文章编号:1671-4598(2024)02-0001-06 DOI:10.16526/j.cnki.11-4762/tp.2024.02.001 中图分类号:TN319 文献标识码:A

典型测试用例推荐与用例期望结果生成系统

邓佳棋，王月波，蒲卿路，李继秀，杨 旭
（西南电子技术研究所，成都 610036）

摘要：测试领域在对公司产品质量的把控中至关重要，但不同测试人员在面向同一个测试产品时，会由于测试人员本身缺少经验以及对产品功能的不熟悉，导致测试不具备系统性和全面性；同时测试团队在长时间针对公司相关业务产品，或者开发人员在系统联试过程中，会形成大量具有典型意义的测试用例；但目前传统的做法并没有将该具有价值的测试用例深度分析；只是形成相关文档后汇总，就将该数据尘封；因此文章以典型测试用例为数据，以知识图谱为展现形式与存储形式，Bert 实体提取为技术基础的推荐系统；根据用户输入，推荐出相关的典型测试用例；同时在某些行业，测试人员在实际工作业务中，需要花费大量时间对测试用例的输入和期望结果进行描述，形成正式文档用于保存记录；该系统可实现根据用例输入自动生成对应的期望结果，以提升测试人员的工作效率。

关键词：典型测试用例；Bert；知识图谱；实体提取；文本生成

Typical Test Case Recommendation and Expected Result Generation System

DENG Jiaqi，WANG Yuebo，PU Qinglu，LI Jixiu，YANG Xu
（Southwest China Institute of Electronic Technology，Chengdu 610036，China）

Abstract：It is essential for testing field in controlling the product quality of the companies. However，when different testers face the same test product，the test results may not be systematic and comprehensive due to their own experience and familiarity with product functions. At the same time，the test team of the system will generate many typical test cases during the long-term test of the related business products or joint testing process of developers. However，traditional methods do not provide the in-depth analysis of these valuable test cases，only collect the relevant documents and seal up the data for keeping. Therefore，typical test cases are taken as data，knowledge graph as presentation form and storage form，and Bert entity extraction as the technical basis of the recommendation system. The related typical test cases based on user input are recommended. At the same time，it takes a lot of time for testers to describe the input of test cases and expected results in the actual work and business of some industries，and form the formal documents for saving records. This system can automatically generate the corresponding expected results based on the input of use cases to improve the work efficiency of testers.

Keywords：typical test case；Bert；knowledge graph；entity extraction；text generation

0 引言

一个公司企业经过长时间的积累、沉淀，会产生大量的典型测试用例。通过对测试用例的整理、积累、学习，可以帮助和解决各种将来可能发生的问题，规避风险，获取经验教训，更能够帮助对质量流程的控制，促进持续改进，提高整体管理者和员工的工作水平，使得公司产品更加完善、全面。

目前大部分公司关于测试用例的管控与运用，都处于阅读非结构化文本的形式。或者各公司可能会定期开展培训宣传工作。但该措施也是有很大局限性的，没有一个完整的系统性的、全流程的管控，只是起了一个临时统计汇报交流的作用。新员工很难从这些零散的数据中进行学习，或者项目与项目之间无法有效共享交流各自曾经发生过的问题，从而总结出一些共性通用的问题。最为核心的问题在于有价值的数据在长时间的迭代下，被抛弃或者淡化。针对上述问题，可以使用智能推荐去改善当前关于典型测试用例的运用场景。最早的推荐系统使用基于协同过滤的推荐技术，其核心思想是以用户历史的选择记录与偏好作为基础，推荐与历史选择关联性较高且未被记录的内容。后期深度神经网络开始迅速发展，其在个性化推荐的使用场景中越来越明显，例如 Adams 等人在 youTube 视频网站中则使用 DNN 模型推荐用户感兴趣的视频。但是目前主流的智能推荐的使用场景都局限于吃喝玩乐、美食推荐、电影推荐及社交推荐中，与日常工作相关的场景较少，尤其是当工作内容与外界隔离的情况。

目前测试行业中，测试的主要目的是发现产品的质量缺陷，但测试的全面性、有效性还是较为依赖测试人员的

收稿日期:2023-06-22； **修回日期**:2023-07-25。

作者简介:邓佳棋(1988-),男,硕士,高级工程师。

引用格式:邓佳棋,王月波,蒲卿路,等.典型测试用例推荐与用例期望结果生成系统[J].计算机测量与控制,2024,32(2):1-6.

专业水平。除相关培训外，需要一个有系统性，针对产品相关性强的辅助系统以推荐的形式帮助测试人员提高测试用例的质量以及全面性。目前推荐算法层出不穷。但该算法在测试用例等具有专业性的领域中应用还是较少。其主要原因还是在于典型测试用例的数据集不充分，测试用例的文本质量参差不齐。因为，最具有代表性的外场问题，或者用户反馈往往不是由专业的测试人员编写、总结，而是由产品的设计、开发人员编写。对问题进行简要总结后形成 word，excel 等非结构化的数据，并且相关人员对其描述存在大量专业术语，后期维护时无法对问题的本质进行总结，从而导致无法形成高质量的训练集或者数据源。

测试用例由测试步骤构成，测试步骤由输入、期望结果及实测结果组成。测试人员需要投入大量精力去填写测试输入、期望结果及实测结果。测试步骤的输入是该测试用例的关键，但期望结果完全可根据输入及相关背景自动生成。因此需要一个能够实现自动生成期望结果的算法，从而减少测试人员编写测试文档的时间。并将其主要关注点集中在输入的设计上。

综合上述描述，关于测试用例在项目全周期中的管控，以推荐测试用例为应用背景的推荐算法和测试人员在相关文档编写中存在以下几个方面的问题：

1）团队或者新员工没有将具有宝贵经验的典型测试用例应用在实际工作中。

2）针对测试用例推荐算法的训练数据集少，且非结构化，需要大量人力、物力去标注后，才能用于相关算法的训练。导致已有数据难以发挥真正的价值。

3）使用场景脱节，目前市面没有一个具有典型、代表性的针对测试用例的推荐系统，大部分场景还是针对消费市场。

4）测试文档编写和非关键内容的编写，增加了测试人员的工作量。

因此基于以上问题及背景，本文采用知识图谱和 Bert 模型为技术基础，推荐与真实测试场景相关的典型测试用例以及自动生成用例步骤期望为目的，帮助测试人员和开发人员发现与解决项目中存在的潜在问题并降低工作量，提出测试用例推荐与用例期望结果生成系统。

1 系统总体构成

1.1 系统方案结构

该系统采用 B/S 架构，前端由 Vue 开发，后端框架选用 Django。该架构在后端中集成知识图谱数据库（Neo4j）和 Mysql 数据库，基于 Bert 实体提取、AC 状态机及 Seq2Seq 文本生成等算法为一体，能够实现对用户的输入，即查询语句，进行相关测试用例的推荐与生成。系统整体方案如图 1 所示。

本文将主要介绍有关知识图谱及实体提取算法相关的部分。

1.2 典型测试用例定义

典型测试用例是测试团队与开发团队在产品开发与测试时，发现的在某一类产品中具有代表性的测试用例，通常由问题描述、原因分析及解决措施组成。该用例需要经过公司内部评审分析其关键程度与价值，判定它是否能够对其他测试具有启示意义等。因此典型测试用例具有深刻含义与推广价值，需要仔细分析并结合用户的输入对其进行推荐。

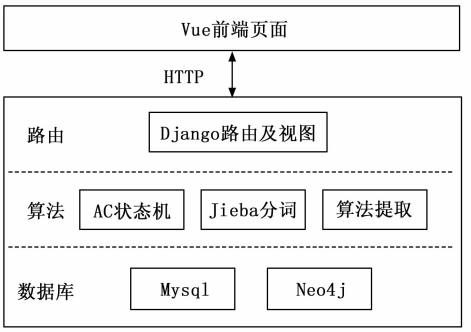

图 1 系统方案结构图

但作为测试用例推荐系统，需要为每一个测试用例总结一句具有代表性的标题，才能在推荐系统中进行展示。目前，典型测试用例接近 4 000 条。为每一个测试用例进行总结十分浪费人力、物力。因此在这里，使用了百度公开的（文本生成）text-generation 算法去生成测试用例的标题。算法的输入即测试用例的现象、原因及解决措施拼接后的字符串，输出即可总结上述文本的语句。因为上述算法是使用公开训练集，所以输出可能会出现错误，甚至出现语句不通的情况，因此生成后还需要再经过一遍筛选。但这种筛选效率很快，不需要关注内容本身，只需要关注输出标题是否存在低级错误。

1.3 测试用例的知识图谱简介与图谱的构建

知识图谱（Knowledge Graph）是一种用于表示知识的语义网络[1]，它描述了事物之间的关系[2]，通常是由一组实体和他们之间的关系来描述某个特定领域的知识[3]。结合本文的目的，系统中的知识图谱由实体、属性、概念、表述和关系组成。实体就是由测试用例标题提取出能够作为主语的词语，在本系统中通常为设备名、软件名等。在后续标注训练样本时，实体类型大致可分为设备、软件、时间、地点及单位等。概念即测试用例发现缺陷的严重等级等，一些可对测试用例进行分类的名词。表述即为测试用例发生时间等。关系则是实体与表述之间，实体与概念之间的关系，例如与表述（2023 年 2 月 10 日）的关系则为发生时间，与概念之间的关系则为“是”，例如该测试用例的严重程度“是”一般。有了上述基本概念后即可在 Neo4j 中建立测试用例的完整知识图谱。

知识图谱通常有自底向上和自顶向下两种构建方法[4]。结合公司实际情况和现有数据，采用自顶向下的方式，由测试团队与开发团队专家制定测试用例统计的维度。目前现有的测试用例数据集在统计时存在以下维度：测试方法、测试输入类型、问题现象、原因分析、解决措施、问题严

重等级、发生时间、问题引入阶段、产品名称、产品规模及运行环境等。上述纬度同时也作为知识图谱中用于匹配查找的关系，而使用该维度对应的内容则作为推荐的内容。

最终将每一个测试用例按照上述维度形成 excel 后，使用 pandas 解析[5]，并结合 cypher 语句，导入 Neo4j 数据库中，以供后续对测试用例推荐进行使用。导入后的局部知识图谱如图 2 所示。

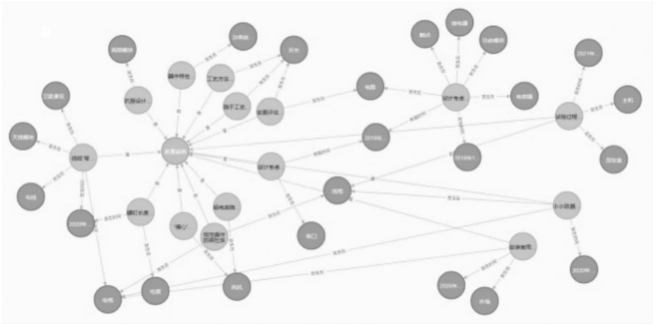

图 2　测试用例的知识图谱

导入至 Neo4j 的同时，上述信息也会同步存储至 Mysql 数据库中。在 Mysql 中创建测试用例表。使用上述的标题作为该测试用例的名称。并使用测试方法，运行环境等，可以对测试用例进行明确分类的属性作为测试用例表的外键。这样，Mysql 数据库与 Neo4j 数据库，都将同时保存着测试用例的详细内容。都可以用于推荐算法的数据来源。

2　测试用例推荐算法及系统相关功能介绍

2.1　基于 Mysql 的推荐算法

传统的 Mysql 推荐算法，本系统将采用两种方法进行案例推荐：1）使用标签及分类进行推荐；2）基于 Jieba 分词在数据库中进行匹配从而进行推荐。

1）因为在 Mysql 表建立的过程中，已经对目前数据库中已有的测试用例进行了分类，可以通过外键关联的形式直接查询。用户可以根据真实场景中相关的分类，在数据库中查看是否具有自己感兴趣的测试用例。

2）根据用户的输入，使用 Jieba 分词，对输入进行拆分。例如“大型运输机航电系统发生过什么缺陷?”上述语句在经过分词后，会得出以下内容：“大型”“运输机”“航电系统”“发生”“过”“什么”“缺陷”。将上述分词后的词语，使用 Mysql 的 like 语句与测试用例的名称进行匹配。最后再统计数据库中测试用例的名称中包含上述词语的个数。并按照数量由高至低返回推荐结果。上述方法匹配出的测试用例包含了用户输入的词语，因此推荐的内容肯定是用户感兴趣的内容。但是该方法存在一个问题，因为“发生”“过”“缺陷”等词语，会经常出现在用户的搜索以及测试用例名称中。但该类词语对推荐用户感兴趣的测试用例没有实质的用处，只是语句中需要该类词语的出现，才能形成一个正确的语句。因此由于该类词语的存在，会对统计个数带来较大的影响。为解决上述问题，在算法中会维护一个全局通用词语字典，该字典会在系统运行前进行初始化，先使用 jieba 分词对数据库中测试用例名称进行一次分词。统计词语出现的个数，并将排列靠前的词语加入该字典树中。后面针对用户的输入，分词后会过滤包含于字典树中的词语。这样，将只使用“大型”“运输机”“航电系统”等具有代表性的词语进行匹配，提高推荐结果的质量。

基于 Mysql 的推荐算法只会在基于知识图谱及实体提取算法推荐内容较少或者没有找到时，才会去调用。这样，即可保证该推荐系统在用户输入冷门语句或者系统不存在类似的测试用例时才会去调用。

2.2　基于 Bert 实体提取与迁移学习的测试用例推荐算法

Jieba 分词是基于字典匹配和规则处理的方式，在前缀词字典中寻找匹配[6]。如果只使用 Jieba 分词对输入句子进行处理，若前缀词字典中无法找到明确语义的单词，分词效果很差，很多专有名词或行业术语会无法识别。因此需要更智能的方法对测试用例进行实体提取和处理，降低对固定单词字典的依赖。

本文选用 BERT 模型对输入语句进行处理，并在其基础上引入 BiLSTM＋CRF 模块，从而提高对专业领域单词实体识别的准确性。模型结构如图 3 所示。

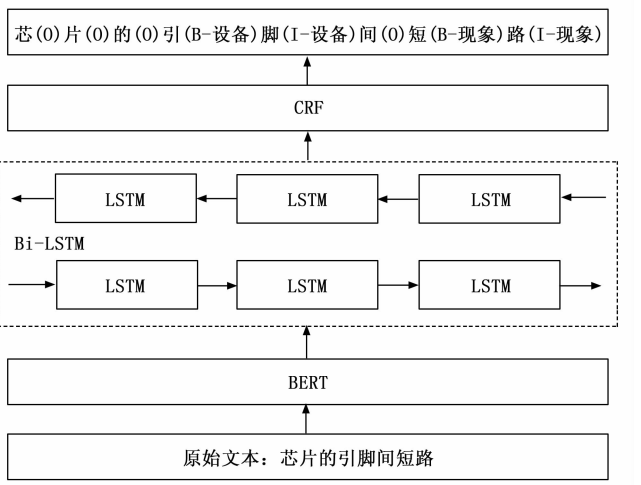

图 3　模型简图

2.2.1　BERT

Bert 是一个基于 Transformer 的语言模型，它能生成高质量的词向量特征[7]。相对于传统单向编码，Bert 能利用语境信息进行双向编码[8]。

Bert 由多层 Transformer 的 Encode 模块构成堆叠而成[9]。Bert 的嵌入层有多种信息，如 Token 嵌入层是单词本身的词向量层，该层将输入语句向量化为固定维度的词向量数据，描述了语句中的单词文本信息[10]。Segment 嵌入层则是对不同输入的 Token 进行区分，Segment 嵌入层的引入帮助 Bert 区分成对的输入序列，此外还有 Position 嵌入层描述了单词向量位置信息，通过对单词进行位置编码，让 Bert 模型能够理解同一个输入序列中的不同位置的

相同单词。

2.2.2 BiLSTM

利用 BERT 生成高质量的词向量表征后，然后引入 BiLSTM+CRF 实现实体提取。BiLSTM 是一种双向循环神经网络结构，他利用 LSTM 对输入序列进行双向编码[11]，从而识别和预测序列中的关键信息。对本文而言，其预测序列即利用 BERT 处理输入语句后的向量序列。与传统 LSTM 不同，BiLSTM 具有双向编码能力，如输入语句“这朵花开得很绚烂”，单词“绚烂”修饰的是前文的实体“花”，传统 LSTM 对语句建模时只能从前往后编码建模，无法感知后面的信息对前文的影响。而 BiLSTM 则引入了双向 LSTM 结构，它采用了前向和后向两个 LSTM 网络，并且完全独立，互相不连通[12]。每个网络接收的输入，不仅包含当前时间步的输入，还包含了另一个方向上的历史信息，这样，在每个时间步的输出向量中，都包含了当前状态下前面和后面的上下文信息，进一步加强了对序列的建模能力，BiLSTM 对长输入具有更好的鲁棒性[13]。

2.2.3 CRF

在处理测试用例数据时，发现很多特征实体具有强关联性，比如某种器件的缺陷案例数据集中，“引脚”实体与“虚焊”“断裂”“焊接工艺”等实体具有强关联性，这是描述现象的测试用例数据集的固有特点。数据集中的各种实体语义互相之间有关联，特定对象的属性特征虽然可能无法穷举，但它与另一个不同对象的属性特征一定有所差异，比如一种疾病实体的特征描述与另外一种疾病实体，甚至另一个类型的对象实体的特征描述，是一定有差异的。

为了更好地捕获特征标签之间的相互依赖关系，本文结合 CRF 层与 BiLSTM 共同进行实体识别。CRF 是一种判别式模型，可以在给定输入序列的情况下，找到最优输出标签序列，它通过添加约束条件保证预测标签的有效性[14]。CRF 由两部分组成：特征函数与权重，特征函数表示输入序列与标签序列的局部特征，权重表示每个特征函数的重要性[14-15]。CRF 的学习过程就是将训练数据中的特征函数权重进行学习，并最大化条件概率，寻找最优输出标签序列[16]。对某些复杂的句子结构和数据集相似标签标注有个别差异时，CRF 可以通过考虑相邻单词之间的关系，以及不同类别之间的转移概率，使模型更加准确和稳健。

Bert 整个模型由输入层、编码层和输出层构成。其中输入为 $\boldsymbol{E}=\{e_1, e_2, \cdots, e_n\}$ 向量，由 3 个嵌入特征叠加而成，3 个嵌入特征分别为字符嵌入 e_i^t、句子嵌入 e_i^s、位置嵌入 e_i^p。向量经过多个双向 Transformer 编码器获得含有丰富语义特征的向量 $\boldsymbol{T}=\{t_1, t_2, \cdots, t_n\}$，向量 $\boldsymbol{T}$ 作为 BiLSTM 层的输入。

在 BiLSTM 中，针对每一个词语，其输出是独立的，无法学习到输出标签之间的关系，因此数据在经过 BiLSTM 之后使用 CRF 去解决上述问题。CRF 模型可以对隐含状态进行建模，学习到标签与标签之间的关系，进一步提升模型预测的准确性。

BiLSTM 层输出与之对应的隐式状态序列 $H=\{h_1, h_2, \cdots, h_n\}$。为了获取全局最优的标签序列，基于 CRF 层考虑标签之间的关系，保证预测标签的合理性。对于序列 $X=(x_1, x_2, \cdots, x_n)$，其输出标签序列 $Y=(y_1, y_2, \cdots, y_n)$。其计算分数函数 $S(X, Y)$ 的公式为：

$$S(X,Y)=\sum_{i=1}^{n}P_{i,y_i}+\sum_{i=0}^{n}A_{y_i,y_{i+1}} \tag{1}$$

式中，$A_{y_i,y_{i+1}}$ 表示由标签 y_i 转移到 y_{i+1} 的概率，P_{i,y_i} 表示第 i 个词预测为 y_i 个标签的分数。CRF 层通过接受 BiLSTM 层的隐式状态作为输入，通过学习标签间的约束条件提升标签预测的准确性，从而得到最终的预测标签。

2.2.4 训练集制作及迁移学习

搭建模型后需要制作训练集对模型进行训练。训练集则使用了在知识图谱准备的过程中的数据。使用标注工具、将现象、原因分析及解决措施的文本进行标注。本次标注主要的关注内容为：设备、软件、发生时机、故障地点及问题类型等。但对于传统 NLP 模型来说，由 4 000 条测试用例制作的训练集还是较少。并且 4 000 条数据中，只有 3 000条数据用以训练，剩余 1 000 条数据，还需要制作成测试集。如果仅仅使用该数据进行训练，则模型会由于样本规模太小导致训练效果差，提取的实体质量低。因此针对训练集较少的情况，提前使用了谷歌官方已经提供了 BERT 的中文预训练模型。并在此基础模型上进行迁移学习，提升测试用例中实体提取的质量。

2.3 推荐流程

在实体提取与知识图谱都已经构建完毕后，可以基于上述内容，在图数据库中进行查找与反馈。整个推荐的流程如图 4 所示。

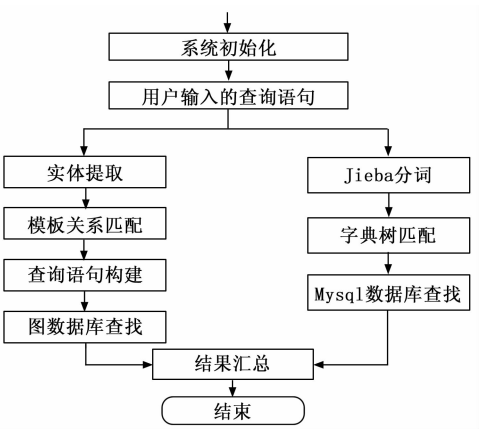

图 4 推荐流程

Mysql 的推荐算法在上述章节已经介绍，因此只针对 Bert 实体提取即知识图谱查找的流程进行详解。

1）初始化。将图数据库即测试用例名称通过模型提取实体，并形成 AC 状态机，记录所有实体经过 Bert 层输出的词向量，存储在内存中，以便提升效率不再需要时进行二次计算。

2）获取用户输入后，使用基于 Bert+BiLSTM+CRF 的模型提取用户输入语句中的实体。

3）因为提取出来的实体并不能与图数据库中的实体完全匹配，因此需要使用相似度计算，找到图中最接近的实体。

4）由于知识图谱是自顶向下构建，所以实体与实体的关系都已经确定。实体与关系的组合即组成查询知识图谱的 Cypher 语句。

最终执行 Cypher 语句得出结果，如果没有找到结果，则会依据 Msql 的推荐算法给出答案。

可以看出在使用算法提取实体后，最主要的难点在于相似度计算。系统中维护的实体数毕竟有限，但用户的输入却有无数种可能。因此为计算相似度，本系统综合使用以下 3 种相似度查找策略，求取系统中最接近的实体。

1）AC 状态机：相似问题，首先是解决字符串包含关系。例如系统中已经存在“vxworks”实体，那么用户在输入“vxworks 系统”后，能够快速定位实体“vxworks”。目前字符串匹配算法中最为高效的是 AC 状态机。该算法是一种只扫描一遍文本就能完成字符串匹配的算法。其核心内容在于建立有效状态转移路线图和失效状态转移路线图，形成一个完整的状态机[17]。

2）使用 Bert 计算词向量的距离：算法模型在 Bert 层时输出的是词向量。为比较两个词语之间的相似度，便转换为直接计算两个词向量之间的距离。而计算词向量的距离有多种指标，最简单的是计算欧式距离。本文选择计算余弦距离，其公式如下：

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| * \|B\|} \tag{2}$$

3）同义词管理：该系统作为管理系统，后台使用了图数据库数据和 Mysql 数据库进行管理，所以可以人为地对一些常见或者与业务关系较大的同义词进行管理。同时用户在系统进行搜索时，都会留下记录，以便后续管理员可以定时人为地分析该系统面向对象的需求点，从而完善系统的推荐功能。

2.4　持久化运维

典型测试用例需要不断地累积与更新。作为开放式的推荐系统，系统提供新增或编辑测试用例的功能，用户可以新增测试用例、用例分类、用例详细内容、用例中实体间的语义关系等。在用户使用的同时会记录用户输入的搜索问题，按照一定的更新周期，将记录中的问题及新增的测试用例制作成训练集，不断地对模型进行训练。

同时对典型测试用例引入“用例版本管理”功能，记录测试用例每个版本的更改人、更改内容的操作记录。管理员可选择某个版本作为测试用例当前展示的内容，或者删除某一个版本甚至整个已经过时的测试用例。

3　测试用例期望结果生成

上述章节简要介绍了基于 Bert 模型完成实体提取任务。而用例期望结果生成则参考了百度飞浆的 Couplet 案例。Couplet 本身任务是完成自动对联。通过上联自动对出下联的行为与通过用例输入得到期望结果的意义极为相同。该案例使用了 Seq2Seq 模型，即编码器－解码器（Encoder-Decoder）结构，用编码器将源序列编码成 vector，再用解码器将该 vector 解码为目标序列。在编码器方面，模型采用了基于 LSTM 的多层的 RNN encoder。在解码器方面，模型使用了带注意力（Attention）机制的 RNN decoder，在预测时使用 Beam Search 算法来生成结果。因此该模型与算法可以完美实现用例期望结果的自动生成任务。

模型虽然是使用了成熟的 Seq2Seq 模型，但还需要根据真实使用场景去构建相对应的训练集。目前公司已经有接近 10 年的真实测试用例数据，该数据是以表格形式存储在 Word 中，因此需要将 Word 中的相关内容提取出来制作为训练集。提取 Word 表格则使用了 Python-docx 库，通过该库遍历文档中的表格，可以提取出表格中的输入与期望结果，从而形成训练集与测试集用于模型训练。

4　实验结果与分析

4.1　Bert＋BiLSTM＋CRF 模型在测试用例中提取能力验证

因为用例期望结果生成是采用成熟模型，本系统只是制作了自己的训练集用于训练，因此实验章节不对该内容进行实验与分析。

而为了验证本文推荐算法中实体提取的模型与传统 Bert，BiLSTM，CRF 模型对测试用例中实体提取的准确性及有效性。使用测试集分别对 4 个模型进行对比。并以准确率 P，召回率 R，F_1 值作为评判标准。各个评价指标的计算方式如下：

$$P = \frac{a}{B} \times 100\% \tag{3}$$

$$R = \frac{a}{A} \times 100\% \tag{4}$$

$$F_1 = \frac{2PR}{(P+R)} \times 100\% \tag{5}$$

式中，a 为识别正确的实体数，A 为总实体个数，B 为识别出的实体数。

测试集由针对测试用例进行标注之后的语料组成，如图 5 所示。本文选用 BIO 标注体系，例如 RTU 软件。“R”为“B-SOFTWARE”，代表软件标签的开始，其他字符则为“I-SOFTWARE”，代表剩余软件标签。

图 5　典型测试用例语料标注

采用上述语料制作的测试集，并使用算法对测试集进行提取，使用准确率、召回率、F_1 值作为评价标准，最终

得出实验数据如表 1 所示。

表 1 多模型的实验结果对 %

模型	准确率	召回率	F_1 值
BiLSTM	60.7	59.2	59.9
CRF	56.3	53.2	54.7
Bert	73.0	65.4	68.9
本文模型	85.4	79.6	82.3

通过上述实验结果表明，单纯使用 BiLSTM 及 CRF 都明显低于 Bert 模型。可见 Bert 模型在自然语言处理任务中的能力明显高于传统模型。但将上述 3 个模型进行整合后，测试用例实体识别的准确率及其他各项指标均再度提高。其主要原因在于使用 BiLSTM 与 CRF 后能够充分利用语句相邻标签之间的关联性，从而获取全局最优的标签序列，从而改善实体识别的性能。

4.2 用例推荐与期望结果生成实例

算法验证后经过部署，系统在公司内部上线试运行。例如用户输入问题“组帧通道出现过什么问题”。经过本文算法提取出“组帧通道”作为设备实体。再遍历系统中自顶向下构建的关系，在遍历至“问题现象”关系时，形成查询知识图谱的 Cpyher 语句“match (node) - [relation] - (answer) where node.name = ‘组帧通道’ and where relation.name = ‘问题现象’ return answer”，最终通过该语句在知识图谱数据库中进行查找，当 answer 为非空时即得出推荐内容。最后经过前端界面的渲染得出的效果如图 6 所示。

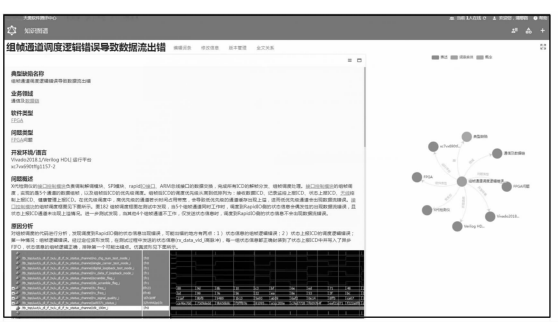

图 6 推荐结果展示

期望结果生成是用户上传已经写好的测试用例步骤说明，系统自动生成期望结果。如表 2 所示。

表 2 期望结果生成实例

用例步骤说明	期望结果
点击运行按钮	软件正常运行
设置频率为 100 MHz	参数设置成功,回显 100 MHz

5 结束语

本文搭建的智能测试用例推荐与期望结果生成系统，是在 Bert＋BiLSTM＋CRF 的实体提取模型与 Seq2Seq 模型和多模态的知识图谱的基础上于后台集成。该系统对传统的测试管理系统进行提升，使得公司能够基于自己的业务积累，搭建并管理具有自身专业领域的典型测试用例，从而帮助、发现并解决未来可能出现的问题，为质量改进提供参考和依据。同时该系统也实现了测试用例期望结果的自动生成，提升了测试人员在文档编写与用例设计的效率，使得用户关注用例质量本身或者关键步骤，而非其他形式上的内容。

参考文献：

[1] 刘 峤，李 杨，段 宏，等. 知识图谱构建技术综述 [J]. 计算机研究与发展，2016，53 (3)：582－600.
[2] 赵雪芹，李天娥，曾 刚. 基于 Neo4j 的万里茶道数字资源知识图谱构建研究 [J]. 情报资料工作，2022，43 (5)：89－97.
[3] 傅源坤，柳先辉，赵卫东. 基于 BERT 的智能制造装备命名实体识别方法 [J]. 制造业自动化，2022，44 (9)：120－124.
[4] 陆 枫. 基于 Neo4j 的人员关系知识图谱构建及应用 [J]. 软件工程，2022，25 (9)：5－8.
[5] 华振宇. 两个 Python 第三方库：Pandas 和 NumPy 的比较 [J]. 电脑知识与技术，2023，19 (1)：71－73.
[6] 曾小芹. 基于 Python 的中文结巴分词技术实现 [J]. 信息与电脑（理论版），2019，31 (18)：38－39.
[7] 苗 将，张仰森，李剑龙. 基于 BERT 的中文新闻标题分类 [J]. 计算机工程与设计，2022，43 (8)：2311－2316.
[8] 沈立力，姜 鹏，王 静. 基于 BERT 模型的中文期刊文献自动分类实践研究 [J]. 图书馆杂志，2022，41 (5)：109－118.
[9] 陈 玮，张 锐，尹 钟. BERT 模型结合实体向量的知识图谱实体抽取方法 [J]. 小型微型计算机系统，2022，43 (8)：1577－1582.
[10] 黄梅根，刘佳乐，刘 川. 基于 BERT 的中文多关系抽取方法研究 [J]. 计算机工程与应用，2021，57 (21)：234－240.
[11] 魏 迪，曾海彬，洪 锋，等. 基于 LSTM 网络和特征融合的通信干扰识别 [J]. 电讯技术，2022，62 (4)：450－456.
[12] 李朝杨，王希胤. 基于 Attention-BiLSTM 模型的 Python 源代码漏洞检测方法 [J]. 华北理工大学学报（自然科学版），2023，45 (2)：95－103.
[13] 李凯月. 基于 XLNet-BiLSTM 模型的个性化混合推荐算法 [J]. 数字技术与应用，2023，41 (3)：50－51.
[14] 李雪涵，陈焕明，华 航. 基于 CRF 的驾驶员意图在线识别 [J]. 汽车实用技术，2023，48 (2)：51－61.
[15] 廖 涛，陈彦杰，张顺香. 融合字词特征的 BiGRU-CRF 中文事件要素识别 [J]. 阜阳师范大学学报：自然科学版，2022，39 (4)：50－55.
[16] 陈月月，李 燕，帅亚琦，等. 基于 BERT-CRF 的中文分词模型设计 [J]. 电脑知识与技术，2022，18 (35)：4－6.
[17] 巫喜红，曾 锋. AC 多模式匹配算法研究 [J]. 计算机工程，2012，38 (6)：279－281.
[18] 罗 玲，孙 学，唐德波. 知识图谱在战术云服务平台中的应用 [J]. 电讯技术，2020，60 (9)：1035－1042.
[19] 殷方言. 基于 Markdown 标记语言的可扩展 CMS 研发 [D]. 成都：西南大学，2022.
[20] 胡学军，李嘉诚. 基于 Scrapy-Redis 的分布式爬取当当网图书数据 [J]. 软件工程，2022，25 (10)：8－11.