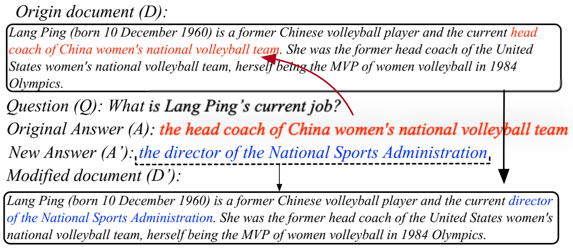
Controlling Text Edition by Changing Answers of Specific Questions

Lei Sha, Patrick Hohenecker, Thomas Lukasiewicz
Department of Computer Science
University of Oxford, United Kingdom
{lei.sha, thomas.lukasiewicz}@cs.ox.ac.uk
patrick@serein.ai

Abstract

In this paper, we introduce the new task of *controllable text edition*, in which we take as input a long text, a question, and a target answer, and the output is a minimally modified text, so that it fits the target answer. This task is very important in many situations, such as changing some conditions, consequences, or properties in a legal document, or changing some key information of an event in a news text. This is very challenging, as it is hard to obtain a parallel corpus for training, and we need to first find all text positions that should be changed and then decide how to change them. We constructed the new dataset WIKIBIOCTE for this task based on the existing dataset WIKIBIO (originally created for table-to-text generation). We use WIKIBIOCTE for training, and manually labeled a test set for testing. We also propose novel evaluation metrics and a novel method for solving the new task. Experimental results on the test set show that our proposed method is a good fit for this novel NLP task.



Origin document (D):

Lang Ping (born 10 December 1960) is a former Chinese volleyball player and the current head coach of China women's national volleyball team. She was the former head coach of the United States women's national volleyball team, herself being the MVP of women volleyball in 1984 Olympics.

Question (Q): What is Lang Ping's current job?

*Original Answer (A): **the head coach of China women's national volleyball team***

New Answer (A'): the director of the National Sports Administration

Modified document (D'):

Lang Ping (born 10 December 1960) is a former Chinese volleyball player and the current director of the National Sports Administration. She was the former head coach of the United States women's national volleyball team, herself being the MVP of women volleyball in 1984 Olympics.

Figure 1: The original text D is in the upper box. The question Q to D has an answer A (in red; its rationale in D also in red). If we would like to change the answer to the new answer A' (in blue), then we have to change some content in D, yielding the modified text D' (with the new content in blue) in the lower box.

1 Introduction

In many cases, we need to change some specific content in a document. For example, in the legal domain, the items and conditions in contract documents often need to be revised many times. We would like to use artificial intelligence to conduct this process for human editors. A major difficulty of this process is that the machine learning model should decide where to edit and how to edit.

Usually, the place of specific content ("where to edit") can be located by a question, and the content updating ("how to edit") can be determined by the answer of the question. Therefore, in this paper, we propose the new task of *controllable text edition (CTE)*. In this task, we would like to achieve the following goal: *adjust some content of a document D, to make the answer A of a document-related question Q changed to a new answer A'*. For example, in Fig. 1, when we change the red part of the original text to the blue part, the answer of the question turned to the new answer as a consequence.

There are three main challenges in this task:

(1) The machine learning model should decide the positions that need to be changed in the document. Usually, finding the answer positions for a given document-related question is similar to extractive machine reading comprehension tasks (Zeng et al., 2020), which requires to fully understand both the question and the document. Nearly all extractive machine reading tasks, such as SQuAD (Rajpurkar et al., 2016, 2018) and CNN/Daily Mail (Hermann et al., 2015), focus on extracting one span from the document as answer. Differently from extractive machine reading, in our task, the answer A is not necessarily a substring of the document, and there may exist multiple positions that have to be changed. Therefore, our task is much more challenging than extractive machine reading.

(2) The model should generate a new document that supports the new answer A' for question Q. Note that this cannot be solved by directly replacing the original words in the edit positions with the new answer A', because the new answer may not fit

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1288–1299
August 1–6, 2021. ©2021 Association for Computational Linguistics

perfectly with the document, which would make the document disfluent.

(3) There are nearly no parallel data for model training, because obtaining a large annotation set for this task is very hard.[1] However, the model may be trained by lists of triples $\langle Q, D, A\rangle$ that can be obtained from datasets in machine reading and/or structured data extraction (as described below).

In this paper, we introduce and define the task of controllable text edition (CTE). We propose to transform the WIKIBIO dataset (Lebret et al., 2016) into a list of triples $\langle Q, D, A\rangle$ for training. WIKIBIO was originally designed for table-to-text generation, in which each case is composed of a Wikipedia passage D and an infobox (which is a list of ⟨field, content⟩[2] pairs). In detail, we take each "field" in the infobox as the question Q, and each "content" in the infobox as the answer A. Therefore, for each ⟨field, content⟩ pair, we can create a $\langle Q, D, A\rangle$ triple. After some pruning, we finally selected 26 different Q's and $141k$ $\langle Q, D, A\rangle$ triples for the training set, as well as $17.7k$ triples for the development set. We also annotated a small test set of about $1k$ data for evaluation in the form of $\langle Q, D, A, A', D'\rangle$ (A' represents the new answer, and D' represents the ground-truth modified text). The resulting new dataset is called WIKIBIOCTE.

In addition, we propose a novel method, called *Select-Mask-Generate (SMG)*, to solve the proposed CTE task. In this method, we use the selector-predictor architecture by Sha et al. (2021) to select the answer-related tokens, and we then use complementary masks to split the text into an answer-related part and an answer-unrelated part. Then, we reconstruct the original text based on the answer-unrelated part and the original answer. The reconstruction process is a partial generation method, which only generates the masked-out part without any length limit. In our experiments, the SMG model has achieved the state-of-the-art performance, compared to baseline models in the generation of modified documents. The code and the test set WIKIBIOCTE are available at: `https://sites.google.com/view/control-text-edition/home`.

2 Related Work

The proposed task of *controllable text edition* is related to the following existing tasks.

2.1 Attribute Disentanglement

Attribute disentanglement tends to control the attributes of a given text or image (such as sentiment, tense, syntax, or face pose) by disentangling different attributes into different subspaces. When transferring attributes, the content of the text/image needs to be preserved. Usually, disentanglement works can be divided into implicit and explicit disentanglement. Implicit disentanglement (Higgins et al., 2017; Chen et al., 2018; Moyer et al., 2018; Mathieu et al., 2018; Kim and Mnih, 2018) separates the latent space into several components in a purely unsupervised way, expecting that each component corresponds to an attribute. However, the number of components cannot be decided in advance, neither does the correspondence between attributes and components. Also, the training process may prune some of the components (Stühmer et al., 2019), which will hurt the interpretability of the latent space. Explicit disentanglement (Chen et al., 2016; John et al., 2019; Romanov et al., 2019; Sha and Lukasiewicz, 2021) tends to separate the latent space into more interpretable components with explicit correspondence to specific attributes. Hence, it usually requires gold labels of attributes in the training set.

In comparison, our task tends to control the content of the text by tuning answers to text-related questions. Attribute disentanglement is difficult to be applied to our task, because the modification of the content should be decided by both the question and the answer simultaneously, which is much sparser than attributes.

2.2 Lexically Constrained Decoding

Lexically constrained decoding (Hokamp and Liu, 2017; Miao et al., 2019; Sha, 2020) directly controls the output of the generation model by adding constraints. Usually, the constraints include hard constraints (requiring the generated sequence to contain some keywords) and soft constraints (requiring the generated sentence to have the same meaning to a given text). The basic methods of lexically constrained decoding can be divided into enhanced beam search (Hokamp and Liu, 2017; Post and Vilar, 2018) and stochastic search (Miao et al., 2019; Liu et al., 2020; Sha, 2020). Enhanced

[1] To annotate a large parallel dataset, we need to prepare a document, a document-related question, and its expected answer. Then, the data grader should provide an adjusted version of the document that satisfies the expected answer, which requires the data grader to have a high education level.

[2] In the Wikipedia Infobox, "field" represents the type of information (such as *Name*, *BirthDate*, and *Known for*), while "content" represents the value of "field".

Table:

ID	Field	Content
1	Name	*Frank Fenner*
2	Born	*21 December 1914, Ballarat*
3	Died	*22 November 2010 (aged 95) Canberra*
4	Occupation	*Virology*
5	Nationality	*Australian*
6	Known for	*Eradication of smallpox, Control of Australia's rabbit plague*

Text: Frank John Fenner (21 December 1914 – 22 November 2010) was an Australian scientist with a distinguished career in the field of virology. His two greatest achievements are cited as overseeing the eradication of smallpox, and the control of Australia's rabbit plague by introducing the Myxoma virus.

Table 1: An example of a Wikipedia infobox and a reference text.

beam search (Hokamp and Liu, 2017; Hasler et al., 2018; Hu et al., 2019) changes some strategies in beam search to make the process of searching for a constraint-satisfying sentence easier. However, for some tasks with an extremely large search space, beam-search-based methods may be computationally too costly or even fail (Miao et al., 2019). Stochastic search tends to edit an initial sentence step-by-step, where the editing position and action can be decided by Metropolis-Hastings sampling (Miao et al., 2019), a discrete scoring function (Liu et al., 2020), or gradient-based methods (Sha, 2020). However, lexically constrained decoding is hard to be applied to our task, because adjusting the text to fit a text-related question's new answer is much more complicated than simply satisfying a hard or soft constraint.

2.3 Text Editing and Infilling

In some tasks, to simplify the text generation problem, researchers tend to edit existing text or prototypes to obtain a refined text that satisfies some specific requirements. Examples are the generation of summaries by template-based rewriting (Cao et al., 2018; Hashimoto et al., 2018) and the generation of text or a response by editing a prototype sentence (Guu et al., 2018; Pandey et al., 2018; Wu et al., 2019). In (Yin et al., 2018), the distributed representations of edit actions are learned and applied to editing Wikipedia records (Faruqui et al., 2018) and Github code (Yin et al., 2018). Panthaplackel et al. (2020) further integrate a copy mechanism into text editing.

Text infilling (Fedus et al., 2018) means to use machine learning models to fill the blanks of a cloze test. Zhu et al. (2019) propose a more general text infilling task, which allows an arbitrary number of tokens (instead of a single token) in each blank.

In the above text editing tasks, the goal of editing is always consistent among all the datasets: for a better summarization, a better response, or a better informative sentence. Differently from them, our proposed task requires the editing to be guided by the document-related answer of the question. So, each above case has a different editing goal. Thus, our task requires deciding where to edit according to the given question in the first step, and then deciding how to edit, which makes our task more complicated than all the above text editing tasks.

3 Dataset

We now formally define the task of controllable text edition and propose a dataset for this task.

3.1 Task Definition

The task of *controllable text edition (CTE)* is defined as follows. The input is a triple $\langle D, Q, A' \rangle$, where D is a document, Q is a document-related question, and A' is an expected answer for Q to D. The output is D', which is a minimal modification of D such that the answer for Q to D' is now A'. Note that the original answer of Q to D is A, but A is not an input to the task, and usually $A \neq A'$.

3.2 WikiBio as Controllable Text Editing Dataset

We propose to modify the WikiBio dataset (Lebret et al., 2016) to make it fit for our task. WikiBio was originally designed for table-to-text generation (Lebret et al., 2016; Sha et al., 2018; Liu et al., 2018), which generates a celebrity's biography according to his/her basic information. Each example in the dataset is composed of a Wikipedia infobox and a text (the first paragraph in the Wiki page) describing the infobox as shown in Table 1.

In an inverse way, the WikiBio dataset can be taken as a question-answering dataset: each *field* can be taken as a question, and each *content* can be taken as an answer. For example, in Fig. 1, the *field* "Occupation" can be interpreted as question "What is the person's occupation?", and the corresponding *content* "Virology" is the answer.

Therefore, we take the *text* in WikiBio as the document (D) in our task, the *field* as the question (Q), and the *content* as the answer (A). Due to the huge cost of data annotation, the model needs to be trained without the changed answer (A') and the referenced document (D').

For the creation of the training and development sets, we count the frequency of *fields* and select the *fields* that occurred more than 5k times in WIKIBIO's training set as candidate questions (Q's). Then, we filter out some Q's that do not have corresponding answers in D[3]. We then get a list of 26 different Q's as shown in Table 2. After filtering the Q's according to Table 2, we get $141k$ $\langle Q, D, A \rangle$ triples for the training set and $17.7k$ triples for the development set.

Then, we manually labeled a small test set in which each example contains (D, Q, A) as well as the changed answer (A') and the referenced document (D'). The annotation process can be illustrated as follows:

1. We randomly sampled an equal number of examples for all the *fields* in Table 2. For each *field*, we sample $\lceil \frac{1000}{\#F} \rceil$ cases ($\#F$ is the number of selected *fields*), to make sure that the size of the test set is around 1k.

2. We assigned a changed answer (A') to each example by randomly picking a similar phrase to the original answer (A). The similar phrase may occur in different examples, but it shares the same Q with the original answer (A).

3. We asked human data graders to give a modified text (D') for each example according to the original text (D), question (Q), and the changed answer (A'). We asked two talented linguistics to annotate the 1k test set.

Note that there are also other datasets that are potentially able to be modified as controllable text editing dataset, such as SQuAD (Rajpurkar et al., 2016), RACE (Lai et al., 2017), and MCTest (Richardson et al., 2013). We did not choose them for the following reasons: (1) For extractive machine reading tasks like SQuAD (Rajpurkar et al., 2016), the answers are simple substrings of the document, so that in most cases, the text modification in our task can be solved by a simple string replacement, which violates the goal of our task. (2) Multiple-choice machine reading tasks like RACE (Lai et al., 2017) usually require full and deep reasoning of the whole document to get the answer, which would make the text modification in our task unable to be solved by partial modification. Differently from them, most *contents* (A) in WIKIBIO usually cannot be directly extracted as substrings from the document (D). Besides, the *contents* usually has some related information that should be modified at the same time. For example, if somebody is a pianist, then he/she may have received a piano award instead of a guitar award. Therefore, WIKIBIO satisfies the goal of our proposed task: making minimal changes to the original document to make it fit the changed answer (A').

birth date	679.4k	name	675.8k	birth place	659.3k
death date	420.7k	death place	377.7k	occupation	231.9k
position	199.4k	nationality	187.1k	spouse	184.0k
fullname	180.2k	alma mater	115.5k	children	114.9k
residence	112.1k	religion	99.3k	predecessor	91.0k
successor	90.1k	known for	63.4k	origin	46.6k
country	43.5k	education	43.1k	instrument	36.7k
college	35.9k	citizenship	29.1k	ethnicity	28.7k
discipline	11.2k	work institutions	5.3k		

Table 2: The selected *fields* from WIKIBIO and their occurrence in WIKIBIO's training set. These are taken as the questions (Q's) in our proposed task.

4 Select-Mask-Generate (SMG) Method for Controllable Text Edition

We introduce the training and testing method of our proposed method. In the training phase, the model is trained to learn to recognize answer-related (A-related) tokens and learn to fill new-answer-related (A'-related) tokens into the blanks after deleting answer-related tokens.

4.1 Training Phase

In the training phase, we only have Q, D, and A. So, we teach the model to (1) identify answer-related information, and (2) be able to reconstruct D from A and $(D - A_p)$ (the original text with all answer-related information masked out, where A_p means the predicted answer-related tokens).

The model architecture is shown in Fig. 2. Inspired by InfoCal (Sha et al., 2021), we use a *Selector-Predictor* architecture to identify the least-but-enough answer-related words in the original document (D). The main architecture of the *Selector* network is a BiLSTM model, which samples[4] a binary-valued mask (M) for each input token (called answer mask), denoting whether to select

[3] Since D is the first paragraph in the Wikipedia page, it usually does not contain everything mentioned in the infobox, such as *death cause* and *high school*.

[4] The sampling process is implemented by Gumbel Softmax (Jang et al., 2016), which is differentiable.

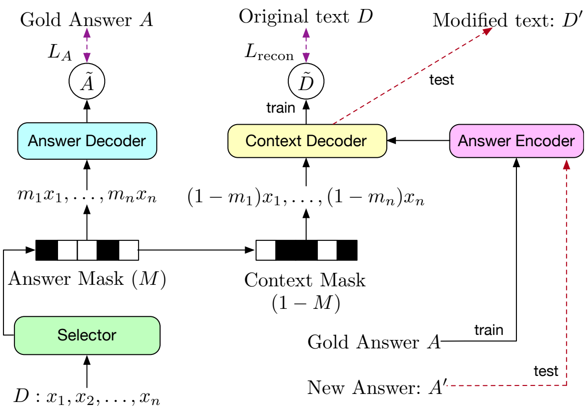


Figure 2: The architecture of our SME model. In the testing phase, we need to replace the input to the answer encoder from the gold answer A to the new answer A', then the output of the context decoder will become the modified text $\tilde{D}'$.

this token as answer-related token (1) or not (0). Given an input document $D = \{x_1, \ldots, x_n\}$ and a question Q, the *Selector* samples an answer-related mask $M = \{m_1, \ldots, m_n\}$ as follows:

$$M \sim \text{Sel}(M|D, Q), \tag{1}$$

where "Sel" represents the selector network. Then, we call the complement of the answer mask ($\overline{M} = 1 - M$) as the context mask, and we denote *context template* as the token sequence after masking out the answer-related tokens.

4.1.1 Answer Reconstruction

We require that the answer-related information contains everything about the answer A, so we use an answer decoder to reconstruct an answer sequence $\tilde{A}$. Then, we calculate the reconstruction loss L_A as follows:

$$p_a(\tilde{A}|M, D) = \text{Dec}_A\Big(\frac{1}{\sum_j m_j} \sum_i m_i x_i\Big), \tag{2}$$

$$L_A = \mathbb{E}_{M \sim \text{Sel}(M|D,Q)} p_a(A|M, D), \tag{3}$$

where Dec_A is the answer decoder, and p_a is the sentence distribution generated by Dec_A. Note that the input to Dec_A is the average vector of the selected token vectors: the answer-related tokens are usually very few, so it is not necessary to use heavier encoders like LSTMs (Hochreiter and Schmidhuber, 1997) or transformers (Vaswani et al., 2017).

4.1.2 Document Reconstruction

On the other hand, D should be reconstructed by the *context template* and the gold answer A. We use an LSTM encoder Enc_D to encode the *context tokens* as shown in Eqs. 4 and 5:

$$h'_1, \ldots, h'_n = \text{Enc}_D([\overline{m}_1 x_1, \ldots, \overline{m}_n x_n]), \tag{4}$$

$$H_m = \text{Maxpooling}(h'_1, \ldots, h'_n), \tag{5}$$

where $h'_1, \ldots, h'_n$ are the encoding vectors corresponding to each input token. We then take the averaged word vector of the input gold answer A, denoted V_A, as an external condition of the decoder.

Differently from conventional decoders, our decoder only partially generates tokens to fill in the blanks of the *context templates*, as shown in Fig. 3. This brings two changes in the training phase: (1) we only need to calculate the loss caused by the tokens filled in the blank, and (2) the model needs to learn an external end-of-answer (EOA) token S_{eoa} for each token filled in the blanks. The EOA token is very important because it is an indicator about when to stop filling the current blank.

Learning to generate the words. In each time step t of the decoder, we use an LSTM (Hochreiter and Schmidhuber, 1997) unit to predict the next word y_t and the EOA token S_{eoa} as follows:

$$h_t = \mathcal{F}_{\text{LSTM}}([y_{t-1}, V_A], h_{t-1}), \tag{6}$$

$$\begin{bmatrix} h_w \\ h_{\text{eoa}} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \end{bmatrix} \mathcal{F}_m(h_t), \tag{7}$$

$$s_t^{\text{lstm}}(w) = \mathcal{F}_w(h_w), \tag{8}$$

$$p(\tilde{S}_{eoa}(t)) = \text{Softmax}\big(\mathcal{F}_{eoa}(h_{\text{eoa}})\big), \tag{9}$$

where h_w and h_{eoa} are hidden layers (the time step index t is omitted), $\mathcal{F}_{\text{LSTM}}$ is an LSTM cell, $\mathcal{F}_m$, $\mathcal{F}_w$, and $\mathcal{F}_{eoa}$ are linear layers, and $s_t^{\text{lstm}}(w)$ is a scoring function that suggests the next word to generate. $p(\tilde{S}_{eoa}(t))$ is the probability distribution of the EOA token.

Note that in the decoder, we use the copy mechanism (Gu et al., 2016), which encourages the decoder to generate words by directly copying from the input context sequence D and answer sequence A. The copy mechanism computes a copy score $s_t^{\text{copy}}(w)$ for each word in D and A. Then, the generated probability of each word is computed as:

$$s_t(w) = s_t^{\text{lstm}}(w) + s_t^{\text{copy}}(w), \tag{10}$$

$$p_t(w) = \text{Softmax}(s_t(w)). \tag{11}$$

Thus, the document D's reconstruction loss is as follows:

$$L_{\text{recon}} = -\mathbb{E}_M\Big[\sum_t m_t \log p_t(y_t|\overline{M}, A)\Big], \tag{12}$$

where $M \sim \text{Sel}(M|D,Q)$, the mask m_t is multiplied in each time step, because we only need the losses of blank-filling tokens.

Learning the end-of-answer (EOA) tags. We have an EOA tag for each blank-filling token. The EOA tag is 1 if the corresponding token is the last token in the blank. For the other blank-filling tokens, the EOA tag is 0. The gold EOA tag in each time step g_t^{eoa} can be computed by the difference between the previous answer mask m_{t-1} and the current answer mask m_t. There are three possible values (-1, 0, and 1): $g_t^{\text{eoa}} = 0$ when the difference is -1 or 0, and $g_t^{\text{eoa}} = 1$ when the difference is 1. Then, we have the cross-entropy loss as Eq. 13:

$$\begin{aligned} g_t^{\text{eoa}} &= \max(m_{t-1} - m_t, 0) \\ L_{\text{eoa}} &= -\mathbb{E}_M \Big[\sum_t \Big(g_t^{\text{eoa}} m_t \log p(S_{eoa}(t) = 1) \\ &\quad + (1 - g_t^{\text{eoa}}) m_t \log p(S_{eoa}(t) = 0) \Big) \Big]. \end{aligned} \tag{13}$$

Therefore, the final optimization objective is shown in Eq. 14:

$$L = L_A + \lambda_r L_{\text{recon}} + \lambda_{\text{eoa}} L_{\text{eoa}}, \tag{14}$$

where λ_r and λ_{eoa} are hyperparameters.

4.2 Inference Phase

In the inference phase, we take the new answer A' as the input to the context decoder instead of the gold answer A. Then, the output of the context decoder will become the modified text $\tilde{D}'$.

We choose an autoregressive partial generation method for inference. Our partial generation method can fill the blanks with any-length phrases and can directly replace any decoder, which cannot be done by any existing alternative methods. For example, in the method using global context (Donahue et al., 2020), it is an pretrained language model by itself. However, in our architecture, the masks are decided by the selector module. Therefore, even the number and length of the blanks cannot be decided before training. So, the ground-truth target sequence for the finetuning of the pretrained language model would also be hard to decide. Therefore, the partial generation method is the best choice for our task.

4.2.1 Partial Generation

Since we already have a *context template* when we are generating the modified document, we only need to generate tokens to fill the blanks in the *context template*. The partial decoding process is shown in Fig. 3. We use an indicator *state*= 0 to denote the *reading* mode (reading the *context template* words), and *state*= 1 to denote the *writing* mode (generating the blank-filling words). The basic generating process is described as follows: when the model is *reading* the *context template*, if it meets a masked token, the mode turns to *writing* mode, and it starts to generate words to fill the current blank. When the EOA tag turns to 1, or the decoding length l_g surpassed a limit $l_{\max}$, the mode turns back to *reading* mode. Note that this decoding process can generate an arbitrary number of words for each blank, and we can fill all blanks in a *context template* in a single decoding pass, which is much more efficient than MaskGAN (Fedus et al., 2018) and text filler (Zhu et al., 2019). The detailed algorithm is shown in Algorithm 1.

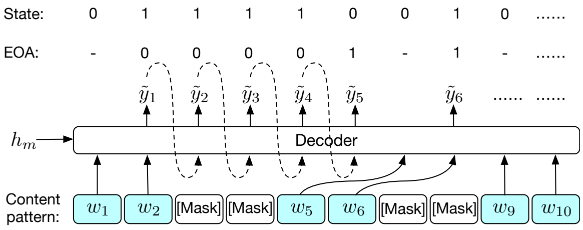

State:
0 1 1 1 1 0 0 1 0 ......
EOA:
- 0 0 0 0 1 - 1 - ......
$\tilde{y}_1$ $\tilde{y}_2$ $\tilde{y}_3$ $\tilde{y}_4$ $\tilde{y}_5$ $\tilde{y}_6$ ....
h_m
Decoder
Content pattern:
w_1 w_2 [Mask] [Mask] w_5 w_6 [Mask] [Mask] w_9 w_{10}

Figure 3: The partial decoding process. This process requires two tags (state and EOA tag) for indicating when to start generation and when to stop generation.

5 Experiments

In the experiment part, we proposed some specific evaluation metric for our *controllable text edition* task and then compare and analysis the performance of our proposed method (SMG) on the WIKIBIOCTE dataset.

5.1 Evaluation Metrics

For the evaluation of the modified document $\tilde{D}'$, we use the following two automatic evaluation metrics:

(1) BLEU ($\tilde{D}'$ vs. D'): This metric measures the BLEU score (Papineni et al., 2002) between the generated modified document $\tilde{D}'$ and the reference document D'.

(2) iBLEU (Sun and Zhou, 2012): This metric is previously widely used in evaluating paraphrase generation tasks (Liu et al., 2020; Sha, 2020). iBLEU is defined as: iBLEU = BLEU($\tilde{D}', D'$) $-$ αBLEU($\tilde{D}', D$)[5], which penalizes the similarity

[5] α is set to 0.9, which is consistent with previous

Algorithm 1: The decoding process.

```
Input: Context template: $C$
Output: Generated Sequence: $\tilde{D}'$
Data: Read-write state: $S$, End-of-answer label:
      $S_{eoa}$, Context template index: $I_c$, Local
      generate length: $l_g$, current input token $x_{in}$
$S \leftarrow 0, I_c \leftarrow 0, l_g \leftarrow 0, \tilde{D}' \leftarrow []$;
Set the first input token $x_{in} \leftarrow C[0]$;
for each time step $t \leftarrow 1, 2, \ldots$ do
    Calculate $\tilde{y}_t$ by Eqn. 11;
    Calculate $S_{eoa}$ by Eqn. 9;
    if $S = 0$ then
        $\tilde{D}' \leftarrow \tilde{D}' + [C[I_c]]$;
        if $C[I_c] \neq$'[M]' and $C[I_c + 1] =$'[M]'
          then
            $I_c \leftarrow I_c + 1$;
            while $C[I_c] =$'[M]' do
                $I_c \leftarrow I_c + 1$;
            end
            if $S_{eoa} \neq 1$ then
                $S \leftarrow 1$;
            end
        end
        else if $C[I_c] \neq$'[M]' and $C[I_c + 1] \neq$'[M]'
          then
            $I_c \leftarrow I_c + 1$;
        end
    end
    else if $S = 1$ then
        $\tilde{D}' \leftarrow \tilde{D}' + [\tilde{y}_t], l_g \leftarrow l_g + 1$;
        if $S_{eoa} = 1$ or $l_g \geq l_{max}$ then
            $S \leftarrow 0, l_g \leftarrow 0$;
        end
    end
    if $S = 0$ then
        $x_{in} \leftarrow C[I_c]$;
    end
    else if $S = 1$ then
        $x_{in} \leftarrow \tilde{y}_t$;
    end
end
return $\tilde{D}'$;
```

between the modified document $\tilde{D}'$ and the original document D. The goal of this metric is to measure the extent to which the model directly copies words from the original document D without taking any content from A'.

	Seq2Seq	SMG (g)	SMG (p)
BLEU ($\tilde{D}$ vs. D)	82.21	**89.29**	87.53
iBLEU	5.63	**10.05**	8.94
diff-BLEU ratio	21.3%	**62.5%**	59.8%
Perplexity	**198**	235	373
Human (Correctness)	73.5%	**80.2%**	76.9%
Human (Fluency)	**4.56**	4.54	4.32

Table 3: The overall performance of all competing methods. SMG (g) denotes that the method SMG is using the gold templates for partial generation, and SMG (p) denotes that the method SMG is using the predicted templates for partial generation.

	Random	Seq2Seq	SMG
BLEU (predicted template)	21.5	59.5	**89.1**
Answer F_1	0.14	0.55	**0.68**

Table 4: Performance of answer-related words selection.

(3) diff-BLEU ratio: diff-BLEU is a BLEU score computed between $\tilde{D}'$ and a *difference sequence* between the gold modified document D' and the original document D. The *difference sequence* is obtained by masking out the longest common sequence between D and D' from D'. Since this maximum value of this BLEU score is the BLEU value between the gold modified document D' and the *difference sequence*, we use their quotient as the diff-BLEU ratio score as shown in Eq. 15:

$$\text{diff-BLEU ratio} = \frac{\text{BLEU}(\tilde{D}', D' - D)}{\text{BLEU}(D', D' - D)}. \quad (15)$$

(4) Perplexity: This metric measures the fluency of the generated content-modified document $\tilde{D}'$. We applied a third-party language model (Kneser-Ney language model (1995)) as the perplexity evaluator. We trained the language model on the whole training set of WikiBio, and use the trained model as the evaluation of fluency, where a lower perplexity value is better.

Besides, we used human effort to evaluate two aspects of the content-modified document $\tilde{D}'$. *Correctness* is an accuracy score from $0.0\% \sim 100.0\%$, which evaluates whether $\tilde{D}'$ has successfully turned the answer of question Q from A to A'. *Fluency* is from $0.0 \sim 5.0$, which evaluates whether $\tilde{D}'$ is fluent from a human being's view. The scoring details are in the supplemental materials.

Also, in our method, the selection of answer-related words is very important, so we have two evaluations for the selection part:

(1) BLEU (predicted template) is the BLEU score between the predicted template (the token sequence after we masked out the answer-related words from the text D) and the gold template (the common sequence of D and D').

(2) Answer F_1 measures the Bag-of-words (BOW) F_1 value of the generated answer $\tilde{A}$ compared to the gold answer A. This metric is difficult to

works (Liu et al., 2020).

Input	D: george evans -lrb- born 13 december 1994 -rrb- is an **english** footballer who plays as a midfielder **or centre-back for** manchester city .	D: andrei UNK -lrb- born 1975 in satu mare , romania -rrb- is a retired romanian **aerobic gymnast** . he had a successful career winning four world championships medals -lrb- two gold , one silver , and one bronze -rrb- after his retirement in 1997 he went with to germany where he works as a gymnastics coach at the UNK gymnastics club in hanover .	D: andrew justin stewart coats -lrb- born 1 february **1958 -rrb- is an australian – british** academic cardiologist who has particular interest in the management of heart failure . his research turned established teaching on its head and promoted exercise training -lrb- rather than bed rest -rrb- as a treatment for chronic heart failure . he was instrumental in describing the “ muscle hypothesis ” of heart failure .
	Q: position	Q: discipline	Q: nationality
	A': halfback quarterback	A': basketball player	A': philippines filipino
Seq2Seq	$\tilde{D}'$: george evans -lrb- born 13 december 1994 -rrb- is an english footballer who plays as a midfielder or centre-back for manchester city . he was a **quarterback halfback** in the manchester .	andrei UNK -lrb- born 1975 in satu mare romania is a retired romanian aerobic gymnast **basketball** he had a successful career winning four world championships medals -lrb- two gold , one silver , and one bronze , after his retirement in 1997 he went with to germany where he works as a gymnastics coach at the UNK **basketball** club	andrew justin stewart coats -lrb- born 1 february 1958 is an **filipino** – british academic cardiologist who has particular interest in the management of heart failure . his research turned established teaching on its head and promoted exercise training -lrb- rather than bed rest -rrb- as a treatment for chronic heart failure . he was instrumental in describing the “ muscle hypothesis ” of **philippines**
With gold template (SMG(g))	$\tilde{D}'$: george evans -lrb- born 13 december 1994 -rrb- is an english footballer who plays as a **halfback and quarterback** for manchester city .	$\tilde{D}'$: andrei UNK -lrb- born 1975 in satu mare , romania -rrb- is a retired romanian **basketball player** . he had a successful career winning four world championships medals -lrb- two gold , one silver , and one bronze -rrb- after his retirement in 1997 he went with to germany where he works as a **basketball coach** at the UNK **basketball club** in hanover .	andrew justin stewart coats -lrb- born 1 february 1958 -rrb- is an **filipino** academic cardiologist who has particular interest in the management of heart failure . his research turned established teaching on its head and promoted exercise training -lrb- rather than bed rest -rrb- as a treatment for chronic heart failure . he was instrumental in describing the “ muscle hypothesis ” of heart failure .
With predicted template (SMG(p))	$\tilde{D}'$: george evans -lrb- born 13 december 1994 -rrb- is an **halfback** footballer who plays as a midfielder **or quarterback** for manchester city .	andrei UNK -lrb- born 1975 in satu mare , romania -rrb- is a retired romanian **basketball player** . he had a successful career winning four world championships medals -lrb- two gold , one silver , and one bronze -rrb- after his retirement in 1997 he went with to germany where he works as a gymnastics coach at the UNK gymnastic club in hanover .	$\tilde{D}'$: andrew justin stewart coats -lrb- born 1 february **philippines** academic cardiologist who has particular interest in the management of heart failure . his research turned established teaching on its head and promoted exercise training -lrb- rather than bed rest -rrb- as a treatment for chronic heart failure . he was instrumental in describing the “ muscle hypothesis ” of heart failure .

Table 5: The example generated cases of competing methods. The underlined tokens are gold answer-related tokens. The bold tokens in the “Input” row are predicted answer-related tokens. In the other three rows, the bold tokens are the modified tokens that are related to the given new answer A'.

achieve, because it requires both to select the correct answer-related tokens and to generate the correct words for the answer A.

5.2 Overall Performance

We compare our method (SMG) with a baseline method (Seq2Seq). In Seq2Seq, the difference with SMG is that the decoder part is a conventional decoder that completely generates the modified document $\tilde{D}'$ ignoring the *context template*. The overall performance is shown in Table 3.

In Table 3, we see that our SMG method has outperformed the Seq2Seq baseline in nearly all evaluation metrics, no matter whether the *context template* applied to the decoding phase is gold or predicted. Especially, in the two most important metrics for the performance of controllable text edition: iBLEU and diff-BLEU ratio, our model has achieved a significantly higher score than competing methods. These results show that our method is effective in controllable text edition.

The human evaluation results are also listed in Table 3. The inter-rater agreements are all acceptable (> 0.85) due to Krippendorff’s principle (2004). According to the human evaluation, when we are using the gold template for partially generating, both the correctness and the fluency of the partially generated text $\tilde{D}'$ are better than using the predicted template, which is also consistent with our intuition. Note that the perplexity score and the fluency score of Seq2Seq are the best of all the three methods; this is because in the partially generated text, the end position of each blank may not fit very well with the next word sometimes, although we have trained an EOA tag.

Table 4 shows the experiments evaluating the selection of answer-related words. We can see that our SMG model has a higher BLEU (predicted template) score than the Seq2Seq model. This fact shows that partially training the blank-filling tokens helps for the selection of answer-related tokens. Also, our model SMG has achieved a higher *answer* F_1 score (0.68) than competing methods.

5.3 Case Study

We have listed some examples of the modified document $\tilde{D}'$ generated by the three competing methods (Seq2Seq, SMG(g), and SMG(p)) in Table 5. We can see that although the answer-related words are already masked out, Seq2Seq still always generates the words in the original answer A and tends to mix up the words in A and the changed answer A' (like in the second example, Seq2Seq mixed “gymnastic” and “basketball” together.) Also, Seq2Seq cannot precisely change everywhere what should be modified, for example, in the second example, Seq2Seq failed to change “gymnastic coach” to “basketball coach”. In the SMG methods, when we are using the gold template for partial generation, the model is able to generate the correct words aiming to change Q's answer to A'. Although there is still some risk to have some answer-related tokens left unchanged due to the error in the predicted template, the *context tokens* in the predicted template are ensured to be generated. Therefore, our model with predicted template is more fit for NLP products than Seq2Seq.

6 Conclusion

In this paper, we proposed a novel task, the goal of which is to modify some content of a given text to make the answer of a text-related question change to a given new answer. This task is very useful in many real-world tasks, like contract editing. We constructed a test set for evaluation and released this test set. We also proposed a novel model SMG to solve this task. In SMG, we first use a selector-predictor structure to select the answer-related tokens in the input document, then we use a novel partial generation technique to generate the modified document without changing answer-unrelated tokens in the original document. The experiments proved the effectiveness of our model.

Acknowledgments

This work was supported by the EPSRC grant "Unlocking the Potential of AI for English Law". We also acknowledge the use of Oxford's Advanced Research Computing (ARC) facility, of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1), and of GPU computing support by Scan Computers International Ltd.

References

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161. Association for Computational Linguistics.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling Language Models to Fill in the Blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315. Association for Computational Linguistics.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better Text Generation via Filling in the _. In *International Conference on Learning Representations*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640. Association for Computational Linguistics.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating Sentences by Editing Prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A Retrieve-and-Edit Framework for Predicting Structured Outputs. In *Advances in Neural Information Processing Systems*, pages 10052–10062.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural Machine Translation Decoding with Terminology Constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *International Conference on Learning Representations*, 2(5):6.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Chris Hokamp and Qun Liu. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546. Association for Computational Linguistics.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved Lexically Constrained Decoding for Translation and Monolingual Rewriting. 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 839–850. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical Reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by Factorising. *arXiv preprint arXiv:1802.05983*.

Reinhard Kneser and Hermann Ney. 1995. Improved Backing-Off for M-gram Language Modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184. IEEE.

Klaus Krippendorff. 2004. Content Analysis: An Introduction to Its Methodology Thousand Oaks. *Calif.: Sage*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-Scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213. Association for Computational Linguistics.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-Text Generation by Structure-Aware Seq2Seq Learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised Paraphrasing by Simulated Annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312. Association for Computational Linguistics.

Emile Mathieu, Tom Rainforth, Siddharth Narayanaswamy, and Yee Whye Teh. 2018. Disentangling Disentanglement in Variational Autoencoders. *arXiv preprint arXiv:1812.02833*.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. CGMH: Constrained Sentence Generation by Metropolis-Hastings Sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. 2018. Invariant Representations without Adversarial Training. In *Advances in Neural Information Processing Systems*, pages 9084–9093.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar Encoder-Decoder for Neural Conversation Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338.

Sheena Panthaplackel, Miltiadis Allamanis, and Marc Brockschmidt. 2020. Copy That! Editing Sequences by Copying Spans. *arXiv preprint arXiv:2006.04771*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation. 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 1314–1324. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A Challenge dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203. Association for Computational Linguistics.

Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2019. Adversarial Decomposition of Text Representation. 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 815–825. Association for Computational Linguistics.

Lei Sha. 2020. Gradient-guided Unsupervised Lexically Constrained Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8692–8703. Association for Computational Linguistics.

Lei Sha, Oana-Maria Camburu, and Thomas Lukasiewicz. 2021. Learning from the Best: Rationalizing Predictions by Adversarial Information Calibration. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, February 2–9, 2021*. AAAI Press.

Lei Sha and Thomas Lukasiewicz. 2021. Multi-type Disentanglement Without Adversarial Training. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-Planning Neural Text Generation From Structured Data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Jan Stühmer, Richard E Turner, and Sebastian Nowozin. 2019. Independent Subspace Analysis for Unsupervised Learning of Disentangled Representations. *arXiv preprint arXiv:1909.05063*.

Hong Sun and Ming Zhou. 2012. Joint Learning of a Dual SMT System for Paraphrase Generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–42. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019. Response Generation by Context-Aware Prototype Editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7281–7288.

Pengcheng Yin, Graham Neubig, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L Gaunt. 2018. Learning to Represent Edits. *arXiv preprint arXiv:1810.13337*.

Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A Survey on Machine Reading Comprehension—Tasks, Evaluation Metrics and Benchmark Datasets. *Applied Sciences*, 10(21):7640.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text Infilling. *arXiv preprint arXiv:1901.00158*.

Appendices

A Human Evaluation Question Marks

Our annotators were asked the following questions, in order to assess the correctness and fluency of the modified document provided by our model.

A.1 Correctness of modified document

Q: Do you think the modification of the document is correct so that it can make the question answer pair $\langle Q, A' \rangle$ true? (For partially correct cases: Partially correct means some places are changed to the new answer, and some places keep the old answer. In this case, only all places (that need to be changed) have been changed can be taken as correct.)

Please choose "Yes" or "No".

After all human annotators finished their work, the correctness score is calculated by dividing the number of "Yes" by the total number of examples.

A.2 Fluency

Q: How fluent do you think the modified document is?

Please choose a score according to the following description. Note that the score is not necessarily an integer, you can give scores like 3.2 or 4.9 , if you deem appropriate.

- 5: Very fluent.
- 4: Highly fluent.
- 3: Partial fluent.
- 2: Very unfluent.
- 1: Nonsense.

B Experiment Details

The word embedding size is 300. The BiLSTM in the selector model has the following hyperparameters: hidden size = 200. The hidden size of decoder's LSTM cell is 200. The rest hyperparameters has the following values: $\lambda_r = 1.0$, $\lambda_{\text{eoa}} = 10$. The hyperparameters are obtained by grid search, the search scopes are $\lambda_r \in [0.0, 2.0]$ with step size

0.2, $\lambda_{\text{eoa}} \in [1, 20]$ with step size 1, the hidden size are searched in $[100, 500]$ with step size 50. The best hyperparameters are selected when the model achieves the highest answer's F_1 in the development set. The total parameter size is $72M$. Each training epoch costs about 1.5 hours on V100 GPU.