulation. In this work, we used publicly available base English XLNet with 110M parameters version implemented

[1] https://github.com/huggingface/bert/

[2] https://github.com/huggingface/roberta

Model	No Context	Context-Aware
RoBERTa	56.23%	**62.4**%
BERT	53.28%	50.43%
XLNet	46.2%	45.7%
GPT-2	23.2%	49%

Table 4: Accuracy for the different models on the Wikipedia test corpus of 450 sentence pairs. Context-aware approaches included the context of the difficult sentence when predicting.

in pytorch[3]. The XLNet fine-tuning was done with a batch-size of 8, 8 epochs, and a learning rate of $5e^{-5}$. Early stopping policy was also similar to BERT fine-tuning.

3.1.4 GPT-2

Like BERT, GPT-2 is also based on the Transformer network, however, GPT-2 uses unidirectional left-to-right pretraining process. We use the publicly released model[4], which has 124M parameters and is trained on web text. The GPT-2 fine-tuning was done with a batch-size of 8, 8 epochs, and a learning rate of $5e^{-5}$. Early stopping policy was also similar to BERT fine-tuning.

3.2 Ensembling Models

By combining adventages of four PNLMs, we examined four ensembling approaches and reported their performance in section 5. Our best ensemling model AutoASMeT, which uses neural trained hypothesis selection mechanism, outperforms the best single PNLM by 2.1%.

3.2.1 Majority Vote

As shown in section 11, PNLMs benefit from the increase in number of suggestions. For this ensembling approach, we take the best 5 suggestions from each model and do a majority count in the pool of combined suggestions. The output of the model is the suggestion with most count. If there is a tie, we randomly select one of them. Due to this randomness, we repeat the experiment for 10 times and report the average performance in table 13.

3.2.2 4CC

The 4CC model is an ensembling approach, which we trained a classifier to pick the most approriate model among four PNLMs, given the next-word prediction task. We trained a neural text classification implemented by huggingface[5] with the training set consists sample similar to table 5. Each next-word prediction task is labeled with one of the four options (RoBERTa, BERT, XLNet, GPT-2). This text classification is used as a model selection for our 4CC ensembling model. We designed a scoring system for model selection as follow:

$$Score(w, X) = \alpha * P(w|X) + \theta * I(X, S) \tag{1}$$

In equation 1, $P(w|X)$ is the model X's confidence on predicted word w, $I(X, S)$ is an identity function (which return 1 if $X = S$ and 0 otherwise), S is the predicted model from model selector, α and θ are scoring parmeters. At testing time, we pick the highest score and output the word w, given a prediction task.

3.2.3 AutoASMeT

Because of the strong bias toward RoBERTa in training data for model selection in section 3.2.2, we decide to use a multi-label classifier for model selector in the AutoASMeT. This choice of model selector, to our knowledge, is novel to trasnformer-network-based ensembling models. For this choice of classifier,

[3]https://github.com/huggingface/xlnet
[4]https://github.com/openai/gpt-2
[5]https://github.com/huggingface/transformers

Prediction Task	Class
(Difficult sentence). This (MASK)	RoBERTa
(Difficult sentence). This insulin (MASK)	BERT
(Difficult sentence). This insulin tells (MASK)	XLNet

Table 5: An example of tranining data for the 4CC model. Class can be one of the four option: RoBERTa, BERT, XLNet, GPT-2

Prediction Task	Sequence of Labels
(Difficult sentence). This (MASK)	1 0 1 1
(Difficult sentence). This insulin (MASK)	0 1 0 0
(Difficult sentence). This insulin tells (MASK)	1 1 1 1

Table 6: An example of tranining data for the AutoASMeT model. For a prediction task, a sequence of 4 labels is give in the order "RoBERTa BERT XLNet GPT-2". The value of 1 means the model correctly predict the right word, and 0 otherwise.

each prediction task is given a sequence of 4 labels with value of 0 and 1. Each label represents one of the four PNLMs. Table 6 shows an example of this dataset. We trained a neural multi-label classifier implemented by huggingface [6] on this training dataset and used it as AutoASMeT's model selector. We designed a scoring system for model selection as follow:

$$Score(w, X) = \beta * P(w|X) + \sigma * S(X, Ls) \tag{2}$$

In equation 2, $P(w|X)$ is the model X's confidence on predicted word w, $S(X, Ls)$ is a function (which return 0.25 if model X is in Ls and 0 otherwise), Ls is the predicted sequence of labels from model selector, β and σ are scoring parmeters. At testing time, we pick the highest score and output the word w, given a prediction task.

3.2.4 Upper Bound

To see how well the AutoASMeT model perform, we examine the upper bound, which is the best performanance any ensemble model can achieve. For the upperbound, as long as at least one model among the four PNLMs correctly predicts the next word, given a predicton task, we mark it as correct for the Upper Bound model. This means that no other possible combination of PNLMs can perform any better.

4 Medical Parallel Wikipedia Corpus

In Progress: Ahmad is currently working on this

TODO: AHMAD: can you make this longer and give more details on the corpora creation. Please make sure to include citation from the paper I sent early

5 Results

We provide the first results on this task using BERT, RoBERTa, XLNet, and GPT-2, with and without context, as well as an no-fine-tuned BERT as a baseline. We provide an initial analysis of model performance and use it to design the AuTS, which is an ensemble model which combines advantages of each PNLM.

5.1 Experimental setup

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

[6] https://github.com/huggingface/transformers

Difficult sentence	Lowered glucose levels result both in the reduced release of insulin from the beta cells and in the reverse conversion of glycogen to glucose when glucose levels fall.
Simple sentence	This insulin tells the cells to take up glucose from the blood. The glucose is used by cells for energy

Table 7: An example of sentence pair in Medical Wikipedia parallel corpus.

Domain	No. Sentence Pairs
General Domain	163,700
Medical Domain	3,300
Total	167,000

Table 8: Number of sentence pairs for General Domain and Medical Domain. The two corpora are exclusive.

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., $argmax_{s_{i+1}}\ p(s_{i+1}|s_i s_{i-1})$.

TODO: Add RoBERTa, XLNet, GPT-2 fintuning set up here

5.2 Prediction performance

Table 4 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 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 9 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.

5.3 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, the number of words typed so far. Due to the application of the autocompletion approach to real-time usages, we also provide the accuracy @ N.

Typed so far	GPT-2 No Context	GPT-2 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 9: 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.

	RoBERTa	BERT	XLNet	GPT-2
All words	62.4	50.43	45.7	49
Nouns	60.3	48.7	45.2	51
Verbs	64	50.7	46.2	54
Adverbs	59.1	39.3	45.1	49
Adjectives	55	35.2	34.7	49
DET	76.3	68.7	51.2	51
PropNoun	25.8	21.8	17.9	34

Table 10: Accuracy of the RoBERTa, BERT, XLNet, and GPT-2 with and without context by part-of-speech on the test data.

POS Table 10 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%.

Difficult sentence length Figure 1 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 2 shows the performance of the two context-aware models based on how many words of the simplification the model 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.

Accuracy@N Table 11 shows the accuracy@N from RoBERTa, BERT, XLNet models on next word prediction. Accuracy@N is a metric that gives a model credit as long as it can provide accurate prediction within its k suggestions. As we can see from here, this relaxing schema helps the models better assist technician because the user can pick the best word in the list of suggestion and therefore can help speed up the process and improve model performance.

Performance on predicting next K words Table 12 show results from four neural network models in predicting the next k words. To further understand how performance affected by the more words it

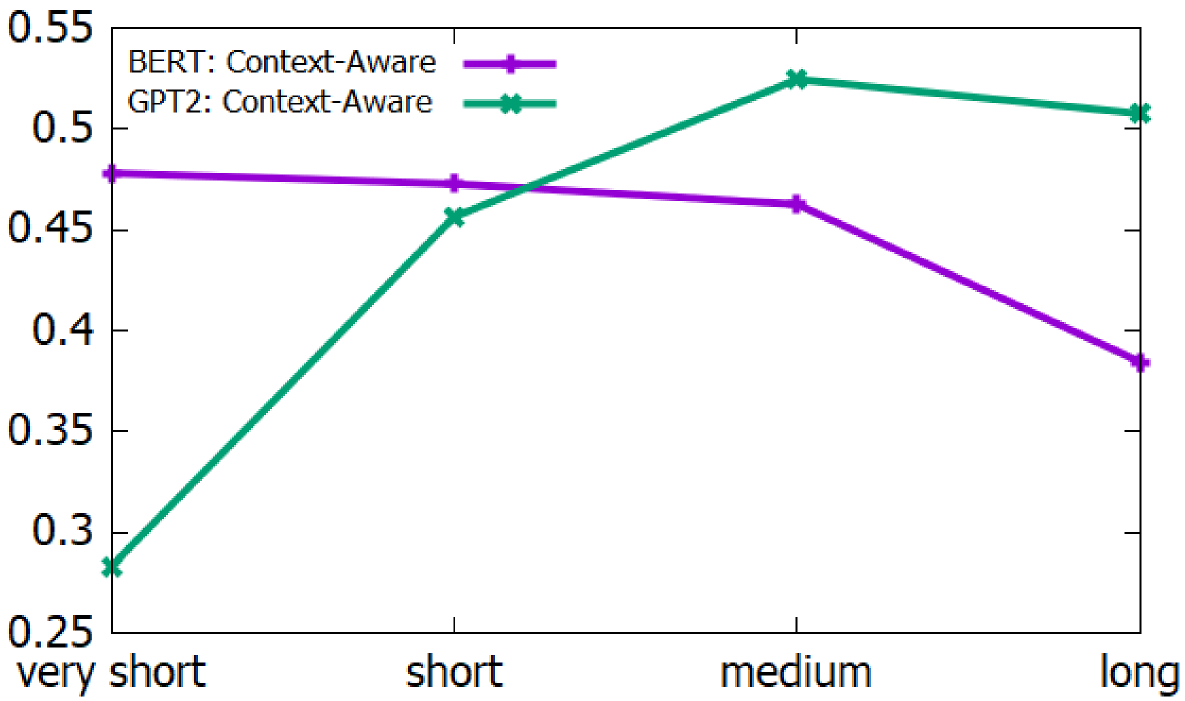

Figure 1: Accuracy for the two context-aware models based on the length of the difficult sentence: very short (≤ 5 tokens), short ($6-15$), medium ($16-19$), and long (≥ 20).

	RoBERTa	BERT	XLNet
accuracy@2	67.2	54.5	46.9
accuracy@3	70	56.2	49.2
accuracy@4	72.1	58.0	51.3
accuracy@5	73.2	59.4	53.5
accuracy@6	73.2	59.4	53.5
accuracy@7	73.2	59.4	53.5

Table 11: Accuracy @ N of the RoBERTa, BERT, and XLNet with context on next word prediction

needs to predict, I run experiments with predicting next 1, 2, 3, and 4 words. This idea is the same with the Google email text suggestion when the model suggest multiple words at a time. As more words need to be predicted, the performance drop significantly for the medical domain. This confirms that the task is unsolved and open for researches to further advance the autocompletion system in medical text simplification.

6 Conclusions

In this paper we have introduced a first-of-its-kind medical corpus for text simplification and proposed new task, text simplification with autocompletion. Unlike most autocomplete tasks, for text simplification, models can be guided by the sentence that the user is simplifying. We compared four recent PNMLs BERT, RoBERTa, XLNet, 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, RoBERTa with context, achieving a prediction accuracy of 62.4%, getting every other word right. With an intial analysis, we designed the AuTS model, which is an ensemble model that combines adventages of PNLMs and outperforms RoBERTa by 2.1%. We hope that this new task will allow for other interesting model adaptations to be explored.

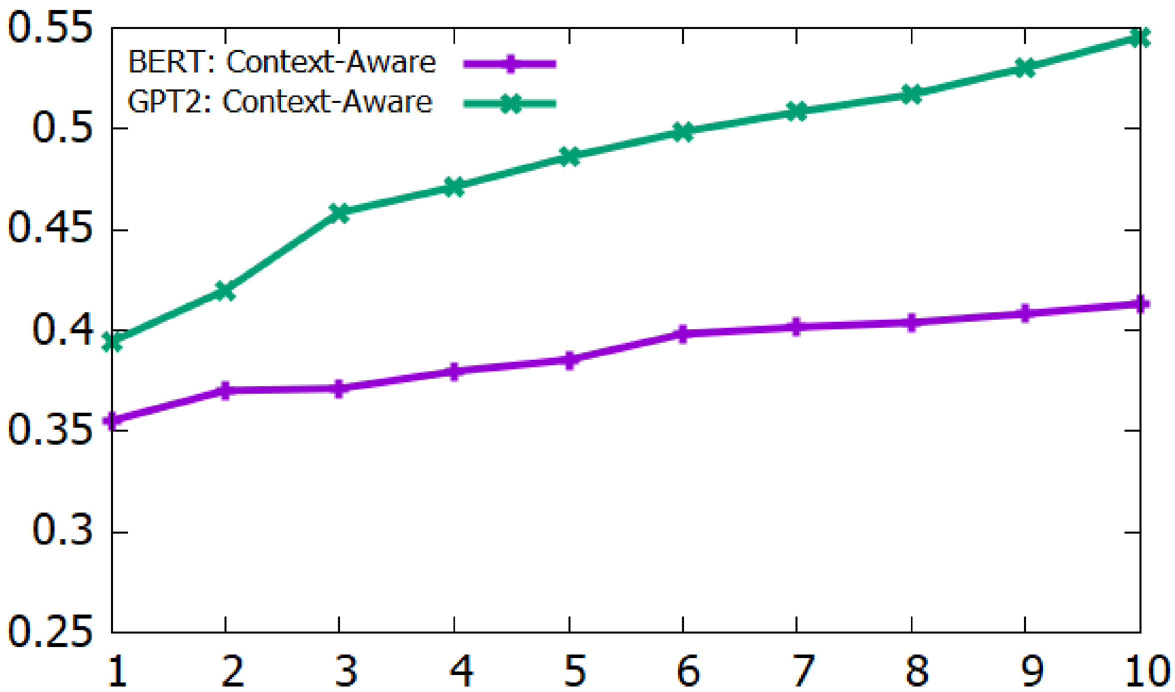

Figure 2: Accuracy for the two context-aware models based on the number of words typed so far (i).

	RoBERTa	BERT	XLNet	GPT-2
Next 1	62.4	53.28	46.2	49
Next 2	45.1	38.7	33.5	41
Next 3	36.8	31.5	26.7	31
Next 4	31.5	24.2	21	14

Table 12: Accuracy of the RoBERTa, BERT, XLNet, and GPT-2 with context on next k prediction

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, October. 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.

Model	Accuracy
Majority Vote	43.25%
4 Class Classification	56.45%
Combined Class Classification	59.3%
AuTS	64.52%
Upperbound	66.44%

Table 13: Accuracy of the majority vote, 4-class classification, combined class classification, AuTS, and the upperbound of ensemble models.

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)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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 and Raheel Nawaz. 2019. Neural text simplification of clinical letters with a domain specific phrase table.

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*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

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

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

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.