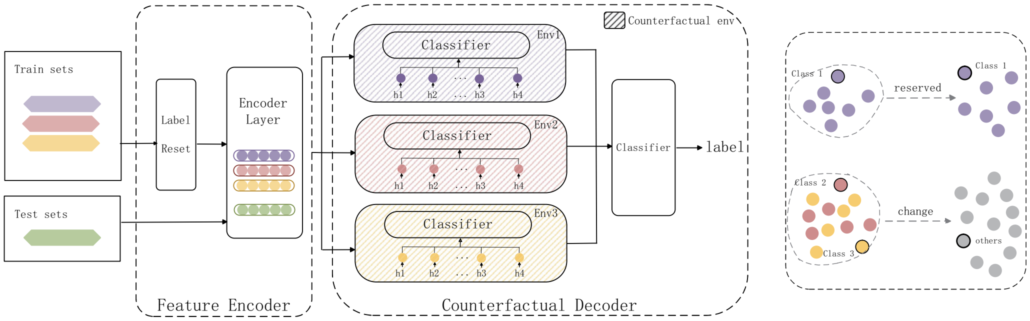
Graphical Abstract

Counterfactual Can be Strong in Medical Question and Answering

Zhen Yang, Yongbin Liu, Chunping Ouyang, Lin Ren, Wen Wen



(a) Model Structure

(b) Label Reset

Counterfactual Can be Strong in Medical Question and Answering

Zhen Yang[1], Yongbin Liu[1], Chunping Ouyang[1], Lin Ren[1], Wen Wen[1]

[a] *Computer School, University of South China, No.28, West Chang Sheng Road, HengYang, 421001, China*

Abstract

Medical Q&A is an extremely important task for medical AI, aiming to improve the efficiency of clinical diagnosis and achieve high assurance for treatment. Many approaches to medical Q&A are already available, but ignore the bias introduced by data generation mechanisms and unstructured text spurious correlation in Q&A tasks. The spurious correlation in the data is shown by that many words in the Q&A are not related to the answer but occupy a large weight, and these words can affect the feature representation. Besides, the data imbalance mechanism can cause the model to blindly follow the classes with large numbers, which affects the final answer. This confounders may mislead the model and limit its performance. In this paper, we propose a new counterfactual-based method that consists of a feature encoder and a counterfactual decoder. The feature encoder uses label reset to construct counterfactual data, ensuring that the model learns only specific types of knowledge at a time. The counterfactual decoder uses counterfactual data to find features that are causally invariant to the answer, learns class-specific knowledge to remove confounding biasfrom the data, and generate the final answer.The method is validated on machine learning and deep learning models for the medical dataset PubMedQA, respectively. Comprehensive experiments show that this method achieve state-of-the-art results and effectively mitigates data imbalance and text-based spurious correlation's confounding bias.

Keywords: Causal Inference, Counterfactual, Medical Question and Answer

1. Introduction

With the rapid development of machine learning and natural language processing in recent years, medical AI has gained full momentum. The main sources of data in the field of medical AI are electronic medical records and medical datasets. Researchers process these data based on two methods. One approach is Machine Summarization, which inputs the context into a sequence-to-sequence model and performs extractive and abstractive tasks. The other approach is the Embedding-based approach. The work inputs the context into a pre-trained model and then adds a linear for classification. These two approaches are widely used in many medical tasks, such as medical text classification, medical Q&A and medical analysis. As one of the important problems in the medical field, medical Q&A plays an important role in both clinical decision making and aiding diagnosis. Therefore, improving the accuracy of medical Q&A tasks is a current challenge that needs to be addressed. The input of medical Q&A consists of questions and contexts. The task is to comprehend and reason the text, find the part which is causally related to the question, and thus get an accurate answer. Most of the generative Q&A require generating a paragraph or a word, such as yes, no, or maybe.For the task of generating a paragraph, it is common to put the text into a sequence-to-sequence model for generalization and generation. Common networks such as pointer generation networks. For medical Q&A tasks that are answered with yes, no or maybe, a common approach[4, 5, 15], is to link the question and context as input, and feed into pre-training model and linear.

In the above task, there is spurious correlation in the text, and bias due to data imbalance. First, in Table 1, the red part indicates the words that are favorable for answering the question, while the blue part is not very helpful. However, due to the large proportion of the blue part, there is an influence on the model. This influence can bias the model, causing it to incorrectly associate these words with the question, leading to spurious correlation of the answers with these words. As a result, the ability of the model is weakened. Second, many datasets have data imbalance, as shown in Table 2. We used the dataset PubMedQA with unbalanced instances of each class, and data imbalance is always a challenge to the task. We give examples in Figure 1. When the proportional gap between the three classes in the data is too large, the model will blindly answer 'no' and 'maybe' as 'yes' because the number of 'yes' is the largest. And we would like to propose a method that can correct this error. As shown in Figure 2, from a causal inference perspective, the task suffers from spurious correlation of text and data imbalance mechanisms bias , which are confounding factors in the medical Q&A task and can affect the feature representation of the input data and mislead the model.

Question	Do mitochondria play a role in remodelling lace plant leaves during programmed cell death?
Context	Programmed cell death (PCD) is the regulated death of cells within an organism. The lace plant (Aponogeton madagascariensis) produces perforations in its leaves through PCD. The leaves of the plant consist of a latticework of longitudinal and transverse veins enclosing areoles. PCD occurs in the cells at the center of these areoles and progresses outwards, stopping approximately five cells from the vasculature.The role of mitochondria during PCD has been recognized in animals; however, it has been less studied during PCD in plants.
Long Answer	Results depicted mitochondrial dynamics in vivo as PCD progresses within the lace plant, and highlight the correlation of this organelle with other organelles during developmental PCD. To the best of our knowledge, this is the first report of mitochondria and chloroplasts moving on transvacuolar strands to form a ring structure surrounding the nucleus during developmental PCD. Also, for the first time, we have shown the feasibility for the use of CsA in a whole plant system. Overall, our findings implicate the mitochondria as playing a critical and early role in developmentally regulated PCD in the lace plant.
Answer	Yes.

Table 1: Case of PubMedQA

Therefore, the text spurious correlation within the textual information, and the bias brought by the data imbalance mechanism, are the confounding factors in this task, as shown in Figure 2a. We want to use causality to eliminate the influence of the confounding factor, that is, to block the path between the confounding factor and the input data, as shown in Figure 2b. By blocking this path, we can find the causal relationship between the input data and the answer, helping to improve the accuracy of the model.

Considering the main challenges mentioned above, we analyze them from a causality perspective. We believe the causal inference can mitigate the two problems[2, 18, 19, 26–29]. In this paper, we propose a new method combining counterfactual to mitigate the confounders due to text spurious correlations and data imbalance bias.

The model consists of a feature encoder and a counterfactual decoder. First, We need to construct counterfactual data for the task. The traditional

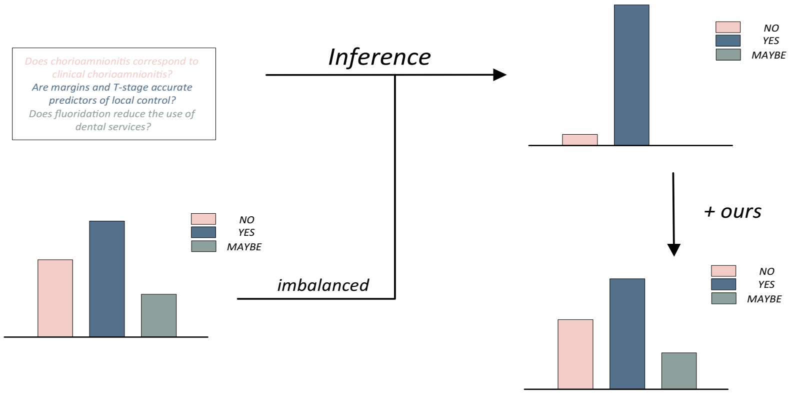


Figure 1: The figure reflects the data imbalance in the task. When there are many 'yes' classes in the dataset, the model will blindly answer as 'yes' and misclassify 'no' and 'maybe'. And our model can correct this error.

construction method always generates adversarial data[6, 22] or finds the antonyms of the text[17]. but the bias of word selection can have an impact on the textual information. So we use label resetting approach, which only changes the labels without changing the original text information.Second, due to the X and Y having causal invariance, each type Y of the answers has its fixed causal invariance factor X. By finding the causal invariant, we can eliminate the confounders from text spurious associations and data selection bias. Therefore, the primary purpose of our counterfactual decoder is to find causal invariance factors in each class. Specifically, the blue part is the confounding factor. When we counterfactualize the data, some feature representations associated with the facts change. However, the blue part may not change significantly because it is not related to the result, so the invariant part before and after the counterfactual is the blue part, which is the confounding factor. In this part, we construct three counterfactual environments to extract each class's causal invariant feature structure and judge the final result based on the three different causal feature structures obtained. The classifiers in the counterfactual environments are achieved by the Bayesian and DPCNN. We conducted experiments on the PubMedQA dataset, and our method achieved a significant improvement in accuracy and F1 values compared with the original results.In the appendix section, we give the overall process of the experiment.

The main contributions of this paper can be summarized as follows:

- We analyze the medical Q&A task from a causal perspective and use counterfactuals to remove confounding bias from the data imbalance generation mechanism and spurious correlations in the text. Our method finds causal invariance of Q&A and extracts the actual causal relationships in the context to get accurate answer.

- We propose label resetting to construct counterfactual data and calculate the probability of counterfactuals. Label resetting avoids the effect of word selection bias and preserves the textual content without destroying the original semantic information. The counterfactual probabilities got by this method can help us to better perform medical Q&A.

- We propose a new counterfactual model that uses an encoder to construct counterfactual data and extracts features with causally invariant features in the classes. A counterfactual decoder uses counterfactual data to find features that are causally invariant to the answer, learning knowledge specific to each class and optimize it to generate medical responses.

- Experimental results on the PubMedQA dataset show that our method significantly outperforms the optimal baseline by 7-20%. By analyzing the experimental results, our method effectively mitigates the textual spurious correlation and data imbalance problems and eliminates the confounding bias in medical Q&A.

Comprehensive experiments validate the accuracy of the method and demonstrate that the counterfactual method is beneficial in mitigating text spurious correlation and data imbalances bias. Due to our use of causality invariance, we believe that our method can also be transferred to other domains for mitigating spurious correlation and data imbalance problems in other domains.

Statistic	PQA-L	PQA-U	PQA-A
Number	1.0K	61.2K	211.3K
Prop.yes (%)	55.2	/	92.8
Prop.no (%)	33.8	/	7.2
Prop.maybe (%)	11.0	/	0.0

Table 2: PubMedQA data information

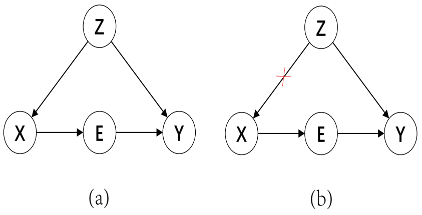


Figure 2: (a)The causal structure, where X is the input, including the question and context, E is the feature representation of the input data, Z is the confounding factor, which represents text spurious correlation and the data imbalance in the data set, and Y is the final answer. (b)Our ultimate goal is to block the backdoor path between X and Z and remove the bias caused by the confusion factor

2. Related Work

2.1. Medical Q&A

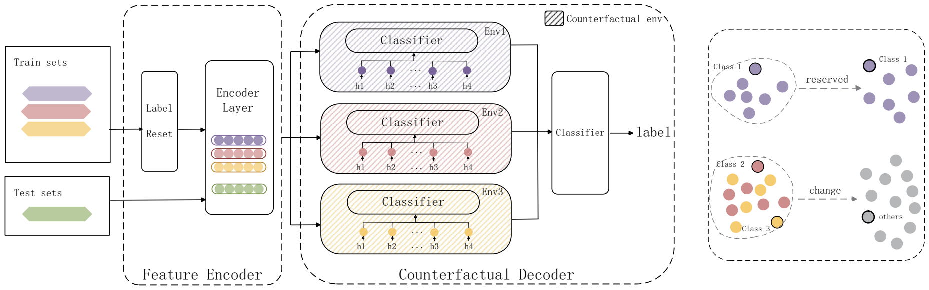
Medical Q&A, as one of the important tasks in medical field, plays an important role in clinical decision making and aiding diagnosis. In recent years, many researchers try to improve the accuracy of Medical Q&A by different pre-trained models or Q&A methods. The input of Medical Q&A generally consists of question and context, and the model needs to reason the text based on the question and context to get the final answer. In generative Q&A, when the answer is a paragraph of text, the model will generally infer from the input data, calculate the probability of the text, and extract the text from the input data to form the final answer, as in the classical pointer-based generative network. When the question is a simple judgmental bool type, a common approach is to concatenate the question and context , fine-tune it by pre-training models, such as Bert[5], Biobert[15] and Electra[4]. After that, the fully connected layer is connected to the downstream tasks to get the class of answers. However, this approach has two disadvantages. The first is that domain-specific pre-trained models due to their strong domain background can better vectorize the input data, but too much domain knowledge may exacerbate spurious associations between the input data. The second is when the data imbalance problem exists in the dataset, the model may blindly follow the classes with large sample size, making the representation and classification of features confusingly biased and causing the model to miss the causal relationships in the input data.

In addition, when the class of answers is limited and few, the answers can be classified directly. In machine learning, the earliest classification methods included SVM model[21], but it could not provide probabilities, so Logistic Regression[7] was proposed to provide probabilistic outputs. However, this is also a simple discriminative model that cannot learn the joint distribution.Therefore, Bayesian model[23] was proposed for the joint probability distribution of the input and output, based on the joint distribution outcome class. Then all these methods above put the features equally into the model, but different pairs of features contribute differently to the results.Therefore, Random Forest model[1] is proposed to evaluate the contribution of individual features efficiently. Then this method requires constant adjustment of parameters to achieve optimal results.And then the proposed XGBoost model[3] uses parallelism to reduce the iterative operations while achieving a great improvement in the algorithm accuracy. However, the XGBoost model also has the problem of overhead caused by the splitting of nodes at each layer.In contrast, the later proposed LightGBM model[11] with selective splitting and histogram algorithm exhibits memory and computational advantages. In addition to machine learning, classification methods have proliferated in deep learning.Besides machine learning, classification methods have emerged in deep learning. Early classification methods such as TextCNN[12], which use CNN to extract key information, are simple in structure and effective. However, where the filter_size is fixed, longer sequence information cannot be modeled, so TextRNN[16] emerged. Considering the complexity of natural language, unlike TextRNN, TextRCNN[14] uses a bidirectional cyclic structure aiming to obtain better information about the context. However, the above approach is less effective when dealing with long texts. Therefore, we need models with stronger feature extraction capability. And DPCNN[9] uses multi-layer convolution to enhance the feature extraction ability, which is better suitable for long text.Then, with the development of attention Transformer[24] was proposed, which allows for better global dependencies for various types of text processing. Besides using classification methods for classification, there are also methods that use pre-trained models for classification, such as BERT and ELECTRA.However, these are generic domains, to better fit the medical domain, BioBERT and BioELECTRA focus more on data from the medical domain and construct pre-trained models. All these methods are mature and can extract features better. However, as shown in Equation 1, when the data distribution is unbalanced, the models may blindly obey a certain class of data, rendering feature extraction and accurate classification meaningless. Besides, although some of these methods mitigate the problem of data imbalance, most of them are fitted to the data.

The models associate all input words with the answers without discovering the causal relationships in the data, and spurious correlations appear.

2.2. Causal Inference

Causal inference is a hot topic of wide interest in recent years, and has an important role in mitigating spurious correlations and data imbalance bias. [19] proposed causal intervention theory, and later intervention causality and counterfactual methods were used to discover causal invariance in data , and help different downstream tasks to eliminate confusion bias caused by spurious correlation or data imbalance. Among them, in the recommender system task, [25] used the counterfactual method to solve the click-through rate problem. In the vision task, [2] uses counterfactuals to help image classification,[28] uses counterfactuals for zero-shooting to improve the overall performance, [27] uses causal intervention to solve spurious correlations, and [18] uses counterfactuals to improve the VQA model. In natural language processing tasks, [29] de-biases remotely supervised named entities by causal interventions, and [26] also uses counterfactuals in causality to mitigate data imbalance.From the above approaches, we can see that causal interventions and counterfactuals can indeed solve the spurious correlation and data imbalance problems effectively. As shown in Figure 2, we plotted causal graph in medical Q&A task. Through the causal graph, we found that data imbalance mechanisms and spurious correlation in the text, as confounding factors, may have an impact on the feature representation of the input data and also on the answers. Therefore, we use the counterfactual to intervene input data to block the backdoor path between the input data and the confounders,to eliminate the spurious correlation in the feature representation and the bias caused by data imbalance. Several counterfactual methods exist for solving spurious correlation and data imbalance, but they have several problems. Some methods require pre-training domain-specific text vectors when acquiring input data vectors, and some methods need to utilize external counterfactual data as an additional aid. However, our advantage is that although we target medical texts, we do not need external knowledge of the medical domain. Moreover, our method can construct counterfactual data directly from the input data without the need to introduce additional counterfactual data.



(a) Model Structure

(b) Label Reset

Figure 3: (a)shows the model structure, consisting of an encoder and a decoder, (b)shows the details of Label Reset

3. Methods

In this section, we will analyze the problems with Medical Q&A tasks from a causal perspective, and then, we will describe our model and give a causal theory explanation.

3.1. Causal inference analysis

Our task is performed on the PubMedQA dataset, and through Table 2 we observe that in PubMedQA, PQA-L denotes labeled data, which is our experimental data. We find serious data imbalance in the experimental data. As shown in Figure 2, when there are confounding factors in the task, such as unbalanced data distribution and spurious correlation of text, this affects the feature representation and leads to confusion of answers and reduces the accuracy of the model. On the other hand, in Equation 1, where X denotes the input to the model and Z denotes confounding factors in the data set, such as spurious correlation and data imbalance. When the amount of data in a class of data is particularly large, P(z=0|X) approximates to 1, the final result depends approximately on P(Y|X, z=0), and the model becomes blind.

$$P(Y|X) = \sum_{z} P(Y|X, z)P(z|X) \tag{1}$$

As shown in Figure 2, in order to eliminate the unbalanced data distribution and spurious correlation of text, we need to block the backdoor path between X and Z to ensure that X only connects to Y through E.Therefore, we apply the counterfactual treatment to X as shown in Equation 2. $\mathrm{Y}_x = \mathrm{y}$

in Equation 2 is the counterfactual representation, and we obtain Equation 2 by the full probability formula,.

$$P(Y_x = y) = \sum_z P(Y_x = y|Z = z)P(z) \tag{2}$$

After that, we obtain Equation 3 according to the backdoor counterfactual law in causality. that is, there is a backdoor path between X and Z. so we usually consider so the addition of X has no effect on Z.

$$P(Y_x = y) = \sum_z P(Y_x = y|Z = z, X = x)P(z) \tag{3}$$

Then, according to the observation consistency criterion, we obtain Equation 4. we change Y$_x$ to Y because the added X happens to be the same as the X in Y$_x$.

$$P(Y_x = y) = \sum_z P(Y = y|Z = z, X = x)P(z) \tag{4}$$

In this task, the data includes a total of three classes: yes, no and maybe, so we divide Z into three values, where Z=0 means the answer is no, Z=1 means the answer is yes, and Z=2 means the answer is maybe, therefore, Equation 4 can be expanded into Equation 5.

$$\begin{aligned} P(Y_x = y) = {} & P(Y = y|Z = 0, X = x)P(z = 0) \\ & +P(Y = y|Z = 1, X = x)P(z = 1) \\ & +P(Y = y|Z = 2, X = x)P(z = 2) \end{aligned} \tag{5}$$

With Equation 4, we find that the second half of the formula, P(z), can be gotten by the classifier. And the first half of the formula, P(Y=y|Z=z,X=x) can be gotten by the counterfactual. Since Z has three different values, we build three counterfactual classifiers, and each classifier gets one result. Theoretically, we only let one classifier see one kind of data at a time, thus avoiding the impact of data imbalance.

3.2. Label Reset

With Equation 5, we want to construct counterfactual data to get the counterfactual probabilities. The counterfactual is to infer possible results from conditions which do not happen. Previous methods usually need to find the contextual antonym, but the word selection bias will have some impact on the data. Unlike the usual counterfactual data construction methods, we only change the labels and the text remains unchanged. Due to the specificity of the generated answers, even the text remains unchanged, the labels contain some important information. For example: *Do mitochondria play a role in remodelling lace plant leaves during programmed cell death? –Yes.* the answer is yes, it means that mitochondria is related to remodel lace plant. When we change the label to no, we have changed some established facts. Here we assume that mitochondria and remodel lace plant are not related. It means that even if there is no mitochondria , it does not affect the remodel lace plant.

As for the label transformation, we give a detailed explanation in Figure 3(b). As shown in Figure 3(b),when constructing the first counterfactual classifier data, we need to keep the data with label 0, Therefore, for the data with answer 'No'. the data is unchanged and the label is preserved. For the data whose original label is not 'No', the text content remains unchanged and the label is changed to other. In this way, we divide the data obtained by the first counterfactual classifier into two types of data with label No and label Other. This approach helps the model to better distinguish the data labeled as No from the other class.

3.3. Counterfactual Bayesian

Bayesian inference is centered on the well-known inverse formula. As in Equation 6. the formula shows that the belief of hypothesis R under the condition that evidence e is known can be obtained by multiplying our previous belief P(R) with the likelihood similarity P(e—R). p(e—R) denotes the probability that e is true if R is true. Where P(R—e) becomes the posterior probability and P(R) is the prior probability. P(e) is almost not to be considered, it is simply a normalized constant. The most important significance of Bayesian is that it represents P(R—e) using a value obtained directly from empirical knowledge, which is a difficult value to estimate.By introducing Bayes, we can perform multiple classification tasks quickly and simply. Meanwhile, the assumption that the distributions are independent basically holds in text data. Therefore we can efficiently train the features using Bayes.

$$P(R|e) = \frac{P(e|R)P(R)}{P(e)} \tag{6}$$

Counterfactual-Bayesian(CFB) combines counterfactual with Bayesian[23]. As shown in Figure 3, both of our methods consist of a feature encoder and a counterfactual decoder. The feature encoder use label resetting to construct the counterfactual data and converting the input data into feature vectors. And the counterfactual decoder consists of three counterfactual classifiers and a general classifier.

Feature encoder. In the feature encoder part of this method, we insert label resetting to construct the counterfactual data by changing the data labels while keeping the textual content of the input data unchanged. And in the vector encoding part, we find that in each category, some common words may appear. Therefore, in the feature vector encoding stage, we use the TF-IDF vector technique to count the occurrence frequency of words in the text and get the vector representation according to the word frequency, as shown in Equation 7, where we splice the question with the context as the input of the encoder and get the word frequency matrix as the output by TF-IDF[13]. Where n$_{p,q}$ is the number of occurrences of the word in the document d$_q$, and the denominator is the sum of the occurrences of all words in the document d$_q$.

$$tf_{p,q} = \frac{n_{p,q}}{\sum_k n_{r,q}} \tag{7}$$

Counterfactual decoder. From Equation 5, we need to calculate three counterfactual probabilities and three ordinary probabilities. Therefore, we use three counterfactual classifiers to calculate three counterfactual probabilities and an ordinary classifier to obtain three ordinary probabilities. In this approach, we focus on common words. We believe that there must be some words that occur frequently in each class. For example, in the class labeled 'yes', words such as 'significant' and 'good performance' occur frequently, which is a positive affirmation. Therefore, when we focus on common words, we want to use a model to find the words that are common in these classes. And use these words for better classification. For the three counterfactual classifiers, as the Bayesian model is effective in text classification, it focuses on the statistics of words. And the multinomial Bayesian model is based on the original Bayesian theory, the probability distribution is assumed to obey a simple multinomial distribution. Therefore, we use the Bayesian model as a counterfactual classifier. We take the vector matrix obtained from the encoder part as input and calculate the counterfactual probabilities of the samples by Equation 8.

$$P(X_i = x_{ij}|Y = C_k) = \frac{x_{ij} + \alpha}{s_k + t\alpha} \tag{8}$$

Where P(X$_i$=x$_{ij}$|Y=C$_k$) is the j-valued conditional probability of the i-dimensional feature of class k. The output of the s$_k$ training set is the number of samples of class k. α is a constant greater than 0, often taken as 1, and is Laplace smoothing. Other values can also be taken. As for the common classifier, in real cases, there will be only one answer, therefore, we set the result of the common classifier as a binary result. It is obtained by Equation 9.

$$P(z = i) = \begin{cases} 1, & P_{z=i} > P_{z=j}, P_{z=i} > P_{z=k}, \\ & i \neq j, i \neq k \\ 0, & others \end{cases} \tag{9}$$

3.4. Counterfactual Deep Pyramid Convolutional Neural Networks

DPCNN has good classification ability. Its bottom layer is the region embedding, which is the result of convolution of the convolution layer of TextCNN containing multi-size convolution filters. After that, it also superimposes two layers of equal-length convolution to improve the richness of the embedding representation. Then the model compresses the sequence length to half by pooling, which increases the perceived text fragments. DPCNN solves the problem that TextCNN cannot obtain long-range text dependencies by a clever structural design. The contexts in Q&A tasks are usually long texts. By introducing DPCNN, we can learn long texts better while extracting features accurately.

Counterfactual-Deep Pyramid(CFDP) combines counterfactual with Deep Pyramid Convolutional Neural Networks[9]. Figure 3 illustrates our model.

Feature encoder. In the vector encoding part, unlike model CFB, we want to find the causal structure that is invariant in the class, so we simply connect the input question to the context and then use glove[20] to vectorize the input data. Here, we use a 300-dimensional vector that fixes the length of the input data to 256.

Counterfactual decoder. In the counterfactual decoding section, by observing the experimental data, we find that all the questions are whether X and Y are related. In the context, the text will make a description of the change of Y when X changes. By inferring and extracting features from the text, we find that each class has its own causally invariant structure. For example, in the yes class, a change in X causes a change in Y, which means that X is causally related to Y, so it is possible to answer the question that X is related to Y. Causal inference works to solve the problem of spurious correlations in text, so we use the counterfactual to let the classifier see only one class of data at a time and work to extract only the feature structure in one class of data and find the fixed causal structure. This is the interpretability of the three counterfactual probabilities in Equation 5. Besides, we believe that the counterfactual data in each class, by changing the label, makes the features having a causal relationship with the label also change. By comparing the features in the before and after classes, we can find the features that have causal invariance. This class of features is the one that really affects the answer. We use the counterfactual data to find this type of knowledge and learn it using a counterfactual classifier. In the counterfactual classifier, we utilize the DPCNN model, which first convolves the text using multiple convolutional layers to generate embedding, followed by equal-length convolution to obtain semantic information for each word modified by context. Also, considering the excessive length of medical text, the pooling layer then shortens the sequence length to make the perceived text fragments longer. By continuously traversing one class of data in the dataset, the training capability of the model is strengthened, and the model can acquire fixed features to enhance the judgment of the class, which is the causal invariant structure that makes the data correctly classified. Similarly, for the common classifier, in the real case, there will be only one answer, so we set the result of the common classifier as a binary result. The same is also obtained by Equation 8.

3.5. Dataset

PubMedQA[8] is a Medical Q&A dataset, the format is shown in Tab 1. We divide the dataset into two small datasets according to [8], one includes question, context, and the final answer according to the input data. the other dataset input includes long answer in addition to question and context, and the final answer is also got according to the input data.

Name	Value	Note
batch_size	50	batch size
Pad_size	256	Maximum length of a sentence
lr	2e-4	learning rate
Embed	300	dimension of word embeddings
Num_filters	250	Number of convolution kernels
Dropout	0.5	dropout
Epoch	20	epoch number
gamma	0.9	scheduler gamma

Table 3: The hyperparameters of CFB and CFDP

3.6. Parameters setting and Evaluation metrics

The hyperparameters of CFB and CFDP is shown in Table3.And we use the accuracy rate and macro avg F1 value as evaluation metrics.

Machine Learning				
Model	**without long answer**		**with long answer**	
	P	F1	P	F1
Bayesian	36.70	20.40	40.10	24.50
SVM	51.70	48.20	53.30	50.70
Random Forest	41.00	45.70	43.70	46.80
XGBoost	51.10	48.20	52.70	51.10
LightGBM	42.40	41.40	44.90	45.80
LogisticRegression	40.40	22.90	45.50	28.70
CFB	**60.60**	**53.00**	**60.80**	**55.00**

Table 4: CFB and machine learning baseline result

3.7. Baselines

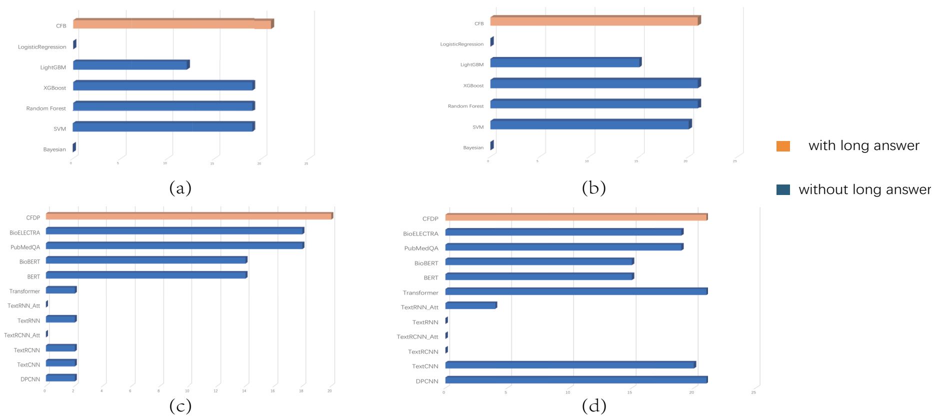
Our method CFB is a combination of counterfactual and machine learning method, so we compare CFB with machine learning methods, including Bayesian[23],which can learn the joint distribution of the input. SVM[21],which performs well in binary classification. Random Forest[1], which is proposed to evaluate the contribution of individual features efficiently. XGBoost[3], which uses parallelism to reduce the iterative operations while achieving a great improvement in the algorithm accuracy. LightGBM[11], which with

Deep Learning				
Model	**without long answer**		**with long answer**	
	P	F1	P	F1
DPCNN	50.10	26.74	51.70	50.45
TextCNN	55.40	23.77	52.70	51.39
TextRCNN	55.40	23.77	50.70	33.48
TextRCNN_Att	51.00	28.74	45.70	33.84
TextRNN	55.40	23.77	55.40	23.77
TextRNN_Att	48.60	29.46	49.90	39.91
Transformer	55.40	23.77	54.10	50.67
BERT	55.40	23.77	55.80	28.26
BioBERT	56.20	26.35	56.40	28.46
PubMedQA	60.80	50.76	61.40	51.54
BioELECTRA	61.40	51.25	61.70	51.87
CFDP	**62.00**	**59.03**	**62.00**	**58.33**

Table 5: CFDP and deep learning baseline result

selective splitting and histogram algorithm exhibits memory and computational advantages. And Logistic Regression[7],which proposed to provide probabilistic outputs.

Our model CFDP combines causal counterfactual and deep learning method, so we compare CFDP with common deep learning algorithms, including (1) DPCNN[9], which can extract long-range text dependencies by deepening the network. (2) TextCNN[12], it uses kernels of different sizes to extract key information in sentences. (3) TextRCNN[14], which uses bidirectional RNN and maximum pooling. (4) TextRCNN/Att, adds attention. (5) TextRNN[16], which can better capture longer sequence information. (6) TextRNN/Att, adds attention. (7) Transformer[24], it forms a very efficient classification model by encoding feature extraction and classification. (8) Fine-tuning of BERT[5], which is trained by language models from a large text corpus for downstream tasks in NLP. (9) Using BioBERT[15], similar to BERT, in that BioBERT involves a medical text corpus, more relevant in the medical field. (10) Fine-tuning of BioBERT[8], which designs a multi-step fine-tuning method for better target text classification of datasets. (11) fine-tuning of BioELECTRA[10], which constructs a new pre-training model to be applied to NLP downstream tasks in NLP downstream tasks.



(a)

(b)

(c)

(d)

Figure 4: The figure shows the results of all methods on the least number of classes. Where (a) is the result obtained for CFB and baseline without long answers as input, (b)is the result with long answers as input. (c) is the result obtained for CFDP and baseline without long answers as input, and (d)is the result with long answers as input.

3.8. Results and Analysis

Tables 4-5 show the results. As can be seen from the tables, there are many results with F1 equal to 23.77. This is due to the blind classification of the model due to data imbalance, which changes the answers into the same class. In Tables 4, our method CFB improved by 5%-33% over baseline. This shows that our approach of focusing on common words is right. This method does lead to better finding the correct answer. And comparing CFB and Bayesian, we found that adding the causal counterfactual can help the model to have a great improvement in the same case of using Bayesian, which also proves the effectiveness of counterfactual. In Table 5, our method CFDP improves by 8%-36% over baseline. This shows that we can find the causal features associated with the answers. By changing the labels, we fix some invariant causal features. When the model learns the invariant causal structure, it can answer better. And comparing CFDP and DPCNN, we find that the introduction of counterfactuals helps the model perform very well in answering with the same use of DPCNN, which again proves the effectiveness of counterfactuals. Comparing Tables 4 and 5,we can see that our model can mitigate spurious correlations. And in the table, our model F1 value is not equal to 23.77 and does not bias all answers to the class with large numbers. This indicates that our method effectively mitigates the problem of data imbalance.

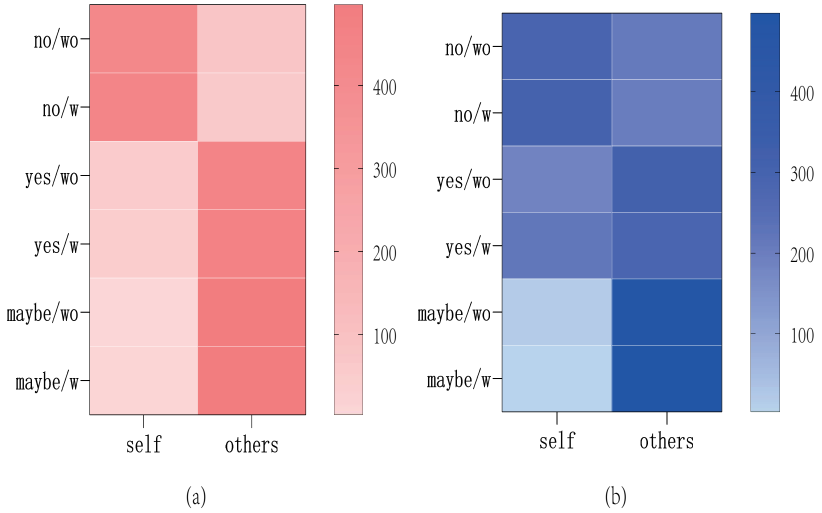
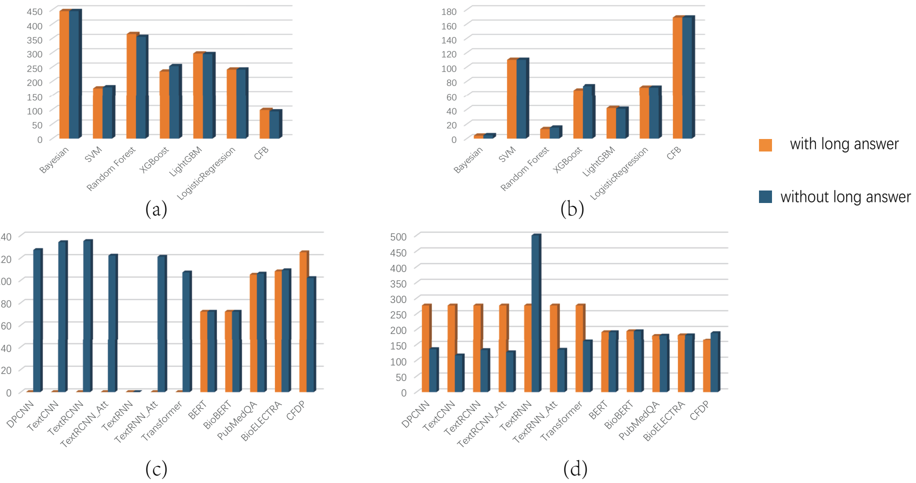


Figure 5: The figure shows the results achieved by each counterfactual classifier in both approaches, where (a)is CFB, (b)is CFDP, wo means without long answer and w means with long answer

In Figure 4, we count the classification results for classes with a low number of classes in the model. In the figure, there are many baselines where the number of discoveries is 0. This is because the model blindly obeys those classes with large numbers and classifies all classes as these classes. And by observing it can be found that our method finds the highest number both in the case of with long answers and without long answers. It can be found that our model can improve the number of identifications and enhance the identification of such classes. This proves that our method does not blindly classify those classes with small numbers into classes with large numbers. This proves that our method alleviates the problem of data imbalance.

Our method finds the invariant causal structure in the text related to the answer. In the combined CFDP, we encode the text using glove vectors and perform feature extraction by constant convolution. In Tables 4-5, our accuracy and F1 values show a significant improvement compared to other models. This shows that we did find some features that other models did not accurately locate, and these are the causal features that really affect the answer. These features are not changed with external changes. These features are also fixed when there are confounding factors in the context. Finding these features leads to better answers. And our model improves the accuracy and F1 value despite the presence of confusion factors in the



(a) (b)

(c) (d)

Figure 6: This figure compares the number of accurate identifications for both yes and no categories obtained for both with long answer and without long answer data under different models. Where (a) is the recognition result of CFB and its baseline in the no class, (b)is the result in the yes class, (c) is the recognition result of CFDP and its baseline in the no class, and (d) is the result in the yes class.

context. This proves that our method reduces some spurious associations and finds more invariant causal structures related to the answers.

In Figure 5, we analyze the effect of the three counterfactual classifiers in the decoding part. We found that in the CFB combination, the classifier for the ’no’ class found more data than the classifier for the ’other’ class, which indicates that it is more effective in finding this type of data. And in the CFDP method, the class labeled ’no’ also achieves good results, while the class labeled ’yes’ can be found equally well. But neither the CFB method nor the CFDP method actively discovers the class with the label ’maybe’, but more often excludes the data with the labels ’yes’ and ’no’ among all are excluded from all data. The ‘maybe ’class is determined by negating other classes.

Comparing the different results gotten with different input data under our model, we found that the data with long answer has a little lower result than the data without long answer. So we compared the number of accurate identification of yes and no classes obtained by each model for both data. As shown in Figure 6, except for the extremely biased case where all are classified into the same class, in most cases, the number of classes identified

Add Label Reset				
Model	without long answer		with long answer	
	P	F1	P	F1
Bayesian	38.50	21.20	41.70	27.30
SVM	53.30	50.60	55.80	53.70
Random Forest	44.20	49.10	45.20	49.30
XGBoost	53.10	51.70	55.30	52.80
LightGBM	46.30	44.10	48.30	47.20
LogisticRegression	43.70	29.70	47.80	33.80
DPCNN	51.60	28.83	53.20	53.27
TextCNN	55.40	23.77	54.30	54.23
TextRCNN	55.40	23.77	51.20	35.28
TextRCNN_Att	53.30	30.45	47.10	33.28
TextRNN	55.40	23.77	55.40	23.77
TextRNN_Att	50.70	30.63	50.90	40.37
Transformer	55.40	23.77	55.60	53.24

Table 6: the baseline add label reset

in the data with long answer remains the same or increases, while the effect is not good for classes with small sample size, which indicates that the long answer is too granular and the text is too long, which plays a confusing role in our case.We can also see from Table 1 that many of the words in the long answer are explanatory or expanded. None of these words are directly related to the final answer and do not help the model to reason directly, but rather can cause confusion to the answer.

3.9. Ablation study and parameter setting

To validate the effectiveness of label resetting, We use the constructed counterfactual data to train the model. but in testing, we use the model to get the answers directly. The results are shown in Table 6. By comparing Table 4-5 and Table 6, we found that the accuracy of baseline was improved. This clearly proves the effectiveness of label resetting. Besides, the blind prediction methods in the table(F1 equals 23.77) were changed. It indicates that label resetting can mitigate the data imbalance problem.

To validate the counterfactual classifier, baseline uses counterfactual data and calculates the counterfactual probability then got the answer. The Table 7 illustrates results. By comparing Table 6 with Table 7, we find that adding the counterfactual classifier can improve the model. It shows that the coun-

Add Label Reset and Counterfactual				
Model	**without long answer**		**with long answer**	
	P	F1	P	F1
Bayesian(CFB)	60.60	53.00	60.80	55.00
SVM	56.80	53.00	57.00	54.60
Random Forest	47.00	51.00	47.00	50.00
XGBoost	54.20	53.20	57.40	54.80
LightGBM	50.40	48.00	50.60	50.00
LogisticRegression	49.00	34.00	49.60	35.40
DPCNN(CFDP)	62.00	59.03	62.00	58.33
TextCNN	54.80	24.67	54.00	56.41
TextRCNN	52.60	36.68	53.60	37.39
TextRCNN_Att	46.60	31.74	49.60	34.59
TextRNN	54.40	36.47	55.40	23.77
TextRNN_Att	52.40	28.87	51.80	41.25
Transformer	56.20	51.85	57.80	56.37

Table 7: the baseline add label reset and counterfactual

terfactual classifier has an important role for the Q&A task.Bayesian adding label reset and counterfactual classifier is our CFB method, and similarly DPCNN becomes CFDP method. The table also shows that our two combinations are the optimal combination.

In order to unify the results of the three models, we want to map the three results into the same interval by setting hyperparameters. In order to compare the effects of different hyperparameters, we conducted experiments on parameter setting.where n_h is the parameter for the no class, y_h is the parameter for the yes class, and m_h is the parameter for the maybe class.Table 8 shows our results.We find that the model performs optimally for a no class parameter of 0.7.

n_h	y_h	m_h	Acc	F1
1.0	1.0	1.0	58.20	50.96
0.9	1.0	1.0	61.80	59.14
0.8	1.0	1.0	62.20	59.39
0.7	1.0	1.0	61.90	59.25
0.6	1.0	1.0	62.00	59.23

Table 8: Effect of different parameter settings on experimental results

In addition, we changed the value of the batch size and then adjusted the parameters of the three classes to the same range of values. After comparison, we found that different batch size has a great impact on the three parameters. The larger the size the larger the range of results obtained, and we need to adjust the parameters more obviously.Table 8 shows the results for a batch size of 50, and Table 9 shows the results for a batch size equal to 64.We found that when the size is too long, the model performs poorly instead.

n_h	y_h	m_h	Acc	F1
0.6	1.0	5.0	54.80	42.97
0.5	1.0	5.3	62.80	46.71
0.5	1.0	5.0	53.60	43.14
0.5	1.0	4.5	57.20	41.45
0.5	1.0	4.0	59.60	41.56

Table 9: Effect of different parameter settings on experimental results

4. Conclusion and Future Work

In this paper, we propose a model consisting of a feature encoder and a counterfactual decoder to combine counterfactual with machine learning. The method mitigates the data imbalance and spurious correlation problems. In the future, we will try to combine causality with machine learning for more applications to further reduce spurious correlations.

Acknowledgements

An earlier version of this paper has been presented in the 2022 IP&MC Causal Inference in Information Science workshop.

References

[1] Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

[2] Chang, C.H., Adam, G.A., Goldenberg, A., 2021. Towards robust classification model by counterfactual and invariant data generation, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. pp. 15207–15216.

[3] Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.

[4] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D., 2020. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 .

[5] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[6] Glockner, M., Shwartz, V., Goldberg, Y., 2018. Breaking nli systems with sentences that require simple lexical inferences, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 650–655.

[7] Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X., 2013. Applied logistic regression. volume 398. John Wiley & Sons.

[8] Jin, Q., Dhingra, B., Liu, Z., Cohen, W.W., Lu, X., 2019. Pubmedqa: A dataset for biomedical research question answering. arXiv preprint arXiv:1909.06146 .

[9] Johnson, R., Zhang, T., 2017. Deep pyramid convolutional neural networks for text categorization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 562–570.

[10] raj Kanakarajan, K., Kundumani, B., Sankarasubbu, M., 2021. Bioelectra: pretrained biomedical text encoder using discriminators, in: Proceedings of the 20th Workshop on Biomedical Language Processing, pp. 143–154.

[11] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems 30.

[12] Kim, Y., 2014. Convolutional neural networks for sentence classification. Eprint Arxiv .

[13] Kumar, M., Vig, R., 2011. Term-frequency inverse-document frequency definition semantic (tids) based focused web crawler, in: International Conference on Computing and Communication Systems, Springer. pp. 31–36.

[14] Lai, S., Xu, L., Liu, K., Zhao, J., 2015. Recurrent convolutional neural networks for text classification, in: Twenty-ninth AAAI conference on artificial intelligence.

[15] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J., 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36, 1234–1240.

[16] Liu, P., Qiu, X., Huang, X., 2016. Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 .

[17] Lu, K., Mardziel, P., Wu, F., Amancharla, P., Datta, A., 2020. Gender bias in neural natural language processing, in: Logic, Language, and Security. Springer, pp. 189–202.

[18] Niu, Y., Tang, K., Zhang, H., Lu, Z., Hua, X.S., Wen, J.R., 2021. Counterfactual vqa: A cause-effect look at language bias, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12700–12710.

[19] Pearl, J., 2009. Causal inference in statistics: An overview. Statistics surveys 3, 96–146.

[20] Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

[21] Platt, J., 1998. Sequential minimal optimization: A fast algorithm for training support vector machines .

[22] Ribeiro, M.T., Singh, S., Guestrin, C., 2018. Semantically equivalent adversarial rules for debugging nlp models, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 856–865.

[23] Sahami, M., Dumais, S., Heckerman, D., Horvitz, E., 1998. A bayesian approach to filtering junk e-mail, in: Learning for Text Categorization: Papers from the 1998 workshop, Citeseer. pp. 98–105.

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

[25] Wang, W., Feng, F., He, X., Zhang, H., Chua, T.S., 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1288–1297.

[26] Wu, Y., Kuang, K., Zhang, Y., Liu, X., Sun, C., Xiao, J., Zhuang, Y., Si, L., Wu, F., 2020. De-biased court's view generation with causality, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 763–780.

[27] Yang, X., Zhang, H., Qi, G., Cai, J., 2021. Causal attention for vision-language tasks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9847–9857.

[28] Yue, Z., Wang, T., Sun, Q., Hua, X.S., Zhang, H., 2021. Counterfactual zero-shot and open-set visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15404–15414.

[29] Zhang, W., Lin, H., Han, X., Sun, L., 2021. De-biasing distantly supervised named entity recognition via causal intervention, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4803–4813.

Appendix A. Appendix

We give a case of the PubMedQA dataset.

Question	Do mitochondria play a role in remodelling lace plant leaves during programmed cell death?
Context	Programmed cell death (PCD) is the regulated death of cells within an organism. The lace plant (Aponogeton madagascariensis) produces perforations in its leaves through PCD. The leaves of the plant consist of a latticework of longitudinal and transverse veins enclosing areoles. PCD occurs in the cells at the center of these areoles and progresses outwards, stopping approximately five cells from the vasculature. The role of mitochondria during PCD has been recognized in animals; however, it has been less studied during PCD in plants.
Long Answer	Results depicted mitochondrial dynamics in vivo as PCD progresses within the lace plant, and highlight the correlation of this organelle with other organelles during developmental PCD. To the best of our knowledge, this is the first report of mitochondria and chloroplasts moving on transvacuolar strands to form a ring structure surrounding the nucleus during developmental PCD. Also, for the first time, we have shown the feasibility for the use of CsA in a whole plant system. Overall, our findings implicate the mitochondria as playing a critical and early role in developmentally regulated PCD in the lace plant.
Answer	Yes.

Table A.10: Case of PubMedQA dataset

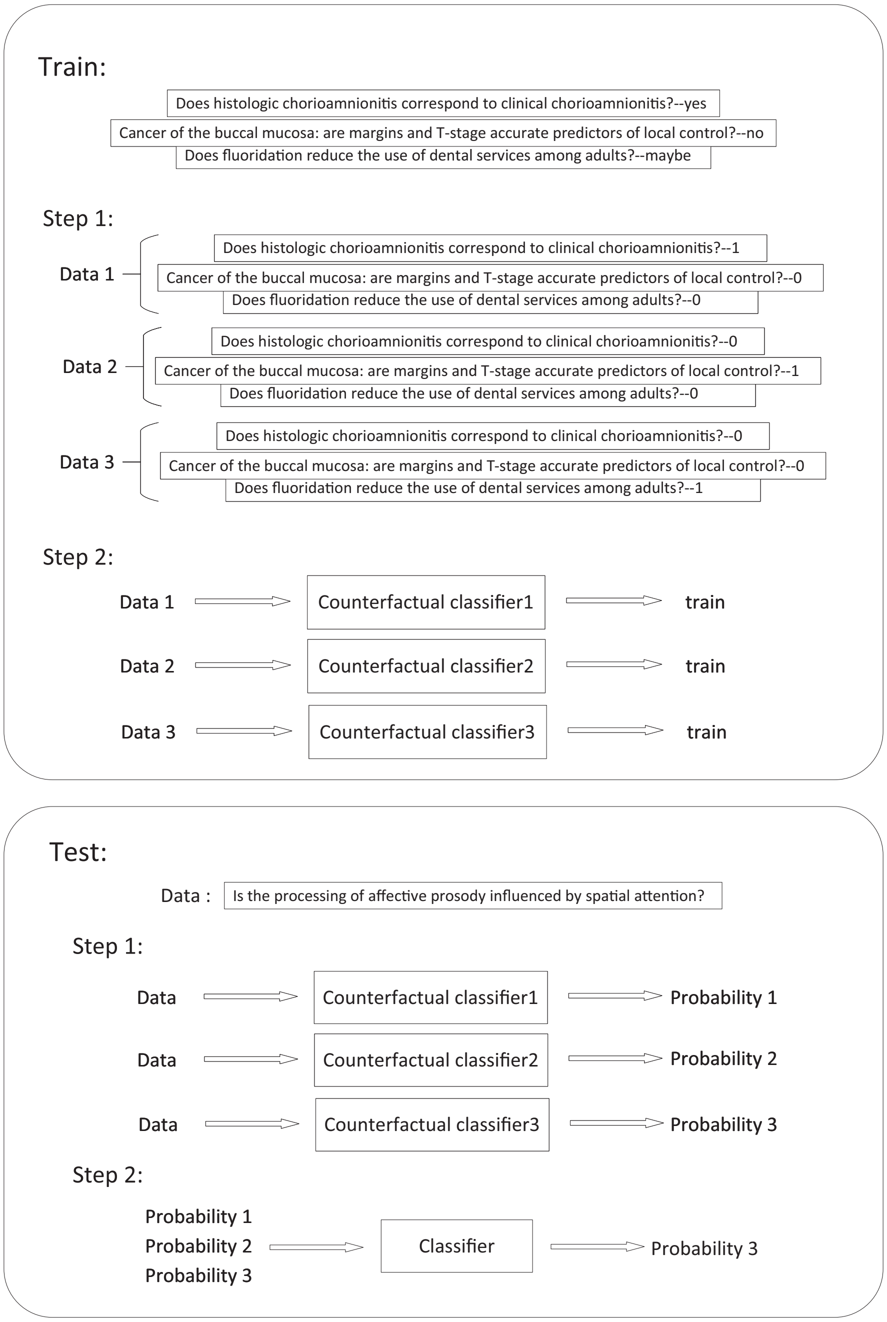

Train:
Does histologic chorioamnionitis correspond to clinical chorioamnionitis?--yes
Cancer of the buccal mucosa: are margins and T-stage accurate predictors of local control?--no
Does fluoridation reduce the use of dental services among adults?--maybe
Step 1:
Data 1
Does histologic chorioamnionitis correspond to clinical chorioamnionitis?--1
Cancer of the buccal mucosa: are margins and T-stage accurate predictors of local control?--0
Does fluoridation reduce the use of dental services among adults?--0
Data 2
Does histologic chorioamnionitis correspond to clinical chorioamnionitis?--0
Cancer of the buccal mucosa: are margins and T-stage accurate predictors of local control?--1
Does fluoridation reduce the use of dental services among adults?--0
Data 3
Does histologic chorioamnionitis correspond to clinical chorioamnionitis?--0
Cancer of the buccal mucosa: are margins and T-stage accurate predictors of local control?--0
Does fluoridation reduce the use of dental services among adults?--1
Step 2:
Data 1
Counterfactual classifier1
train
Data 2
Counterfactual classifier2
train
Data 3
Counterfactual classifier3
train
Test:
Data :
Is the processing of affective prosody influenced by spatial attention?
Step 1:
Data
Counterfactual classifier1
Probability 1
Data
Counterfactual classifier2
Probability 2
Data
Counterfactual classifier3
Probability 3
Step 2:
Probability 1
Probability 2
Probability 3
Classifier
Probability 3

Figure A.7: example process