ALTER: Auxiliary Text Rewriting Tool for Natural Language Generation

Qiongkai Xu **Chenchen Xu**
The Australian National University
Data61 CSIRO
Qiongkai.Xu@anu.edu.au
Chenchen.Xu@anu.edu.au

Lizhen Qu
Laboratory for Dialogue Research
Monash University
Lizhen.Qu@monash.edu

Abstract

In this paper, we describe *ALTER*, an auxiliary text rewriting tool that facilitates the rewriting process for natural language generation tasks, such as paraphrasing, text simplification, fairness-aware text rewriting, and text style transfer. Our tool is characterized by two features, i) recording of word-level revision histories and ii) flexible auxiliary edit support and feedback to annotators. The text rewriting assist and traceable rewriting history are potentially beneficial to the future research of natural language generation.

1 Introduction

Generative modeling of editing text with respect to control attributes, coined *GMETCA*, has seen increasing progress over the past few years. Such a generative task is referred to as *style transfer*, when the control attributes indicate a change of writing styles (Mir et al., 2019; Fu et al., 2018). This generative task subsumes also gender obfuscation (Reddy and Knight, 2016), authorship obfuscation (Shetty et al., 2018), and text simplification (Xu et al., 2015), when the control attributes indicate protection of gender information, protection of authorship, and simplifying the content and structure of the text, respectively.

The research on *GMETCA* are impeded by the lack of standard evaluation practices (Mir et al., 2019; Tikhonov and Yamshchikov, 2018). Different evaluation methods make system comparison across publications difficult. In light of this, Mir et al. (2019); Fu et al. (2018) proposed both human evaluation and automated methods to judge style transfer models on three aspects: a) style transfer intensity; b) content preservation; c) naturalness. However, it is still difficult to reach an agreement on how to measure to what extent a generated text satisfy all three criterion. Moreover, the lack of human generated gold references hinders the progress of related research, as they i) automate error analysis as in (Li et al., 2018); ii) avoid repeated efforts in user studies to check if system outputs reproduce human-like editing. Therefore, it is beneficial to collect gold references, human edited text, as test corpora for those emerging tasks.

The collection of gold references can be conducted on a crowd-sourcing platform, such as Amazon Mechanical Turk[1], or through existing writing tools (Goldfarb-Tarrant et al., 2019). However, the existing crowd-sourcing platforms and annotation tools do not have the flexibility to add task-specific classifiers and language models, which are widely used for evaluating *GMETCA* models (Mir et al., 2019). As pointed out by Dow et al. (2011), it is important to incorporate task-specific feedback to achieve the improvement of user engagement and quality of results. Feedback is particularly important for *GMETCA* according to our user study (details in Section 4.1), because annotators fail to capture the weak associations between certain textual patterns and attribute values. For example, for gender obfuscation on *'The dessert is yummy !'*, people can easily overlook the implicit indicator *'yummy'* of female authors.

To tackle the aforementioned challenges, we design *ALTER*, an **A**uxi**L**iary **TE**xt **R**ewriting tool, to collect gold references for *GMETCA*. Our tool contains multiple models to provide feedback on rewriting quality and also allows easy incorporation of more task-specific evaluation models. In addition, our tool has a module to record word-level revision histories with edit operations. The revisions are decomposed into a sequence of word-level edit operations, such as insertions (I), deletions (D), and replacements (R), as illus-

[1] https://www.mturk.com/

Proceedings of the 2019 EMNLP and the 9th IJCNLP (System Demonstrations), pages 13–18
Hong Kong, China, November 3 – 7, 2019.

Ori: My husband and I enjoy LA Hilton Hotel.
P_1: Family enjoy LA Hilton Hotel. (Rs)
P_2: Family enjoy Hilton Hotel in LA. (Ro)
P_3: All family members enjoy Hilton Hotel in LA. (I)
P_4: All family members love Hilton Hotel in LA. (Rv)

(a) Revision history 1 (RH1)

Ori: My husband and I enjoy LA Hilton Hotel.
P_1: My husband and I love LA Hilton Hotel. (Rv)
P_2: My husband and I love Hilton Hotel. (D)
P_3: My husband and I love Hilton Hotel in Los Angeles. (I)
P_4: My husband and I love Hilton Hotel in LA. (Ro)
P_5: Family love Hilton Hotel in LA. (Rs)
P_6: All family members love Hilton Hotel in LA. (I)

(b) Revision history 2 (RH2)

Table 1: Two revision histories, RH1 and RH2, from 'My husband and I enjoy LA Hilton Hotel.' to 'All family members love Hilton Hotel in LA.'. Although the overall transformations of RH1 and RH2 are similar, they follow different revision histories.

trated in Table 1. The benefits of revision histories are three-fold. Firstly, revision histories can provide supervision signals for the generative models, which consider rewriting as applying a sequence of edit operations on text (Li et al., 2018; Guu et al., 2018). Secondly, revision histories can potentially provide deep insights regarding cognitive process and human edit behaviours in varying demographic groups. For example, in Table 1, human writers could prefer replacing the subject (Rs) and the object (Ro) as RH1 than replacing the verb (Rv) as RH2. Statistics on revision histories could provide supporting evidence about related assumptions. Thirdly, there are often multiple gold references for the same text. It is more accessible using revision histories to acquire multiple references than rewriting every reference from scratch. As shown in Table 1, P3, P4 in RH1 and P1, P3, P4 and P6 in RH2 are all valid revisions of the original sentence.

To sum up, our contributions are:

- We implemented a tool *ALTER*, which is capable of providing instant task-specific feedback on rewriting quality for *GMETCA*.
- *ALTER* records revision histories with edit operations, which are useful for comparing and analyzing human edit behaviours.

The code of *ALTER* is publicly available under MIT license at `https://github.com/xuqiongkai/ALTER`. A screencast video demo of our system is provided at Google drive.

2 Related Work

Our work is related to the research on edit history of text and assistant text rewriting.

Document-level edit records were used as data to analyze the evolution of knowledge base (Ferschke et al., 2011; Medelyan et al., 2009) and retrieve sentence paraphrases (Max and Wisniewski, 2010). In contrast, our work focuses on word-level edit operations with order. We believe such paradigm introduces more linguistic features, that will benefit both linguistic and social behavior research. Recently, there has been a series of work on conducting edit operations on text to advance automatic natural language generation (Guu et al., 2018; Li et al., 2018). We believe the real-world human rewriting history collected by our system will strengthen these works.

A writing assistant has been proposed to facilitate users, organizing and revising their document. Zhang et al. (2016) proposed to detect the writers' purpose in the revised sentences. Goldfarb-Tarrant et al. (2019) developed a collaborative human-machine story-writing tool that assists writers with story-line planning and story-detail writing. The assistant and feedback generally improved the user engagement and the quality of generated text in those works.

3 ALTER

In this section, we describe the design of *ALTER*, an auxiliary text rewriting tool that is able to i) provide instant task-specific feedback to encourage user engagement, and ii) trace the word-level revision histories. We demonstrate an example of adapting our system on a *GMETCA* task, namely generating the gender-aware rewritten text, which is i) semantically relevant, ii) grammatically fluent, and iii) gender neutral.

3.1 System Overview

Figure 1 depicts the overall architecture of *ALTER*, which consists of a rewriting module, an administrative module, and multiple machine assistance services. The rewriting module offers annotators a user friendly interface for editing a given sentence with instant feedback. The feedback and revision histories in the interface are provided by the machine assistance services. Moreover, the administrative module provides administrators an interface for user management and assigning target

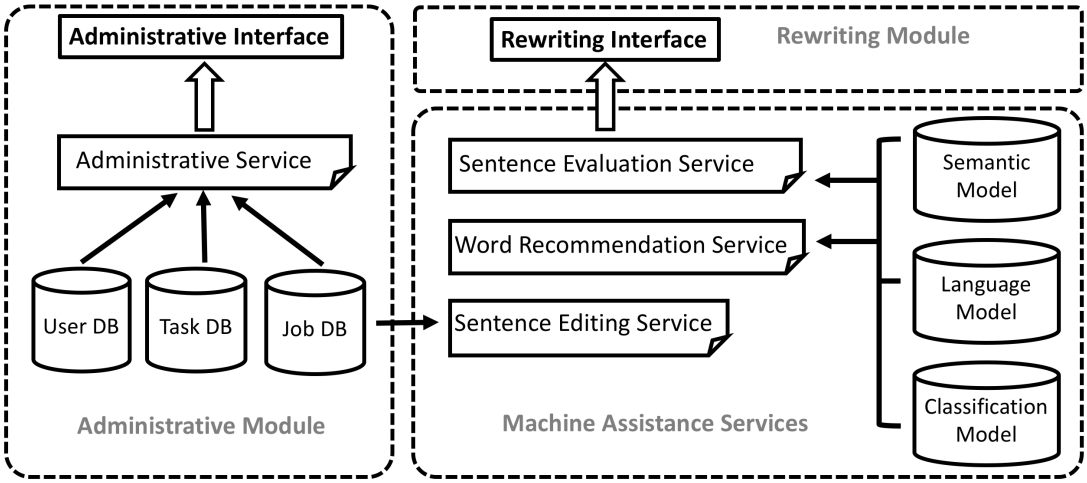

Figure 1: System architecture of the Auxiliary Text Rewriting Tool (ALTER).

tasks, which are basically a set of sentences for rewriting, as *jobs* to individual annotators.

ALTER is based on an easy-to-extend web-based framework that follows the Model-View-Controller (Krasner et al., 1988) software design pattern. The models are the wrappers of the databases (DB). The controller decides what should be displayed on the interfaces, which are considered as the views. This flexible design enables various feedback providers to be easily plugged in and out, making it possible to support different text generation tasks. The front-end is developed with React[2] that enables cross-platform support for major operating systems.

3.2 Rewriting Interface

Figure 2 illustrates a screenshot of the annotator interface. In the left column, there is a list of jobs, which are the sentences assigned to the annotator. The completed jobs are marked in blue. An annotator starts with selecting an incomplete job from the job list, which will be shown in the auxiliary edit panel in the right column. We support two edit modes:

- **Direct typing mode**: Annotators can directly type a whole sentence into the text input field. This mode is provided for the annotators who prefer typing to clicking. To save time, the original sentence is copied to the input field as default value.

- **Auxiliary mode**: Annotators can click on a word shown above the text input field, and choose one of the edit operations from a set, $S = \{$*Word Typing*, *Deletion*, *Substitution*, *Reordering*$\}$. If the annotator chooses *Substitution*, he can select to show a list of words in the gray panel recommended by either word similarity or a pre-trained language model. In this mode, the annotator receives feedback from the upper right corner. Each feedback is a numerical score computed by a feedback provider based on the current sentence. After each edition, a record is added to the revision history below, with the corresponding edit operation and the modified sentences. The annotators are also allowed to roll back the sentences to a previous status by clicking the corresponding record in a history.

3.3 Machine Assistance Services

The machine assistance services in our system include feedback providers and word recommendation services. The machine assistance services can be categorized as sentence-level and word-level.

At the sentence-level, we provide automatic sentence evaluation scores as feedback. In our current system, we consider evaluation metrics widely used in style transfer and obfuscation of demographic attributes (Mir et al., 2019; Zhao et al., 2018; Fu et al., 2018).

- **PPL**. PPL denotes the perplexity score of the edited sentences based on the language model BERT[3] (Devlin et al., 2019).

- **WMD**. WMD is the word mover distance (Kusner et al., 2015) between the origi-

[2] https://reactjs.org

[3] https://github.com/google-research/bert

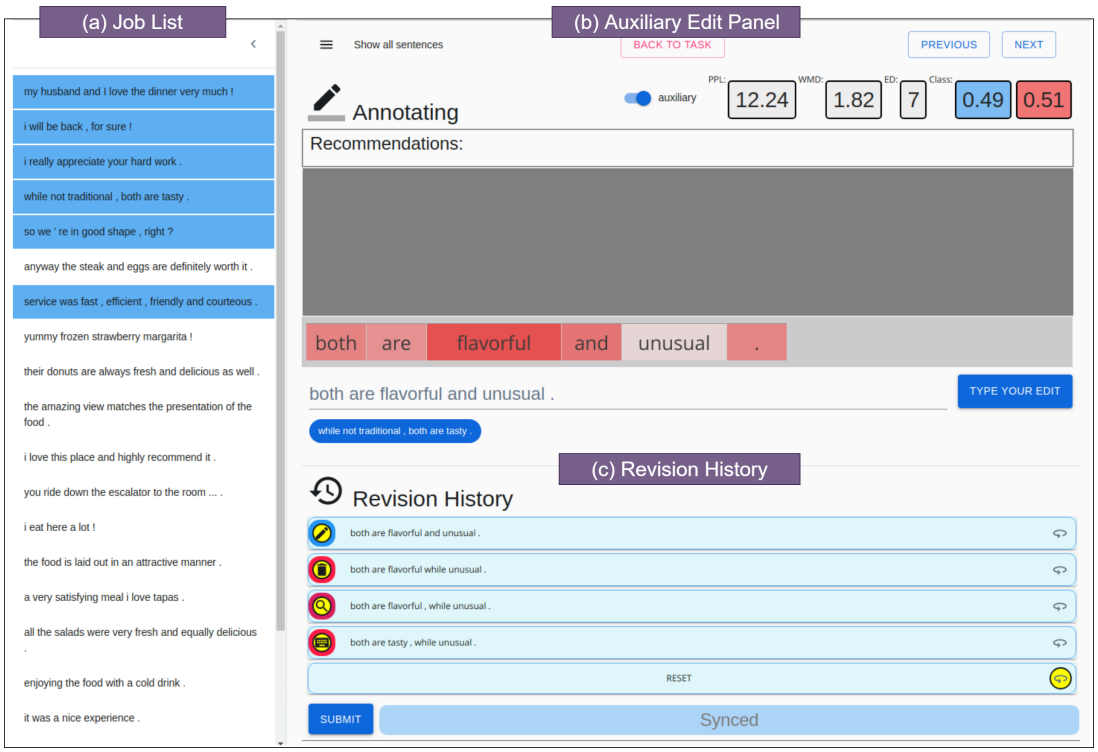

Figure 2: The Auxiliary Text Rewriting Interface is composed of (a) a job list, (b) an auxiliary edit panel and (c) a list of the revision history of current job.

nal sentence and the edited sentence based on Google's pre-trained Word2Vec model[4].

- **ED**. ED denotes the word edit distance between the original sentence and the rewritten sentence.

- **Class**. Class denotes the probability of the attribute value given the edited sentence. It is used to measure style transfer intensity or the degree of obfuscation. In our user study, we employ a transformer-based (Vaswani et al., 2017) binary classifier trained on the **Gender** (Reddy and Knight, 2016) corpus, which contains 2.6M balanced training samples.

At the word-level, we provide two word recommendation services for word substitution, which are based on word embedding similarity and language model, respectively. We include also a word-level feedback provider, which characterizes the contributions of individual words to the sentence-level classification results.

- **Word Similarity Recommendation**. Given a selected word, this service recommends a list of words ranked by the cosine similarity computed based on pre-trained Google word embeddings.

- **Language Model Recommendation**. The services apply a pre-trained language model BERT to the context around the selected word to predict top-k most likely words.

- **Salience**. This module utilizes the sentence classifier trained on the Gender corpus to compute a salience score for each word. A salience score is defined as $S(X, i) = P(Y|X) - P(Y|X \setminus x_i)$, where $P(Y|X)$ denotes the probability of an attribute value Y given the input sentence X, and $X \setminus x_i$ denotes the sentence X excluding the ith word.

4 User Study

We conduct empirical studies to demonstrate i) annotators fail to capture certain textual patterns leading to worse estimation accuracy than the classifier; ii) *ALTER* improves user engagement; iii) machine assistance consistently collects more references per sentence than asking annotators directly typing edited sentences. Both studies are based on the *Gender* (Reddy and Knight, 2016) dataset, which consists of reviews from Yelp annotated with the gender of the authors. In the first

[4] https://code.google.com/archive/p/word2vec

study, we ask annotators to estimate the gender of authors given a sentence. In the second study, We consider a privacy-aware text rewriting task. We ask annotators to rewrite sentences that i) leak less gender information, ii) maximally preserve content; iii) are grammatically fluent.

4.1 Awareness of Gender Information

In the first study, we compare the accuracy of predicting gender information between two human annotators and the classifier[5]. Both of them predict the authors' gender of 300 sentences randomly sampled from the test set. Human annotators obtain merely 66.0 of accuracy on average, while the classifier achieves 77.3. We have carefully investigated the prediction results and the sampled sentences. We found out that it is indeed difficult for humans to estimate correctly the authors' gender based on a short piece of text, e.g.,"the food is delicious" and "the people were nice". Both examples are perceived as neutral for our annotators. Apart from human failure to capture weak associations between certain textual patterns and gender, we conjecture that the bias in the corpus may help the classifier achieve better performance.

4.2 User Engagement

In this study, three graduate students are invited to rewrite 100 sentences randomly selected from the test set of the Gender corpus. All students take two steps to rewrite each sentence:

1. In the *direct typing mode*, type the edited sentence directly in the input field .

2. In the *auxiliary mode*, improve the edited sentence from the first step when necessary. The annotators are instructed that i) it is fine to leave the sentences as they are if feedback do not provide useful clues; ii) all feedback and recommendations are machine generated, thus not perfect.

We consider the two-step approach to compare the differences between the two modes while minimizing individual differences between annotators.

We analyze the revision history collected in the second step, and found out that feedback indeed leads to significant improvement of user engagement. In the second step, 89.67% of the sentences were modified in the auxiliary mode. The average number of edit operations in the second step is 4.63, showing the willingness of writers to further edit the text under auxiliary mode. The distribution of edit operations is illustrated in Figure 3, *word typing* and *deletion* are clearly the most popular edit operations. Word recommendation services are also effective, contributing more than 10% of the new edits in the auxiliary mode.

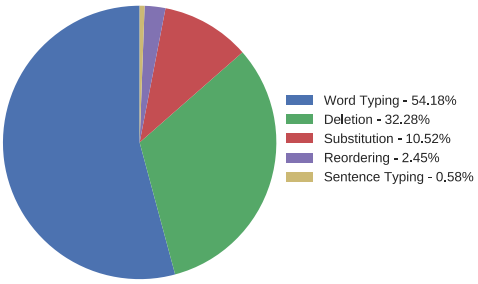

Figure 3: Distribution of operations in revision history by Word Typing, Deletion, Substitution, Reordering and Sentence Typing.

The references collected in the second step result in less leakage of gender information than the ones in the first step. We measure the leakage of gender information by applying the transformer-based classifier on references collected in both steps. We compute averaged *entropy* score, $-\sum_i p_i \log p_i$, based on the predication of each class p_i. Higher *entropy* indicates better obfuscation of gender. The sentences collected in the first step and the second step achieve 0.347 and 0.535 respectively. The entropy of the sentences collected in the first step is just 0.027 better than that of the original sentences.

We further investigate the revision histories, and find more gold references per sentence in the second step than in the first step. We consider semantically relevant and grammatically fluent sentences as valid references. The average number of the valid references generated in auxiliary mode is 3.79, while we can merely obtain one reference per sentence in the direct typing mode.

5 Conclusion and Future Work

In this paper, we demonstrate our auxiliary text rewriting tool *ALTER* to collect gold references for *GMETCA*, assisted with word-level revision histories and task-specific instant feedback. In the future, we will apply *ALTER* to collect high-quality benchmarks for *GMETCA*.

[5]We use a linear SVM model trained on **Gender**.

Acknowledgement

This project is supported by the partnership between ANU and Data61/CSIRO. We also gratefully acknowledge the funding from Data61 scholarship that supports Qiongkai Xu and Chenchen Xu's PhD research.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Steven Dow, Anand Kulkarni, Brie Bunge, Truc Nguyen, Scott Klemmer, and Björn Hartmann. 2011. Shepherding the crowd: managing and providing feedback to crowd workers. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1669–1674. ACM.

Oliver Ferschke, Torsten Zesch, and Iryna Gurevych. 2011. Wikipedia revision toolkit: efficiently accessing wikipedia's edit history. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pages 97–102.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 2019. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 89–97.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450.

Glenn E Krasner, Stephen T Pope, et al. 1988. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3):26–49.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874.

Aurélien Max and Guillaume Wisniewski. 2010. Mining naturally-occurring corrections and paraphrases from wikipedia's revision history. In *LREC*.

Olena Medelyan, David Milne, Catherine Legg, and Ian H Witten. 2009. Mining meaning from wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 495–504.

Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 17–26.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4nt: Author attribute anonymity by adversarial training of neural machine translation. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1633–1650.

Alexey Tikhonov and Ivan P. Yamshchikov. 2018. What is wrong with style transfer for texts? *CoRR*, abs/1808.04365.

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, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Fan Zhang, Rebecca Hwa, Diane Litman, and Homa B Hashemi. 2016. Argrewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 37–41.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *International Conference on Machine Learning*, pages 5897–5906.