文章编号：2095-2163(2021)01-0001-07 中图分类号：TP183 文献标志码：A

基于 Transformer 的机器阅读理解对抗数据生成

范 玚，刘秉权
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要： 机器阅读理解任务是衡量模型对于文本信息理解程度的一种重要方式，一直以来备受关注。近年来，很多学者在这一任务上提出了自己的模型，并取得了相当不错的成绩，其中一部分甚至已经超越了人工回答的准确率。然而，这些模型是否真正地、深入地理解了文本语义，还是仅依靠浅层的词语相似度和答案类型来进行简单的搜索？为了进一步评价阅读理解模型对于文章语义的理解程度，本文提出了一种基于 Transformer 结构的对抗数据生成方法，并对主流阅读理解模型进行了检测。

关键词： 机器阅读理解；文本生成；Transformer 结构；深度学习

Adversarial data generation for reading comprehension with Transformer

FAN Yang, LIU Bingquan
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 Machine reading comprehension is an important way to measure the model´s understanding of nature language. In recent years, many researchers have proposed their own models in this task, and achieved quite good results, some of which have even exceeded the human performance. However, do these models really and deeply understand human language, or simply rely on shallow word similarity and answer type to search for true answers? In order to evaluate systems´ real language understanding abilities, the paper proposes a new method of to generate adversarial data based on Transformer structure, and test the mainstream reading comprehension models on the dataset in the paper.

【Key words】 machine reading comprehension; text generation; Transformer architecture; deep learning

0 引 言

近年来，随着深度学习、预训练语言模型[1-5]等先进技术的相继问世，计算机理解人类语言的能力获得了长足的进步，许多自然语言处理领域的任务都有了新的突破，机器阅读理解任务也重新受到了人们的关注。

许多学者提出了不同的机器阅读理解数据集，其中较为出名的，有斯坦福研究者提出的 Stanford Question Answering Dataset（SQuAD）数据集[6]。这是一个片段抽取型的阅读理解数据集，其中共包含了 536 篇文章和 107 785 个文章-问题对。许多研究者针对这一任务提出了自己的方法，其中一些优秀的模型得到的性能甚至已经超过了人工的准确度。

然而，这样的片段抽取式阅读理解任务，由于答案原文可以直接在文章中找到，并且答案所在原文中的位置附近的词汇和问句中的词汇往往具有很大的相似度[7-8]。所以，仅通过简单的词语相似度的匹配，和对答案词性的预测，就可以很大程度上寻找到正确答案，进而解决这一问题。这就使得人们不由得产生了一个疑问，机器是否真正地、深入地理解了文章的意思？为了解决这一问题，本文通过对 SQuAD 数据集进行一定限度的改进，即在文章中增加一些可能对模型选择答案产生误导的句子，来进一步检测模型对于文章理解的程度。

本文的主要内容安排如下：第 1 节主要介绍本实验用的数据集；第 2 节简要阐述了现有的 2 种机器阅读理解对抗数据集，及其生成的方法；第 3 节主要提出了本文中生成对抗阅读理解数据的方法，及其中涉及的相关数据和工具；第 4 节给出了主流机器阅读理解模型在新生成的对抗数据集上的性能表现及结果分析，第 5 节是本次研究的工作总结。

1 实验数据集

本文所主要使用的数据集为斯坦福学者 Rajpurkar 等人提出的 Stanford Question Answering Dataset（SQuAD）数据集。SQuAD 数据集共有 2 个版本，第二个版本[9]在前者的基础上新增了一个无答案检测，即其中的某些文章-问题对中，该问题对

作者简介：范 玚（1996-），男，博士研究生，主要研究方向：自然语言处理、阅读理解；刘秉权（1970-），男，博士，副教授，博士生导师，主要研究方向：移动计算、Web 知识挖掘、自然语言处理等。

收稿日期：2020-07-22

应的答案在相应的文章中无法获得,此时模型应返回一个 No Answer。该设计对本实验没有作用,因此为了简化实验、突出重点,本文选择更早的 SQuAD1.1 版本作为本实验采用的主要数据集。

SQuAD1.1 数据集中,所有问题的标准答案都是原文中存在的短语,大概可分为时间、除时间外其余数字、人物、地点、常用名词短语、形容词短语、动词短语等共 10 个类别,每个类别在整体数据中所占比重见表 1。

表 1 答案类别示意

Tab. 1 Categories of SQuDA answers

Answer type	Percentage/%	Example
Date	8.9	19 October 1512
Other Numeric	10.9	12
Person	12.9	Thomas Coke
Location	4.4	Germany
Other Entity	15.3	ABC Sports
Common Noun Phrase	31.8	Property damage
Adjective Phrase	3.9	Second-largest
Verb Phrase	5.5	returned to Earth
Clause	3.7	to avoid trivialization
Other	2.7	quietly

2 现有的对抗性阅读理解数据集介绍

在 SQuAD1.1 数据集发布以后,大量的学者提出了自己的解决方案,使得这一数据集的 state-of-the-art 成绩被不断提高,其中最优秀的一些模型甚至已经超越了人工准确率。因此,有学者对此数据集进行了改进,来增加其难度,并更加全面地验证阅读理解模型的鲁棒性和其对于自然语言的真实理解程度。

主要的改进思路有 2 种. 第一种是对文章进行改进,在不扰乱文章语义结构的前提下,通过在文章中加入一些带有迷惑性的句子,来干扰模型提取答案;第二种是对问题进行改进,在不改变问题语义的前提下,仿写出多个问句,来测试模型对于不同表述的问句,是否能够一致地找到正确答案。对此拟展开研究分述如下。

2.1 基于文章扩充的方法简要介绍

在 SQuAD 数据集的基础上,同样是来自斯坦福的研究者 Jia 等人[10]提出了通过向文章中填充一些具有迷惑性的句子,来干扰阅读理解模型对于答案的选择。

该文作者首先讨论了用于扩充文章的句子应放在文章的什么位置比较合适,此后得出结论,新增句子放在文章中间最容易打乱文章的语义结构,破坏文章上下文的连贯性;而放在文章开头则会使得文章第一句不是中心句,可能会影响模型对文章的理解。综上所述,在本方法中,新增的句子将被统一放置在文章的末尾。

此方法通过改造问句和标准答案来生成被填充的语句。主要步骤可以分为以下 4 步:

首先,对问句进行改动。将问句中的形容词替换为 WordNet[11]中的反义词;再对问题中的命名实体和数字,在 GloVe[12]词向量空间中选择最相近词汇来进行替换。在本步骤中,如果问句没有产生变化,则返回原样例。

其次,制造了一个和原始答案相同类型的假答案。通过 Stanford CoreNLP 工具命名实体识别和词性标注的结果,该文作者构建了 26 种答案类别,并为每一种类别手工设计了一些假答案。当获得真答案和问题时,通过模型计算出真答案的类别,并从该类别中选出相应的假答案。

然后,将改动过后的问句和新生成的假答案转变为陈述句。为了实现这一目标,论文通过 CoreNLP 工具来对问句进行成分分析,并人工设计了超过 50 条规则,这就可以将问句和答案转化成陈述句。

最后,考虑基于规则的方法生成的陈述句很可能出现语法的错误,论文对于每个生成的陈述句,让 5 个工作人员来进行人工检测,当有超过 3 名工作人员认为这一句有语法错误时,将取消这一样例,对原文章不做修改。

该文作者使用 Match-LSTM[13]和 BiDAF[14]两种方法在新生成的数据集和原始的 SQuAD1.1 数据集上分别进行了测试。发现对比在 SQuAD 数据集上的结果,在新生成数据集上的结果均下降了超过一半。

2.2 基于问句仿写的方法改进介绍

和上一小节不同的是,Gan 等人[15]提出了通过对问句进行仿写,来检验阅读理解系统稳定性的方法。

该文提出了 2 种仿写问句的方法,并分别构建了各自的数据集,来检测阅读理解系统的过敏感性和过稳定性。其中,过敏感性使用了仅做微小变更的相同语义的问句,来对系统进行检测,判断阅读理解模型是否对于问句的微小变动过于敏感;过稳定性则在文章中找一个正确答案词性相同的短语,继

而用该短语附近的词汇来改写问句，并保持问句语义不变，以此来判断阅读理解模型是否过于依赖词汇表面意思的匹配。为此可做剖析概述如下。

2.2.1　针对过敏感性的方法

该方法主要采用了基于 Transformer[16] 的编码-解码结构，并对其中的解码器进行了修改，加入了复制机制，即考虑了从原问句中选择词汇的概率分布，使得在生成仿写的问句的时候，可以考虑加入原句中现有的词语。

此方法将仿写建议和原问句首尾相接在一起作为模型的输入。其中，仿写建议是可用来替换原问句中部分词汇的单词或词组。

该模型的训练数据来源有 2 个，分别为 WikiAnswers dataset 和 Quora dataset 这两个仿写数据集。在 WikiAnswers dataset 数据集中，原句和仿写句中的词组存在一一对应关系，故而，此方法只需从仿写句随机选择一个对应好的词组，即可获得仿写建议；对于 Quora dataset 数据集中的数据，研究中使用了 TextRank 方法来分别从原句和仿写句中获得关键词，并将仿写句中排名最高且未在原句中出现过的关键词作为仿写建议。

在预测过程中，作者使用 paraphrase database (PPDB)[17] 仿写数据库来根据问句生成仿写建议。PPDB 中包含了数以百万记的仿写词组对。作者首先从原问句中提取出所有的 1 ~ 6-grams 词组，并去除掉其中 unigram 词语中的停止词；然后在 PPDB 中为剩下的词组寻找相似度大于 0.25 的仿写词组；再将这些仿写词组分别作为仿写建议，和原句一起传入模型，获得多个仿写问句；最后，将获得的多个仿写问句和原问句进行相似度计算，在此基础上就可去除相似度小于 0.95 的仿写问句。

2.2.2　针对过稳定性的方法

为了检测模型的过稳定性，该研究从 SQuAD 1.1验证集中人工选择了一些样例，并从这些样例文章中分别选择了一个与原答案类型相一致的短语，接下来则用这个短语附近的词组对问句进行改写，保持问句语义不变。

这一过程均为人工完成，过程结束后总共编写了 56 个新的问句。

至此，研究又采用了 BERT、DrQA、BiDAF 三种模型对 SQuAD 数据集和 2 种仿写问句的数据集进行性能测试。测试结果发现相比于标准的 SQuAD 数据集，这三种方法在第一种针对过敏感性的测试集上的性能都略微下降了 2 ~ 3 个百分点，而在第二种针对过稳定性的测试机上的性能则下降了大约一半。

3　基于 Transformer 结构的对抗阅读理解数据生成方法

综上小节所述，现存的强调对抗性的阅读理解数据集大多都是依赖人工方法生成，这使得基于人工编写规则的生成结果容易出现错误，而纯手工编写的数据集过于耗时耗力，往往规模很小。本小节将介绍一种基于端到端模型的对抗阅读理解数据生成方法，使得这一过程可以摆脱人力的限制。

3.1　相关技术介绍

3.1.1　命名实体识别相关技术介绍

命名实体识别的主要任务是识别出文本中的人名、地名等专有名词，以及时间、日期、百分比等有意义的数据短语，并对其加以分类。

早期的命名实体识别大多采用的是基于启发式算法和手工编写规则的方法[18]，由于这一任务本身是一个标注任务，很多专有名词或数据都较容易识别，并且原先的数据集对于实体划分的种类较少，使得这些比较朴素的方法也表现出较好的性能。随着机器学习技术的兴起，有学者相继提出了基于隐含马尔可夫模型的方法[19]、基于条件随机场的方法[20]等一系列基于学习的方法，进一步提高了命名实体任务的准确率。

本文中所使用的是斯坦福大学提供的 Stanza 工具[21]。这是基于深度学习的方法。该方法先是通过字符级的长短时记忆网络来学习每个单词的向量表示，并将其和对应单词的词嵌入相连接，再传入一层双向长短时网络标注器，最终使用条件随机场方法来对标注结果进行解码。

本文采用的命名实体分类标准采用的是 OntoNotes5.0 数据集中规定的标准，共包含 11 类名称和 7 类数据，所有 18 种分类见表 2。

3.1.2　Transformer 结构简介

Transformer 是由 Ashish Vaswani 等人提出的 Seq2Seq 模型。与之前的模型不同，Transformer 完全依赖于注意力机制来挖掘文本中的上下文联系，而没有采用循环神经网络结构及其变体。Transformer 模型的结构如图 1 所示。

Transformer 的编码器由 2 个子层组成。其中，一个子层是多头注意力机制层。和普通的注意力机制相比，多头注意力机制将<Q，K，V>三元组进行多次映射，并在每次映射后分别进行注意力机制的运算，再将多个结果首尾相连，如此则可获得多头注意

力机制的结果。另一个子层是一个前馈神经网络。每个子层之后都应用了残差连接和层归一化。

表 2 命名实体分类示意表

Tab. 2 Categories of named entities

分类	含义
PERSON	People, including fictional
NORP	Nationalities orreligious or political groups
FACLITY	Buildings, airports, highways, bridges, etc.
ORGANIZATION	Companies, agencies, institutions, etc.
GPE	Countries, cities, states
LOCATION	Non - GPE locations, mountain ranges, bodies of water
PRODUCT	Vehicles, weapons, foods, etc. (Not services)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK OF ART	Titles of books, songs, etc.
LAW	Named documents made into laws
LANGUAGE	Any named language
DATE	Absolute or relative dates or periods
TIME	Times smaller than a day
PERCENT	Percentage (including "%")
MONEY	Monetary values, including unit
QUANTITY	Measurements, as of weight or distance
ORDINAL	"first", "second"
CARDINAL	Numerals that do not fall under another type

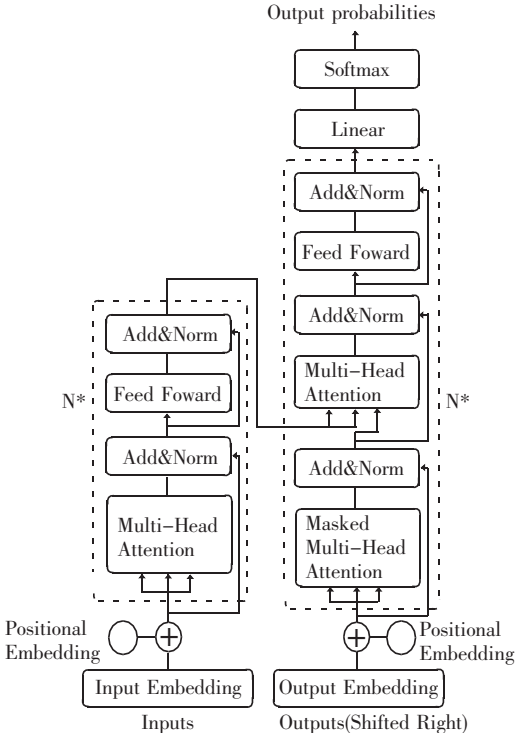

图 1 Transformer 模型结构示意图

Fig. 1 Architecture of Transformer

Transformer 的解码器和编码器结构相似,最大的区别在于解码器多了一个子层。该子层的作用是对编码器的输出使用多头注意力机制进行处理。

值得一提的是,由于没有采用类似于循环神经网络的时序模型,为了保存文本中的上下文顺序关系,Transformer 采用了位置嵌入。

本文将使用 Transformer 作为生成对抗数据的基本结构。另外,由于本任务的目的只是将疑问句转换为陈述句序,为了让模型在生成语句的同时会考虑输入中原有的词汇,本文在 Transformer 解码器中引入了复制机制。模型具体结构将在下文中予以详细解释。

3.2 模型整体结构

和大量现有的工作不同,本实验将采用一种不需要人工介入的生成干扰句的方法,此方法主要分为 2 个模块,即:问句改写和虚假答案生成模块、基于端到端结构的干扰句生成模块。相关的研究论述详见如下。

3.2.1 问句改写和虚假答案生成模块

本模块的目的,是使用原始数据中的问题和标准答案,生成一个与原意不同但形式类似的问句,以及一个和标准答案类型相同的虚假答案。

首先,本文使用上文提到的 Stanza 工具对 SQuAD 开发集中的所有段落进行命名实体识别,并将获得的所有不重复的实体按照其本身的类别分别存储,以此来构建备用的分类实体库。

然后,同样对原问句进行命名实体识别,对于其中识别出来的实体,随机地从此前获得的分类实体库中按照类别选择实体来进行替换。如果最终问句没有变动,则不对这一样例进行处理,返回原样例。

最后,对于原答案进行分类,判定其所属的实体类别,并从分类实体库中选择相同类别的实体作为虚假答案。如果原答案分类不成功,则不对该样例进行处理,返回原样例。

问句改写和虚假答案生成模块的结构如图 2 所示。

3.2.2 基于 Transformer 结构的干扰句生成模块

本模块使用基于 Transformer 框架的端到端结构来进行干扰句生成。模块的输入是 3.2.1 节中生成的改写后的问句和虚假答案,首尾相接并用[Sep]标志隔开,模块的输出是陈述句形式的干扰句。由于新生成的干扰句中必然包含输入中的很多词语,因此本文改写了 Transformer 框架的解码器结构,在其中加入了复制机制,即在生成每一个词语

时，会考虑从输入中选择一个词语的可能性。

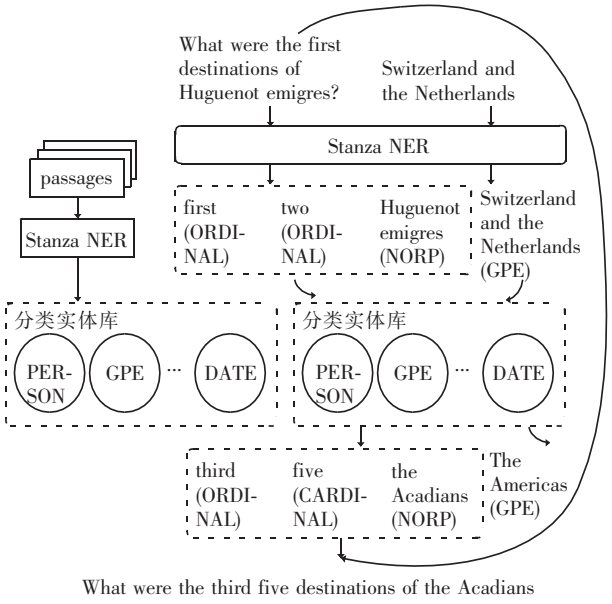

图 2　问句改写和虚假答案生成模块示意图

Fig. 2　Question rewriting and fake answer generation system

基于 Transformer 结构的干扰句生成模块结构如图 3 所示。

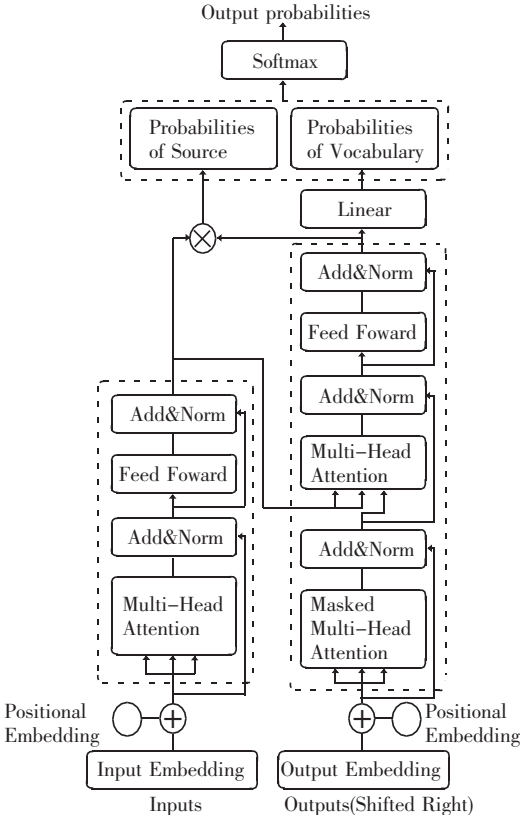

图 3　基于 Transformer 结构的干扰句生成示意图

Fig. 3　Interference sentence generation system based on Transformer

由图 3 可知，和传统的 Transformer 结构相比，本文还利用 Encoder 的输出和 Decoder 的输出计算了从输入中复制词语的概率。对于位于词汇表中却不在输入中的词语，其概率即为生成概率；对于输入中的词语，其概率等于生成概率与复制概率相加。具体公式如式(1)～(3)所示：

$$P(y_t) = P(y_t, g) + P(y_t, c), \tag{1}$$

$$P(y_t, g) = \begin{cases} \dfrac{1}{Z} e^{f_g(y_t)}, & y_t \in V; \\ 0, & y_t \in X \cap \overline{V}; \\ \dfrac{1}{Z} e^{f_g(UNK)}, & y_t \in \overline{X} \cap \overline{V}. \end{cases} \tag{2}$$

$$P(y_t, c) = \begin{cases} \dfrac{1}{Z} \sum_{j: x_j = y_t} e^{f_c(x_j)}, & y_t \in X, \\ 0, & y_t \in \overline{X}. \end{cases} \tag{3}$$

其中，$P(y_t)$ 为第 t 个生成的单词为 y_t 的概率；$P(y_t, g)$ 和 $P(y_t, c)$ 分别为从词汇表中生成单词 y_t 的概率，以及从输入中复制单词 y_t 的概率；V 表示词汇表所代表的集合；X 表示输入中所有单词所代表的集合；$f_g(y_t)$ 和 $f_c(y_t)$ 分别表示单词 y_t 在生成过程中和复制过程中的得分；Z 表示正则化的参数，在复制和生成过程中通用。$f_g(y_t)$ 的计算公式可写为：

$$f_g(y_t = v_i) = v_i^{\mathrm{T}} \boldsymbol{W}_g S_t \quad v_i \in V \cup \{UNK\}, \tag{4}$$

其中，v_i 为词汇表中第 i 个单词的 one-hot 向量；$\boldsymbol{W}_g$ 为 $(|V|+1) * d_s$ 维的参数矩阵；S_t 为 t 时刻 decoder 的输出。

$f_c(y_t)$ 的计算需用到如下数学公式：

$$f_c(y_t = x_i) = \sigma(\boldsymbol{h}_j^{\mathrm{T}} \boldsymbol{W}_c) S_t \quad x_j \in X, \tag{5}$$

其中，x_j 为输入中第 j 个单词的 decoder 输出；$\boldsymbol{W}_c$ 为 $d_h * d_s$ 维的参数矩阵；σ 为非线性变换。

Z 的计算公式具体如下：

$$Z = \sum_{v \in V \cup \{UNK\}} e^{f_g(v)} + \sum_{x \in X} e^{f_c(x)}. \tag{6}$$

本实验使用的是在斯坦福的 Jia 和 Liang 两位学者提出的 AddOneSent 数据集中，提取出的 300 条数据作为本次研究的训练数据。

研究中，先从该数据集中提取出(文章、问题、答案)三元组，接着选择文章中的最后一句话，若其不是生成的干扰句，舍去这一样例，否则，将根据这一干扰句，人工生成改写的问题和虚假答案，并将改写的问题和虚假答案作为训练集样本，干扰句作为样本的标签。

4 实验结果与分析

4.1 问句改写和虚假答案生成的实验结果

在此模块中,本文先是利用 SQuAD 开发集中的全部文档和命名实体识别方法提取出了不同类别的命名实体,以此构建了分类实体库,用来进行后续的替换操作。由于 WORK OF ART 和 LAW 两个种类中的实体较少,所以在构建分类实体库时,本文将这两类删去。该分类实体库的构建结果见表 3。

然后,本文利用分类实体库,对问句进行了改写。

最后,根据真实答案的类别,本文将从分类实体库中选择相应的实体进行替换,进而生成虚假答案。问句改写和虚假答案生成的样例见表 4。

4.2 基于 Transformer 结构的干扰句生成实验结果

在此模块中,本文采用了 Transformer 结构,并引入了复制机制,实现了由改写问句和虚假答案生成干扰句的功能。本实验中生成的部分样例与现有人工方法生成的干扰句样例比较见表 5。

表 3 分类实体库构建结果

Tab. 3 Results of classified entity libraries

分类	样例
PERSON	Arizona Cardinals, Cam Newton
NORP	American, Canadian
FACLITY	Children´s Memorial Health Institute, UniversityLibrary garden
ORGANIZATION	the National Football League, American Institute of Electrical Engineers
GPE	Southern California, the United States
LOCATION	the Mississippi River, the Carpathian Mountains
PRODUCT	Amazonia:Man and Culture in a Counterfeit Paradise
EVENT	Hurricane Floyd, World War II
LANGUAGE	the German language, English
DATE	20 December 1914, 14 October 1904
TIME	9:00 a. m. ,eight o´clock
PERCENT	85%, 30%
MONEY	fifty thousand dollars, one million dollars
QUANTITY	138 metres, 50 kilometers
ORDINAL	first, second
CARDINAL	850, 103

表 4 问句改写和虚假答案生成的样例

Tab. 4 Examples of question rewriting and fake answer generation

原问句	改写问句	原答案	虚假答案
What is the highest reference hospital in all of Poland?	What is the highest reference hotel in all of Russia?	Children´s Memorial Health Institute	University Library garden
What were the first two destinations of Huguenot emigres?	What were the third five destinations of Huguenot the Acadians?	Switzerland and the Netherland	The America

表 5 生成干扰句与人工方法对比

Tab. 5 Comparison of generated interference sentence and handwork

The proposed Model	AddOneSent
Cam Newton won Champ Bowl 20.	The Broncos won Champ Bowl 40.
Polonia Warsaw was founded in year 1865.	The Michigan Vikings franchise was founded in the year 1970.
Newton is Canadian numeral for 65.	LSTM is the Byzantine numeral for 40

由表 5 中可以看出,本文中的结构所生成的大部分干扰句都能保证语义正确、成分完整,整体性能较好。对于一小部分语句可能出现少量的语义错误和前后不连贯的问题,但这并不会影响到测试模型抗干扰能力的功能。

最后,本文采用较为主流的 3 种机器阅读理解模型,在本文中生成的对抗性阅读理解数据集(ADV)进行实验,试验结果见表 6。

表 6 主流机器阅读理解模型的性能表现

Tab. 6 Performance of popular MRC model on the proposed dataset

模型	SQuAD F_1	ADV F_1	AddOneSent F_1
Match-LSTM	71.2	38.9	39.2
BiDAF	75.4	46.8	47.6
BERT	87.8	56.6	59.1

由表 6 的结果中可以看出,这三种主流机器阅读理解模型在生成的对抗数据集上的结果都出现了较大程度下降,并和现有的人工生成的 AddOneSent 数据集上的表现相近。这说明在测试机器阅读理解模型抗干扰能力上,本文使用机器自动生成的数据集和现有的人工生成的数据集有相似的表现。

5 结束语

传统的机器阅读理解数据集难度较低,涉及的很多问题仅通过答案的类型和词汇相似度的计算就可以得到正确答案,因此这样的数据集无法真正、全

面、深入地评价一个机器阅读理解模型对于文章的理解程度。

近年来，多位学者陆续提出了不同的方法，来对传统阅读理解数据集进行改进，以增加其难度和对抗性，从而更好地评价机器阅读理解模型对于文章的理解程度。然而，由于对抗数据生成的难度较大，现有的方法往往都大量地借助了人工编写的方法。只是这些方法耗时耗力，极大地限制了对抗数据集的规模。针对这一状况，另有学者则提出了人工编写规则来生成数据的方法，但是大量的编写规则不仅繁琐，而且也不能保证性能，生成的结果仍然需要依靠人工来进行筛选。基于此，本文提出了一种基于 Transformer 结构的对抗数据生成方法。将生成对抗数据的过程分为了问句改写及虚假答案生成和干扰句生成两个模块。

首先，为了更加方便地进行问句改写和虚假答案生成，本文利用命名实体识别技术从 SQuAD 开发集的全部文章中抽取出了分类实体库，并以此对问句中的实体根据类别进行随机替换。同样地，也对真实答案根据其类别标签从分类实体库中选择相应的实体进行了替换。

然后，本实验利用现有的对抗数据集 AddOneSent 中的数据，人工生成了训练数据，这也是本实验中仅有的一个利用人工完成的部分。这一模块中，本文在 Transformer 结构的基础上，在其解码器上拓展了复制机制，使其在生成干扰句的同时可以从改写过后的问句和虚假答案中选择词汇。

最后，本文使用现有的 3 种主流机器阅读理解模型对生成的对抗数据集进行测试。通过测试结果看出，本实验中机器自动生成的数据集可以和人工手写的数据集相似地检测机器阅读理解模型的鲁棒性，并能以此来评价其对于文章真正的理解程度。

参考文献

[1] DEVLIN J, CHANG Mingwei, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805,2018.

[2] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations [C]//North American Chapter of the Association for Computational Linguistics. Louisiana, USA:NAACL, 2018: 2227-2237.

[3] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training [EB/OL]. [2018]. https://s3-us-west-2. amazonaws. com/openai - assets/researchcovers/languageunsupervised/language understanding paper. pdf.

[4] YANG Z L, DAI Z, YANG Y M, et al. XLNet: Generalized autoregressive pretraining for language understanding [C]// Advances in Neural Information Processing Systems. Vancouver: NIPS foundation,2019: 5753-5763.

[5] LAN Z, CHEN M, GOODMAN S, et al. Albert: A lite bert for self-supervised learning of language representations[J]. arXiv preprint arXiv:1909.11942, 2019.

[6] RAJPURKAR P, ZHANG J, LOPYREV K, et al. Squad: 100, 000+ questions for machine comprehension of text[J]. arXiv preprint arXiv:1606.05250, 2016.

[7] SEN P, SAFFARI A. What do models learn from question answering datasets? [J]. arXiv preprint arXiv: 2004. 03490, 2020.

[8] WEISSENBORN D, WIESE G, SEIFFE L. Making neural qa as simple as possible but not simpler[J]. arXiv preprint arXiv: 1703.04816, 2017.

[9] RAJPURKAR P, JIA R, LIANG P, et al. Know what you don't know: Unanswerable questions for SQuAD[C]//Meeting of the Association for Computational Linguistics. Melbourne, Australia: ACL, 2018: 784-789.

[10] JIA R, LIANG P. Adversarial examples for evaluating reading comprehension systems[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark:ACL, 2017: 2021 - 2031.

[11] MILLER G A. WordNet: A lexical database for English[J]. Communications of the ACM,1995,38(11): 39-41.

[12] PENNINGTON J, SOCHER R, MANNING C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar:dblp,2014:1532-1543.

[13] WANG S, JIANG J. Machine comprehension using match-lstm and answer pointer[J]. arXiv preprint arXiv:1608.07905, 2016.

[14] SEO M, KEMBHAVI A, FARHADI A, et al. Bidirectional attention flow for machine comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.

[15] GAN W C, NG H T. Improving the robustness of question answering systems to question paraphrasing[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy:ACL,2019:6065-6075.

[16] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]// Advances in Neural Information Processing Systems. Long Beach:NIPS, 2017:5998-6008.

[17] GANITKEVITCH J, VAN DURME B, CALLISON-BURCH C. PPDB: The paraphrase database[C]//Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Atlanta:IEEE ,2013: 758-764.

[18] RAU L F. Extracting company names from text[C]//Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications. Los Alamitos:IEEE,1991: 29-32.

[19] BIKEL D M, SCHWARTZ R, WEISCHEDEL R M. An algorithm that learns what's in a name[J]. Machine Learning, 1999, 34(1-3): 211-231.

[20] LIAO Wenhui, VEERAMACHANENI S. A simple semi-supervised algorithm for Named Entity Recognition [C]// Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing. Boulder, Colorado:ACL,2009: 58-65.

[21] PENG Qi, ZHANG Yuhao, ZHANG Yuhui, et al. Stanza: A Python natural language processing toolkit for many human languages[J]. arXiv preprint arXiv:2003.07082,2020.