electronics

MDPI

Article

Dialogue-Rewriting Model Based on Transformer Pointer Extraction

Chenyang Pu, Zhangjie Sun, Chuan Li and Jianfeng Song *

School of Computer Science and Technology, Xidian University, Xi'an 710071, China; 19040500011@stu.xidian.edu.cn (C.P.); 20031211558@stu.xidian.edu.cn (Z.S.); chuanli@stu.xidian.edu.cn (C.L.)
* Correspondence: jfsong@mail.xidian.edu.cn

Abstract: In the multi-turn dialogue scenario, users commonly encounter challenges with pronoun referents and information omission, leading to semantically incomplete representations. These issues contribute to textual incoherence, as unclear referents and missing components hinder the semantic understanding of the spoken representations of text by machines. Currently, scholars frequently resort to multiple rounds of dialogue rewriting to address the semantic challenges posed by the machine comprehension of semantically missing texts with pronoun referents and information omissions. However, existing dialogue-rewriting methods often suffer from low precision and high latency in handling such texts. To mitigate these shortcomings, this paper proposes a Transformer-based dialogue-rewriting model that utilizes pointer extraction. The method leverages a Transformer pre-training model to effectively extract the potential semantic features of text and extract the key information of text by a pointer address. By extracting keywords and appropriately replacing or inserting text, the model restores referents and missing information. The experimental findings on an open-source Chinese multi-turn dialogue-rewriting dataset demonstrate the effectiveness of the proposed method in improving both the accuracy and efficiency of rewriting compared with existing methods. Specifically, the ROUGR-1 value increased by 2.9%, while the time consumption decreased by 50% compared with the benchmark method.

Keywords: multi-turn dialogue; semantic relevance; dialogue rewriting

check for updates

Citation: Pu, C.; Sun, Z.; Li, C.; Song, J. Dialogue-Rewriting Model Based on Transformer Pointer Extraction. *Electronics* **2024**, *13*, 2362. https://doi.org/10.3390/electronics13122362

Academic Editor: Arkaitz Zubiaga

Received: 16 May 2024
Revised: 11 June 2024
Accepted: 14 June 2024
Published: 17 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Single-round dialogues are often used to address simple problems where there is no transformation between semantic topics. This implies a lack of contextual semantic correlation, absence of personal pronoun references, and no significant omission of key information. Conversely, in multi-turn dialogues, users frequently encounter semantically incomplete representations characterized by pronoun references and information omissions. These deficiencies render the text incoherent, leading to unclear references and missing components, thereby impeding machine comprehension of spoken representations and resulting in poor dialogue quality. Human–machine multi-turn dialogue rewriting refers to the rewriting of the semantically missing text currently inputted by the user in combination with the user's last-round input to produce a coherent, comprehensive semantic text [1].

In the practical context of multi-turn dialogues, it is particularly difficult to deal with incomplete texts within the existing framework. Text incompleteness can be reflected in pronouns and ellipses, and addressing these issues has triggered extensive research and discussion in both academia and industry. This has led to two sub-tasks: denotation disambiguation and ellipsis completion, which aim to rewrite incomplete text with missing semantics into semantically complete and context-independent text. Currently, the mainstream approach in dialogue systems is to enhance the machine's semantic understanding by using contextual dialogue history information through dialogue rewriting, combining the dialogue history of the user and the machine, simplifying the multi-turn dialogue

problem into a single-round dialogue problem, and allowing the dialogue-rewriting model to restore all the pronoun referent information and omitted information by rewriting the semantically missing text currently inputted by the user to help the machine understand the user's real intention [2].

Unlike English and other languages, which generally have a more complete structure, Chinese is able to express more hidden meanings with fewer words through the use of referents and ellipsis. According to Hui et al. [3], the vast majority of conversational communication in Chinese involves omission and reference, for which humans can easily understand the true intention through contextual clues, but it is very difficult for machines.

The conversation-rewriting model based on Transformer pointer extraction designed in this study directly outputs the encoded text vector without using conversation-rewriting technology, which excludes the pointer prediction layer shown in Figure 1. The generated dialogue results are presented in Table 1. The user first enters question 1: "What is C language", which is a complete semantic text, and the machine can directly understand its semantic meaning to reply; then, the user continues to enter question 2: "What are its characteristics", which obviously uses the pronoun "It" to refer to the "C language" in the previous round of text input, which is semantically missing text, and the machine cannot understand the real intention of this semantically missing text with pronouns, and thus, it may not be able to give a reply. The user continues to type in question 3: "What is a computer", which is also a complete semantic text, and the machine can directly understand its semantic meaning to reply; then, the user continues to type in question 4: "Who is the inventor", which obviously requires the semantic meaning of question 4 to be understood according to the abovementioned. The machine cannot understand the real intention of this semantically missing text with missing information, and thus, it may not be able to give a reply.

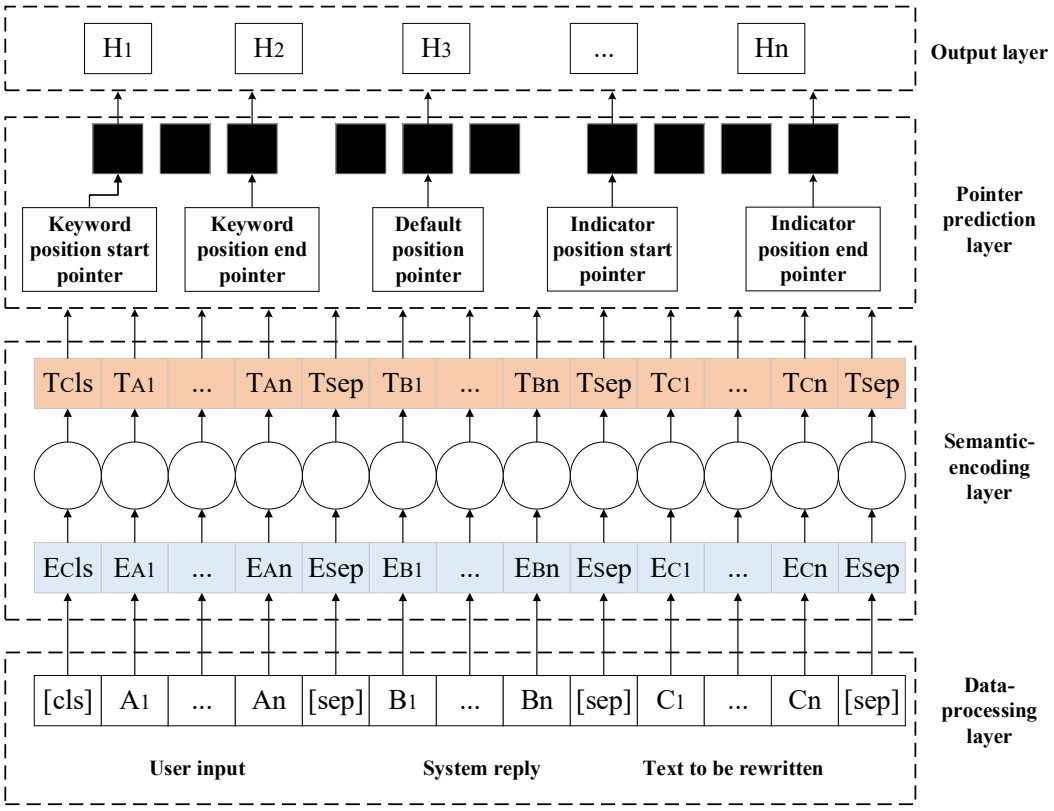

Figure 1. Schematic diagram of the structure of the dialogue-rewriting model.

Table 1. Example of a multi-turn dialogue situation.

Conversation Rounds	Conversation Information
Question 1	What is C language?
Reply 1	C language is a programming language.
Question 2	What are its characteristics?
Reply 2	?
Question 3	What is a computer?
Reply 3	Computers are machines that can perform data operations.
Question 4	Who is the inventor?
Reply 4	?

The above dialogue scenario illustrates that human communication is commonly characterized by semantically incomplete expressions, often involving pronoun referents and missing information. The dialogue rewriting task combines the user's dialogue history with the machine's to rewrite semantically incomplete text, extracting keywords and replacing or inserting the current text to restore the referent or missing information.

In this paper, in order to solve the above problems, a dialogue-rewriting model based on Transformer pointer extraction is proposed. A Transformer pre-training model is used to pay deep attention to the text semantics, effectively extract the potential features of the text, and extract the key information of the text through the pointer address instead of the pointer to rewrite the text, i.e., the key words are extracted and replaced or inserted into the current text to be rewritten in order to recover the pointer or missing information for text rewriting. The specific structure of this paper is as follows:

(1) Introduction: introduces the background and significance of multi-round dialogue-rewriting research and introduces the main research content of this paper and the organizational structure of the article.
(2) Related work: outlines the research work and research methods related to dialogue rewriting and introduces the problems solved by the proposed model.
(3) Model: details the transformer-based model designed by the authors.
(4) Experimentation: gives the dataset and evaluation metrics of the authors' experiments and compares the performance with other models.
(5) Summary: summarizes the scientific results of the whole paper and gives an outlook for subsequent scientific work.

2. Related Work

Conversation-rewriting tasks are text sequence tasks in natural language processing. Before the advent of deep learning, techniques such as grammatical parsing and manual feature engineering were primarily used. However, errors in grammatical parsing were propagated to subsequent models, and manually constructed features lacked flexibility and were difficult to adapt to different languages, resulting in unsatisfactory outcomes. With the rise of deep learning, researchers began designing neural network models for dialogue-rewriting tasks, shifting the focus to leveraging network models for deeper semantic information extraction. However, models like convolutional neural networks (CNNs) have limitations in feature extraction for text sequences and can lose important features. One-way long short-term memory (LSTM) networks focus only on past information in processing long texts, while bi-directional LSTMs (BiLSTMs) capture contextual information but lack strong feature extraction capabilities and an attention mechanism, which are critical for extracting semantic features in dialogue-rewriting tasks.

In the realm of machine translation, Niehues et al. [4] used phrase-based mechanical decoding to pre-convert inputs into the target language, generating final hypotheses based on these pre-conversions. Junczys-Dowmunt et al. [5] explored various neural structures suitable for machine translation results. Gu et al. [6] aimed to enhance the translated text quality by extracting and combining relevant utterance pairs from datasets. See et al. [7] proposed a pointer generation network model crucial for text summarization

that generates new words and copies words from previous texts to rewrite the original text. Chen et al. [8] introduced a model that selects key statements from documents and rewrites them abstractly to form comprehensive summaries. Cao et al. [9] developed a traditional template-based approach by extracting, classifying, and overlaying appropriate text summaries.

In dialogue modeling, Weston et al. [10] focused on rewriting outputs from search paradigms but did not address the restoration of hidden information in referents and omissions. Fields like information retrieval, semantic parsing, and problem-solving predominantly use simplistic lexicons and template-based overlays. However, multi-turn conversations require template-based rewriting due to the complexity of human language, making this approach time-consuming.

Quan J. et al. [11] proposed a new approach for session status tracking and referential expression analysis, solving associative expressions in multi-turn conversations and building referential analysis into user conversational contexts. Wu W. et al. [12] formulated the utterance rewriting task as a span prediction task similar to machine reading comprehension (MRC), generating queries for candidate mentions and using a span prediction module to extract referred text spans. Song et al. [13] introduced a multi-task model MLR for sequence tagging and query rewriting, reformulating multi-turn conversational queries into single-round queries to clearly convey users' true intentions. Xu K. et al. [14] proposed a semantic-role-labeling (SRL) approach to highlight core information for procedural model rewriting. Vakulenko S. et al. [15] developed a Transformer-based question-rewriting model that divides dialogue QA into question rewriting and question replying, redefining the questions according to the conversational context. Therefore, these clear questions can be directly addressed by standard QA components. Chinese scholars Yang et al. [16] combined BERT and pointer networks. The bi-directional modeling function of BERT enables better coverage work and determines whether to rewrite the current session with an incomplete statement. Yang et al. [16] combined BERT and pointer networks, utilizing BERT's bi-directional modeling capabilities for better coverage and determining the need for rewriting incomplete statements in current sessions.

More recently, large language models (LLMs) have been extensively studied for multi-turn dialogue systems. A comprehensive survey by Yi et al. [17] explored the advancements and applications of LLMs in this domain. LLMs, such as GPT-3 and its successors, have demonstrated significant improvements in understanding and generating coherent multi-turn dialogues. These models utilize large-scale pre-training on diverse datasets, enabling them to capture nuanced contextual information and generate more accurate and contextually relevant responses. In the multi-round dialogue scenario, scholars frequently resort to multiple rounds of dialogue rewriting to deal with challenges such as pronoun referents and information omission. However, existing dialogue-rewriting methods often suffer from low accuracy and high latency when handling such texts. Our work builds on these advancements by integrating the strengths of LLMs with pointer extraction mechanisms to enhance the semantic completeness of dialogues, especially in the presence of pronouns and omitted information.

3. Model

For a machine to engage in more intelligent conversations with humans, it must comprehend the semantic content of the user's text, which may be semantically incomplete, within the context of their conversation history. The task of conversation rewriting involves revising the user's currently entered semantically incomplete text in the context of their dialogue history with the machine, thereby restoring coherence. This approach integrates the user–machine dialogue history to address missing semantic elements in the user's input, consequently restoring its intended representation or default information. The predominant strategy in addressing the challenge of semantic comprehension in incomplete text within dialogue systems is through dialogue rewriting. This process enables the dialogue-rewriting model to rectify all pronoun references and omitted information by revising the semantically

incomplete text entered by the user, thereby facilitating the machine's understanding of the user's genuine intent.

To address the problem of low precision and high latency in existing dialogue-rewriting methods, we designed a Transformer pointer-extraction-based dialogue-rewriting model. This model extracts keywords for referent disambiguation and omission completion to restore semantically incomplete text representations for semantic understanding. This paper introduces a dialogue-rewriting model based on Transformer's pointer extraction. The model utilizes Transformer's rbt3 pre-training model to extract and encode text features, identify pointer addresses corresponding to key information, and extract keywords for replacing pronouns or inserting them into default positions, ultimately yielding a complete semantic text post-rewriting. By employing this approach, the constructed dialogue-rewriting model enhances the rewriting accuracy while addressing the slow generation speed inherent in previous text generation methods.

3.1. Model Design

The overall structure of Transformer's pointer extraction-based dialogue-rewriting model is shown in Figure 1. The whole model structure includes four parts: a data-processing layer, semantic-coding layer, pointer prediction layer, and output layer. First, loading and separating each piece of data of the training sample is achieved through the data-processing layer, and the data text, setting head, and join and tail markers are stitched to obtain the complete stitched text. Then, Transformer's rbt3 pre-training model is used to obtain the complete stitched text by feature extraction and dynamic semantic coding. The text vector is then input to the pointer prediction layer to obtain the pointer addresses of the five key pieces of information of the data text, and then the five pointer addresses are input into the output layer to obtain the rewritten text of the network.

(1) Data-processing layer

The data-processing layer is responsible for loading and separating the training samples and splicing the two separated texts, setting head, and join and tail markers to obtain the complete spliced text.

Each training sample comprises four data components: (a, b, current, right), where "a" represents the user's input text from the previous round, "b" denotes the system's reply text from the last round, 'current' signifies the text undergoing rewriting, and "right" denotes the theoretically correct rewritten text. The components are concatenated, with markers placed at the beginning, between connections, and at the end to form the complete concatenated text, denoted as S = ([cls]a, [sep]b, [sep] current, [sep] right[sep]). The first word is the "[cls]" flag, which is used to mark the starting position, while "[sep]" indicates text separation. The output concatenated text S serves as the input for the semantic-encoding layer.

(2) Semantic-encoding layer

The semantic-encoding layer is responsible for encoding the text using Transformer's rbt3 pre-training model to extract features and encode the stitched text S to obtain the text vector; due to Transformer's powerful feature extraction ability and its own multi-head self-attention mechanism, the encoded text vector can focus more on the deeper semantics of the text.

In order to learn the positional and contextual relationships between words in a sentence, the Transformer model needs to additionally add position embedding in addition to the embedding operation of the word itself at the time of input; therefore, for each word vector, its input embedding is the sum of word embedding and position embedding, and the calculation formula is shown in Equation (1):

$$I(w_i) = WE(w_i) + PE(w_i) \quad (1)$$

In Equation (1), $I(w_i)$ denotes input embedding of the word vector w_i, $WE(w_i)$ denotes word embedding, and $PE(w_i)$ denotes position embedding.

The powerful feature extraction capability of Transformer lies in its internally structured multi-headed self-attentive mechanism by mapping the input embedding of each word vector into three new vectors of the same dimension: the query vector, the key vector, and the value vector. The calculation is shown in Equation (2):

$$\text{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \text{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V} \tag{2}$$

In the process of attention computation, query (Q), key (K^T), and value (V) vectors are derived by multiplying the input-embedding vector with three distinct weight matrices. They share the same dimensionality, denoted by d_k in the formula.

Each attention layer of the Transformer is connected to a feedforward neural network that maps the results of the attention layers to a higher dimensional space through a non-linear transformation, which is calculated as shown in Equation (3):

$$\text{FNN}(\mathrm{x}) = \max(0, \mathrm{xW_1} + \mathrm{b_1})\mathrm{W_2} + \mathrm{b_2} \tag{3}$$

In Equation (3), W_1 and W_2 are the parameters to be learned, and b_1 and b_2 are the bias terms. Equation (3) reveals that the Transformer's feedforward neural network undergoes a sequence of transformations. Initially, a linear transformation is applied, followed by the application of a ReLU activation function, and finally, another linear transformation.

To mitigate the issues of gradient vanishing and exploding, Transformer employs residual connections and normalization operations in both the attention and feedforward neural network layers. The specific principle behind this approach ensures that each layer, including the original input and the output from the previous sub-layer, is included. This design safeguards against potential problems that arise from gradient vanishing or exploding during sub-layer computations, ensuring the integrity of the layer's output, as illustrated in Equation (4):

$$\text{Layer} = \text{Layer}(\mathrm{x} + \text{Sublayer}(\mathrm{x})) \tag{4}$$

(3) Pointer prediction layer

The pointer prediction layer is responsible for extracting pointer addresses of key information of the text, including the keyword position start pointer, keyword position end pointer, default position pointer, pronoun position start pointer, and pronoun position end pointer, and by replacing the pronoun of the text to be rewritten with the keyword extracted from the pointer address or inserting the default position, the complete semantic text after rewriting is obtained.

For the extraction of the five pointer addresses of the key information of the text, the difference between the current text to be rewritten and the theoretically correct rewritten text in the data text is compared to derive the keyword, the pronoun, and the default position. The start and end pointers of the keyword position in the last round of user input text are extracted according to the keyword, the start and end pointers of the pronoun position in the current text to be rewritten are extracted according to the pronoun, and the default position pointer in the current text to be rewritten is extracted according to the default position.

In the process of model training, it is necessary to continuously learn the connection between the contexts of key information in the text and predict the keyword position start pointer, keyword position end pointer, default position pointer, pronoun position start

pointer, and pronoun position end pointer. For character i, the position pointer probability distribution of its key information is calculated as shown in Equation (5):

$$\mathrm{P(i)} = \frac{\exp(\mathrm{wx_i} + \mathrm{b})}{\sum_{\mathrm{k}=1}^{\mathrm{L}} \exp(\mathrm{wx_k} + \mathrm{b})} \tag{5}$$

In Equation (5), $P(i)$ denotes the predicted probability of the position pointer for character i with its key information, x_i denotes the output vector corresponding to character i, w is the parameter to be trained and learned, b is the bias value, and L is the total length of the input sequence. Probability distributions for the start and end pointers of keyword positions, the default position pointer, and the start and end pointers of the pronoun positions are computed. Subsequently, by deriving the probability distribution for each position, the predicted position for each component is determined as the one with the maximum value.

The pointer prediction layer has some similarities with position embeddings in large language models (LLMs), but there are significant differences in functionality and implementation between the two. (1) Differences in function and purpose: In LLMs, position embedding introduces sequence information into the Transformer model, allowing it to recognize the positions of various elements in the input sequence and consider the order of words when processing natural language. Our pointer prediction layer, however, not only introduces sequence position information but also identifies and extracts key information positions in the text, such as the positions of keywords and pronouns. This information is used for text rewriting, including predicting the start and end pointers of specific positions for precise replacement or insertion operations. (2) Different implementation methods: Position embedding is usually achieved through predefined fixed position encoding (such as sine and cosine functions) or trainable position vectors. In contrast, the pointer prediction layer in this paper learns the differences between input text and target rewritten text through model learning, thereby predicting the position pointers of key information. These pointers compare the current text to be rewritten with the theoretically correct rewritten text to obtain specific keyword positions, pronoun positions, and default positions. This method requires the model to continuously learn the contextual relationships of key information in the text rather than just sequential positional information. Based on these characteristics, our model can more accurately locate key information positions in the text, better understand and process the semantic information, improve the quality of rewritten text, and demonstrate significant advantages in handling complex text-rewriting tasks.

(4) Output layer

The output layer is responsible for decoding the rewritten text vector for the model to obtain the complete semantic text after rewriting. According to the five predicted pointer addresses of the key information of the text, the keywords of the previous text are first extracted. If the default position pointer exists, the keywords are inserted into the default position of the information in the text to be rewritten. If both the start and end pointers of the pronoun position exist, the keywords are substituted with the start and end positions of the pronoun in the text slated for rewriting. Subsequently, the Transformer decoder is employed to reverse decode the text, resulting in the rewritten text output.

3.2. Model Analysis

The model utilizes Transformer's rbt3 pre-training model to process and extract text features. It extracts the pointer addresses of key information within the text and subsequently identifies key words from these addresses to replace pronouns or insert default positions in the text slated for rewriting. This approach enables the generation of rewritten, fully semantic text, addressing issues of low quality and time consumption associated with rewriting semantically incomplete user text using current technology.

The model design centers on two key aspects: first, the selection of the neural network must account for both the long-term dependency and the text's feature extraction

capability. Consequently, the dialogue-rewriting model presented in this section utilizes Transformer to extract contextual global features from the text. Second, addressing the rewriting of semantically incomplete text involves identifying and replacing key textual information. The current method with better effect is the generation method based on the pointer network, which uses the mechanism of a complete copy, i.e., the idea of generating rewritten text from scratch by copying the user's dialogue history through the pointer address and simplifies the multi-turn dialogue problem into a single-round dialogue problem. However, the method of generating rewritten text from scratch based on pointer networks is time-consuming. The dialogue-rewriting model proposed in this paper adopts a pointer extraction approach, which involves extracting the pointer addresses of key textual information, including the keyword position start pointer, keyword position end pointer, default position pointer, pronoun position start pointer, and pronoun position end pointer, and extracting keywords according to the pointer addresses. This method has a great advantage over the idea of pointer generation in terms of the time taken for the model rewriting.

To train the dialogue-rewriting model based on Transformer's pointer extraction, the training set was initially fed into the model. The relevant text designated for rewriting was concatenated using the data-processing layer to generate the complete concatenated text. Subsequently, the concatenated text underwent semantic encoding through the semantic-coding layer. Transformer's rbt3 pre-training model was employed to extract and encode features from the concatenated text, resulting in a text vector. This vector was then input into the pointer prediction layer to obtain the pointer address corresponding to the key information in the text data. Finally, the pointer address was fed into the output layer to produce the fully rewritten semantic text by the model using the decoder.

4. Experiments

In this section, using the Chinese multi-round dialogue-rewriting dataset open-sourced in the paper by Hui su et al. and selecting the metrics of BLUE, ROUGE, EM, and time consumption, comparative experiments were conducted for the dialogue-rewriting method based on the pointer extraction of Transformer and the results of the experiments were analyzed. The specific structure of this section is as follows:

(1) Datasets: describes the source of the dataset for the controlled experiment and the composition of the data.
(2) Evaluation indicators: introduces the evaluation indicators of the controlled experiment.
(3) Comparison experiment: introduces the characteristics of different comparison models.
(4) Experimental results: introduces the experimental environment and analysis of the experimental results.

4.1. Datasets

The dialogue-rewriting model designed in this study used a dataset from the Chinese multi-turn dialogue-rewriting dataset open-sourced by Hui et al. [3] in their study, which contained a total of 20 K multi-turn conversational data instances. We used this dataset to train the model and measure its effectiveness in rewriting semantically missing texts with pronoun referents and missing information. The percentage of dialogue text with pronoun referents was 46%, the percentage of dialogue text with missing information was 33%, the percentage of dialogue text with complete semantics was 21%, and the average length in characters was 12.5, as shown in Table 2. The dialogue-rewriting model was used to rewrite the semantic missing text in the human–machine multi-turn dialogue scenario for the existence of pronoun referents and missing information about the user, and restore the complete semantic text, which needed to be combined with the dialogue history of the user and the machine; therefore, the data samples were organized and labelled into the (a, b, current, right) format, where "a" indicates the user's last round of input text, "b" indicates the reply text from the previous round of the system. The 'current' indicates the current text to be rewritten, and "right" indicates the text after theoretically correct rewriting.

Table 2. Dataset specifics.

Data Property	Value
Total number of samples	20,000
Pronouns refer to sample size	9200
Information default sample size	6600
Number of complete semantic samples	4200
Average character length	12.5

4.2. Evaluation Indicators

(1) BLUE Index

BLUE metrics can be divided into several evaluation metrics according to the n-gram; the common ones are BLUE-1, BLUE-2, and BLUE-3, where the number indicates the number of consecutive words. BLUE-1 measures word-level accuracy, and higher-order BLUE can measure sentence fluency.

(2) ROUGE Index

The ROUGE-N metric can be used to measure the degree of match between the generated results and the standard results, and ROUGE-N actually calculates the recall rate by splitting the model-generated results and the standard results by n-gram. The experiments in this study employed ROUGE-1, ROUGE-2, and ROUGE-L metrics. ROUGE-1 measures the proportion of overlapping words between the model-generated text and the reference text relative to the total number of words in the reference text. ROUGE-2 measures the proportion of overlapping 2-g between the model-generated text and the reference text relative to the total number of words in the reference text. ROUGE-L measures the proportion of the length of the longest common subsequence between the model-generated text and the reference text relative to the length of the reference text.

(3) EM Index

The EM index represents the exact matching degree by comparing the model-generated text with the standard text one by one for all data samples and counting the number of samples where the model-generated text matches the standard text exactly as a proportion of the number of all data samples.

(4) Time Consumption Index

The time consumption metric is the time consumed by the model to complete the rewriting of all data samples and reflects how quickly the model rewrites the user's semantically missing text.

4.3. Compared Models

In this study, a dialogue-rewriting model based on Transformer pointer extraction, referred to as the Trans-Exa model, was designed, and the comparison models for this experiment were the LSTM-Gen model, Trans-Gen model, RUN-BERT model, and LSTM-Exa model. Among them, the LSTM-Gen model and Trans-Gen model are from the model proposed by Hui et al. [3] for generating rewritten text based on a pointer generation network. The RUN-BERT model is from the model proposed by Liu et al. [18] for complementing incomplete dialogue text using semantic segmentation idea. The experiment utilized the LSTM-Exa model to demonstrate the contextual semantic feature extraction capability of the model. This approach was based on the idea of extracting key information using pointer addresses, as proposed in this paper. The extracted information was directly replaced in the text intended for rewriting. The analysis and comparison of the model constructed using the LSTM network without an attention mechanism were conducted using the methods designed for each model.

The idea of generating rewritten text based on pointer generation networks is to achieve excellent results by modeling the discourse rewriting problem as an extraction gen-

eration problem using pointer networks, where the rewritten discourse is generated based on the attention mechanism to extract contextual features by copying from the conversation history or current discourse. Therefore, in order to demonstrate the impact of the contextual semantic feature extraction ability of the text on the model, the LSTM-Gen model proposed in this paper was used, which uses the LSTM network with a missing attention mechanism to extract the contextual semantic features of the text and generate the text-embedding vector. The Trans-Gen model uses the Transformer network with a self-attentive mechanism to extract contextual semantic features of the text. However, the concept behind generating rewritten text using a pointer generation network relies on exact replication. This approach involves copying the user's conversation history using pointer addresses and then generating rewritten text from scratch. In theory, most of the words in the fully rewritten semantic text originate from the semantically incomplete text intended for rewriting, and the overlapping words theoretically remain unchanged. Therefore, copying the conversation history and then generating rewritten text from scratch inevitably increases the computational burden. Additionally, when working with Transformer, employing six layers for both the encoder and decoder prevents the network from loading certain pre-trained model weights, ultimately leading to increased time consumption during text rewriting.

To reduce the time consumption and enhance the rewriting speed, Qian Liu et al. proposed the RUN-BERT model, which leverages semantic segmentation to supplement incomplete conversational text and extract contextual features using a BERT pre-training model. Unlike generating rewritten text from scratch, the RUN-BERT model treats the rewritten result as a series of edits to the incomplete input rather than complete conversational rewriting. These editing operations occur between contextual word pairs and the incomplete input, resembling semantic segmentation. Given the relevant features of the word pairs, the model predicts the type of editing for each pair simultaneously. The method of editing operations is less intricate compared with generating rewritten utterances from scratch, leading to faster inference compared with the word-by-word decoding method.

In this study, to address the issue of a prolonged rewriting time, we refined the concept of generating rewritten text using a pointer generation network. We abandoned the approach of generating rewritten text from scratch and introduced the concept of extracting key information based on pointer addresses and directly substituting it into the text intended for rewriting. Specifically, by extracting pointer addresses, such as keyword position start and end pointers, we employed Transformer's pre-training model to extract contextual semantic features and generate text-embedding vectors. This approach led to the development of the Trans-Exa model.

4.4. Experimental Results

The model constructed in this experiment had the task of rewriting semantically missing texts with pronoun referents and missing information, restoring the referents and missing information, and generating complete semantic texts. In order to evaluate the quality of the model rewriting, first, 70% of the dataset was randomly selected as the training set and the remaining 30% as the test set. The trained model was used in the test set to obtain the specifics of the model regarding the BLUE-1, BLUE-2, BLUE-3, ROUGE-1, ROUGE-2, ROUGE-L, and EM metrics, and validate them with each other and the comparison model, as shown in Table 3:

The statistical results in Table 3 reveal that the similarity between the rewritten text and the standard text differed between the models. Specifically, the dialogue-rewriting model based on Transformer pointer extraction, as proposed in this paper, and the model that employed the semantic segmentation idea exhibited comparable performance. Notably, both outperformed the method of generating rewritten text using pointer network addresses. These findings underscore the significant impact of contextual semantic feature extraction on the model's performance. However, the accuracy of the rewritten text matching the standard text surpassed that of methods that employed pointer network addresses or the semantic segmentation idea to supplement the full text. This improvement stemmed from

the process of extracting pointer addresses in this study. By comparing the differences between the current text to be rewritten and the theoretically correct rewritten text in the dataset, keywords, pronouns, and default positions were identified. Specifically, the start and end pointers of keyword positions in the user's last input text were extracted based on keywords. Likewise, the start and end pointers of pronoun positions in the current text to be rewritten were determined according to pronouns. Additionally, the start and end pointers were extracted based on the default positions. The successful extraction of default position pointers enhanced the accuracy of the rewritten text generated by this method.

Table 3. Comparison of experimental results across different models on multi-turn dialogue-rewriting task.

Model	BLUE-1	BLUE-2	BLUE-3	ROUGE-1	ROUGE-2	ROUGE-L	EM
LSTM-Gen [3]	73.23	63.12	48.17	74.57	58.62	75.43	52.32
Trans-Gen [3]	78.75	69.16	54.32	77.52	62.63	78.83	56.84
RUN-BERT [18]	80.05	72.35	56.43	81.21	65.13	80.46	65.63
LSTM-Exa	74.15	64.28	49.22	76.34	59.82	76.37	53.29
Trans-Exa (our model)	80.31	71.02	56.82	81.43	65.39	80.69	68.72

In order to check the dependence of the method of extracting key information and replacing text based on the pointer address and the method of generating text based on the pointer network on the training samples, the number of training samples was changed in the experiment, where 15,000, 8000, 2000 and 1000 data samples were randomly selected to rewrite the training, and 2000 samples were selected from the test set for testing the effect. The ROUGE-1 index results of each model were counted to observe the rewriting effect of the model on the text, and the results are shown in Table 4.

Table 4. Changes in ROUGE-1 index with different numbers of training samples.

Model	15,000 Samples	8000 Samples	2000 Samples	1000 Samples
LSTM-Gen	73.26	69.53	38.75	15.32
Trans-Gen	76.32	72.69	39.62	17.62
LSTM-Exa	74.67	72.28	71.21	69.45
Trans-Exa (our model)	81.82	77.12	76.63	74.62

According to the statistics in Table 4, it can be seen that the method of generating text based on the pointer network had a high dependence on the training sample, while the method proposed in this paper of extracting key information and replacing the text based on the pointer address had less dependence on the training sample and could produce excellent results, even on a smaller dataset.

To measure the performance of the model, this experiment selected 3000, 1000, 500, and 100 samples from the test set to test the effect and counted the time consumption of each model to complete the rewriting for different numbers of samples. The results are shown in Table 5.

According to the statistical results in Table 5, it can be seen that the method of generating text based on a pointer network had the most time consumption and the lowest performance. The method of extracting key information and replacing text based on pointer addresses proposed in this paper performed well in terms of time consumption compared with the method of generating text based on a pointer network, and the method of using a pre-trained model was faster compared with the method of building a model using a native network. However, it was slightly inferior to the method that used semantic segmentation ideas to supplement the full text. Upon analysis, this disparity arose from the necessity of

the proposed method to compare differences between data texts in order to obtain pointer addresses for key information, resulting in an increased computational overhead.

Table 5. Working time consumption of each model with different numbers of test samples (unit: s).

Model	3000 Samples	1000 Samples	500 Samples	100 Samples
LSTM-Gen	320	124	68	21
Trans-Gen	102	35	18	3
RUN-BERT	32	10	3	1
LSTM-Exa	160	62	30	6
Trans-Exa (our model)	50	15	6	2

The comparison results above demonstrate that the Transformer pointer extraction-based dialogue-rewriting model proposed in this paper exhibited efficient performance in terms of processing time and achieved outstanding accuracy in text rewriting.

5. Conclusions

This paper introduces a Transformer-based dialogue-rewriting model that utilizes pointer extraction to address the challenge of understanding semantically incomplete text in human–machine dialogues. Particularly in multi-turn conversations, where users employ pronouns and omit information, our model extracts keywords from the text to disambiguate references and complement omissions, thereby restoring the semantic integrity of the text for better comprehension. Unlike traditional methods that rely on pointer networks, our approach leverages a Transformer pre-training model to deeply delve into text semantics. It effectively captures semantic nuances and identifies key textual elements using pointer addresses. By replacing or inserting current text, our model reconstructs text while preserving referential and missing information. Experimental findings on a publicly available Chinese multi-turn dialogue-rewriting dataset demonstrate that our proposed model not only enhanced the accuracy of dialogue rewriting but also addressed the sluggishness observed in previous text generation methods.

In this paper, a dialogue-rewriting model based on transformer pointer extraction is proposed for rewriting semantically missing text, but there is still room for further improvement of the algorithm due to limited human capabilities. First, despite the emergence of recent dialogue-rewriting datasets [19], high-quality datasets specifically for Chinese are still very rare. The lack of high-quality Chinese dialogue datasets is still an urgent problem to be solved. In addition, the performance and convergence of the semantic rewriting model proposed in this paper in specific contexts still need to be strengthened. In future work, not only the existing Chinese datasets can be expanded but also our method can be combined with the Chinese entity recognition methods through entity extraction of the user's unstructured spoken expression text, and then through entity matching to find more accurate answers.

Author Contributions: Conceptualization, C.P.; methodology, C.P.; data curation, Z.S.; writing—original draft, Z.S., C.P., and C.L.; writing—review and editing, C.L.; visualization, Z.S.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by grants from the Continuing Education Teaching Reform Research Program of Xidian University (no. JA2301).

Data Availability Statement: The original contributions presented in this paper are included in the article; further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hao, J.; Song, L.; Wang, L.; Xu, K.; Tu, Z.; Yu, D. Robust Dialogue Utterance Rewriting as Sequence Tagging. U.S. Patent 17/192,260, 8 September 2022.
2. Jiang, W.; Gu, X.; Chen, Y.; Shen, B. DuReSE: Rewriting Incomplete Utterances via Neural Sequence Editing. *Neural Process. Lett.* **2023**, *55*, 8713–8730. [CrossRef]
3. Su, H.; Shen, X.; Zhang, R.; Sun, F.; Hu, P.; Niu, C.; Zhou, J. Improving Multi-turn Dialogue Modelling with Utterance ReWriter. *arXiv* **2019**, arXiv:1906.07004.
4. Niehues, J.; Cho, E.; Ha, T.L.; Waibel, A. Pre-Translation for Neural Machine Translation. *arXiv* **2016**, arXiv:1610.05243.
5. Junczys-Dowmunt, M.; Grundkiewicz, R. An Exploration of Neural Sequence-to-Sequence Architec-tures for Automatic Post-Editing. *arXiv* **2017**, arXiv:1706.04138.
6. Gu, J.; Wang, Y.; Cho, K.; Li, V.O. Search engine guided neural machine translation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LO, USA, 2−7 February 2018; Volume 32, pp. 5133–5140.
7. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. *arXiv* **2017**, arXiv:1704.04368.
8. Chen, Y.C.; Bansal, M. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. *arXiv* **2018**, arXiv:1805.11080.
9. Cao, Z.; Li, W.; Li, S.; Wei, F. Retrieve, rerank and rewrite: Soft template based neural summarization. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volu-me 1: Long Papers), Melbourne, Australia, 15−20 July 2018; pp. 152–161.
10. Weston, J.; Dinan, E.; Miller, A.H. Retrieve and Refine: Improved Sequence Generation Models for Dialogue. *arXiv* **2018**, arXiv:1808.04776.
11. Quan, J.; Xiong, D.; Webber, B.; Hu, C. GECOR: An End-to-End Generative Ellipsis and Co-reference Resolution Model for Task-Orien-ted Dialogue. *arXiv* **2019**, arXiv:1909.12086.
12. Wu, W.; Wang, F.; Yuan, A.; Wu, F.; Li, J. CorefQA: Coreference resolution as query-based span prediction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5−10 July 2020; pp. 6953–6963.
13. Song, S.; Wang, C.; Xie, Q.; Zu, X.; Chen, H.; Chen, H. A two-stage conversational query rewriting model with multi-task learning. In Proceedings of the Companion Proceedi-ngs of the Web Conference, Taipei Taiwan, 20−24 April 2020; pp. 6–7.
14. Xu, K.; Tan, H.; Song, L.; Wu, H.; Zhang, H.; Song, L.; Yu, D. Semantic Role Labeling Guided Multi-turn Dialogue ReWriter. *arXiv* **2020**, arXiv:2010.01417.
15. Vakulenko, S.; Longpre, S.; Tu, Z.; Anantha, R. Question rewriting for conversational question answering. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Jerusalem, Israel, 8−12 March 2021; pp. 355–363.
16. Yang, S.; Fu, B.; Yu, C.; Hu, C. Multi-Turn Conversation Rewriter Model Based on Masked-Pointer. *Beijing Da Xue Xue Bao* **2021**, *57*, 31–37.
17. Yi, Z.; Ouyang, J.; Liu, Y.; Liao, T.; Xu, Z.; Shen, Y. A Survey on Recent Advances in LLM-Based Multi-turn Dialogue Systems. *arXiv* **2024**, arXiv:2402.18013.
18. Liu, Q.; Chen, B.; Lou, J.G.; Zhou, B.; Zhang, D. Incomplete Utterance Rewriting as Semantic Segmentations. *arXiv* **2020**, arXiv:2009.13166.
19. Li, J.; Chen, Z.; Chen, L.; Zhu, Z.; Li, H.; Cao, R.; Yu, K. DIR: A Large-Scale Dialogue Rewrite Dataset for Cross-Domain Conversational Text-to-SQL. *Appl. Sci.* **2023**, *13*, 2262. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.