

# 基于知识库的自动问答系统

方桂安, 刘梦莎, 梁静蕾

**摘要**—本次期中项目我们小组复现并改进了基于知识库的自动问答系统, 由方桂安负责代码复现、背景调研、改进实现和报告润色, 由刘梦莎负责资料整理、报告规划、编写润色, 由梁静蕾负责报告编写与分析排版。

**关键词**—实体识别, 属性映射, 问答系统, BERT

## I. 实验概述

ChatGPT 的火爆出圈及其后续 LLM 等火热研究进展引发了我们小组对问答系统的兴趣, 所以我们选择了基于知识库的自动问答系统进行实验复现与改进。

## II. KBQA

### A. 概述

知识库 (KB) 是一个结构化数据库, 包含形式为 (主题、关系、对象) 的知识集合 (别名三元组)。大型知识库, 如 Freebase、DBpedia、Wikidata 和 YAGO, 已被构建为服务于许多下游任务。

知识图谱问答 (KBQA), 是一种基于结构化知识库 (即知识图谱) 的智能问答方法。即给定自然语言问题, 该类方法基于知识图对问题进行理解, 并根据问题理解的结果从知识图谱中查找或推理出问题对应的答案。

KBQA 的早期工作侧重于回答一个简单的问题, 其中只涉及一个事实。例如, “姚明的国籍是?”是一个简单的问题, 包括主题 “姚明”、关系 “国籍” 和关于事实, “(姚明, 国籍, 中国)” 的客体实体 “中国” 的查询。从包含数百万甚至数十亿事实的大规模知识库中检索正确的实体并非易事。因此, 研究人员花了很多精力提出不同的模型来回答简单的问题。

现在, 研究人员开始关注使用知识库回答复杂问题, 即复杂的 KBQA 任务。复杂问题通常包含多个主题, 表达复合关系, 或数值运算。如图 1 所示。此示例

方桂安, 20354027, (e-mail: fanggan@mail2.sysu.edu.cn)。  
刘梦莎, 20354091, (e-mail: liumsh6@mail2.sysu.edu.cn)。  
梁静蕾, 20354072, (e-mail: liangjlei@mail2.sysu.edu.cn)。

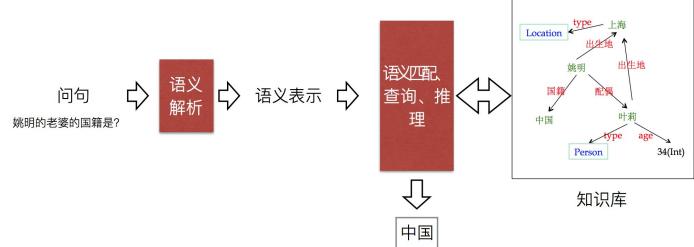


Fig. 1. 复杂 KBQA 的示例, 用于回答 “姚明的老婆的国籍是?” 这个问题。我们为这个问题提供了相关的 KB 子图。通往答案的真实路径用箭头表示。主题实体和答案实体分别以绿色和红色字体显示。

问题以主题 “姚明” 开头。不是查询单个事实, 而是问题需要两个关系的组合, 即 “配偶” 和 “国籍”。此查询还与实体类型约束 “(姚明, 出生地, 上海)” 相关联。通常, 复杂问题是涉及多跳推理、约束关系或数值运算的问题。

### B. 知识库

知识图谱 (KB) 是一个结构化数据库, 用于相关领域的知识采集、整理及提取。知识库中的知识源于领域专家, 是求解问题所需领域知识的集合, 包括一些基本事实、规则和其他相关信息, 其基本组成形式为 “实体—关系—实体”的三元组。

“奥巴马出生在火奴鲁鲁。”可以用三元组表示为 (BarackObama, PlaceOfBirth, Honolulu)。

这里把三元组理解为 (实体 entity, 实体关系 relation, 实体 entity), 把实体看作是结点, 实体关系 (包括属性, 类别等等) 看作是一条边, 那么包含了大量三元组的知识库就成为了一个庞大的知识图谱。

知识库可以分为两种类型, 一种是以 Freebase, Yago2 为代表的 Curated KBs, 它们从维基百科和 WordNet 等知识库中抽取大量的实体及实体关系, 可以把它们看作是一种结构化的维基百科。知识库的另外一种类型, 是以 Open Information Extraction (Open IE), Never-Ending Language Learning (NELL) 为代

表的 Extracted KBs，它们直接从上亿个网页中抽取实体关系三元组。与 Freebase 相比，这样得到的知识更加具有多样性，而它们的实体关系和实体更多的则是自然语言的形式，如“奥巴马出生在火奴鲁鲁。”可以被表示为 (“Obama”， “was also born in”， “Honolulu”)，但其精确度要低于 Curated KBs。

Extracted KBs 知识库涉及到的两个关键技术是：

- 1) **实体链接 (Entity linking)**，即将文档中的实体名字链接到知识库中特定的实体上。它主要涉及自然语言处理领域的两个经典问题实体识别 (Entity Recognition) 与实体消歧 (Entity Disambiguation)，简单地来说，就是要从文档中识别出人名、地名、机构名、电影等命名实体。并且，在不同环境下同一实体名称可能存在歧义，如苹果，我们需要根据上下文环境进行消歧。
- 2) **关系抽取 (Relation extraction)**，即将文档中的实体关系抽取出来，主要涉及到的技术有词性标注 (Part-of-Speech tagging, POS)，语法分析，依存关系树 (dependency tree) 以及构建 SVM、最大熵模型等分类器进行关系分类等。

### C. 主流方法

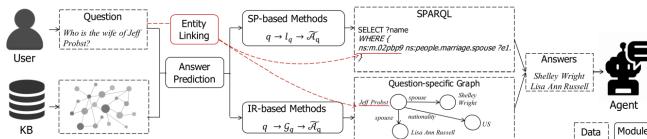


Fig. 2. KBQA 系统的架构。实体链接过程以红色显示。

关于简单 KBQA 任务的解决方案传统的主流方法可以分为两类。如图 2 所示。展示了简单的 KBQA 系统的总体架构。这两类方法通常被称为基于语义解析 (SP-based) 的方法和基于信息检索 (IR-based) 的方法。这两种方法首先识别问题中的主题并将其链接到 KB 中的实体 (称为主题实体)。然后通过执行解析的逻辑形式或在从知识库中提取的特定问题图中进行推理，在主题实体的邻域内得出答案。他们设计不同的工作机制解决 KBQA 任务。前一种方法用符号逻辑形式表示一个问题，然后针对知识库执行得到最终答案。后一种方法构建一个特定于问题的图，提供与问题相关的综合信息，并根据提取的图生成最终答案。

基于 SP 的方法是通过对逻辑形式进行自底向上的解析，得到一种可以表达整个问题语义的逻辑形式，通

过相应的查询语句在知识库中进行查询，从而得出答案。基于 IR 的方法是通过检索实体的知识库子图，依据某些规则或模板进行信息抽取，得到问题特征向量，建立分类器对候选答案进行筛选，从而得出最终答案。总而言之，这两种方法要么遵循先解析后执行的范例，要么遵循检索并生成的范例。为了显示两种范式之间的区别，我们在图 3 中用详细的模块说明了它们的问答过程。

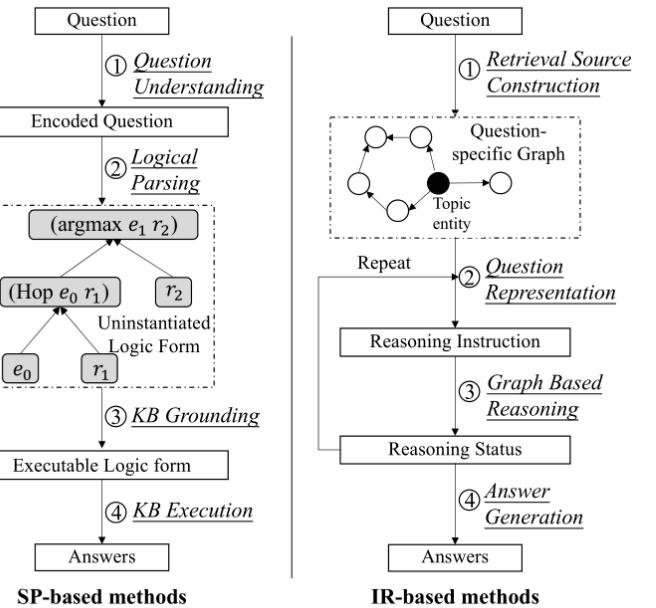


Fig. 3. KBQA 两种主流方法的图示

1) **基于语义解析 (SP-based) 的方法:** 基于语义解析的方法如图 4 所示，我们将基于 SP 的方法的过程概括为以下四个模块：

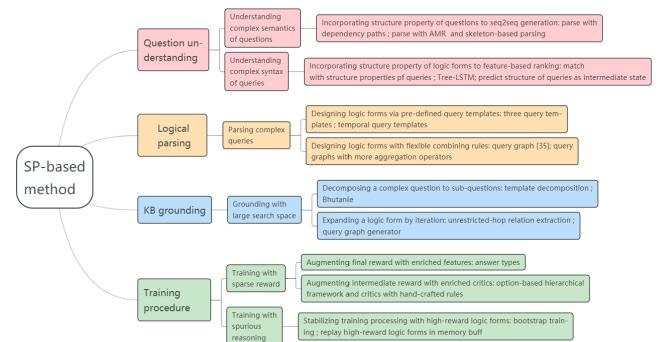


Fig. 4. 基于语义解析的方法

- 1) 使用问题理解模块将问题转化为机器能够理解和执行的语义表示，获取一个 encoded question 为

后续的逻辑解析步骤所使用。我们将这个模块表示如下：

$$\tilde{q} = \text{Question\_Understanding}(q)$$

,

其中  $\tilde{q}$  是捕获自然语言问题的语义和句法信息的编码问题。它可以是分布式表示、结构化表示或它们的组合。

- 2) 通过逻辑解析模块把 encoded question 解析成未被实例化的逻辑形式。未被实例化是指问题的某种句法表示，其中的实体和关系还没被实例化：

$$\bar{l}_q = \text{Logical\_Parsing}(\tilde{q})$$

其中  $\bar{l}_q$  是未实例化的逻辑形式，没有填充详细的具体设计。在这里，可以通过生成一系列标记或对一组候选者进行排名来获得  $\bar{l}_q$ 。在实践中，Seq2seq 模型或基于特征的排名模型被用来根据编码问题生成  $\bar{l}_q$ 。

- 3) 为了在知识库上执行查询，实例化模块会将未被实例化的逻辑形式填充实体和关系实例化，并且和结构化知识库中的实体关系做对齐。需要注意的是，在有的模型里面，逻辑解析模块和实例化模块可能是一起进行的，边解析边实例化：

$$l_q = \text{KB\_Grounding}(\bar{l}_q, \mathcal{G})$$

在这一步之后， $l_q$  被  $\mathcal{G}$  中的实体和关系实例化，这样我们就得到了一个可执行的逻辑形式  $l_q$ 。值得注意的是  $l_q$  总是包含  $e_q$ ，这是通过实体链接模块检测到的。它的格式不限于 SPARQL 查询，但总是可以转换为 SPARQL。

- 4) 在 KB 上执行实例化的逻辑形式来生成预测答案：

$$\tilde{\mathcal{A}}_q = \text{KB\_Execution}(l_q)$$

其中  $\tilde{\mathcal{A}}_q$  是给定问题  $q$  的预测答案。该模块通常通过现有的执行器来实现。在训练期间，逻辑形式  $l_q$  被视为中间输出。这些方法使用  $\mathcal{D} = \{(q, \mathcal{A}_q)\}$  格式的 KBQA 数据集进行训练，其中目标设置为生成与问题语义匹配的逻辑形式并得出正确答案。

以上四个模块，question understanding, logical parsing, KB grounding, KB execution，在处理复杂 KBQA 时都会遇到不同的挑战：首先，当问题复杂，问题理解模块在语义和句法方面都是非常困难的；其次，逻辑解析必须复杂问题的不同类型，尤其是涉及多个关系和主语实体的复杂问题会显著增加解析时的搜索空间，使解析过程变得低效；最后，用于监督解析情况的逻辑形式标签，手工标注极其昂贵、需要大量投入，弱监督的情况（即问题只有答案实体作为监督信号）下是非常具有挑战性的任务。

2) 基于信息检索 (IR-based) 的方法：同样，我们将基于 IR 的方法的过程总结为四个模块，如图 5 所示：

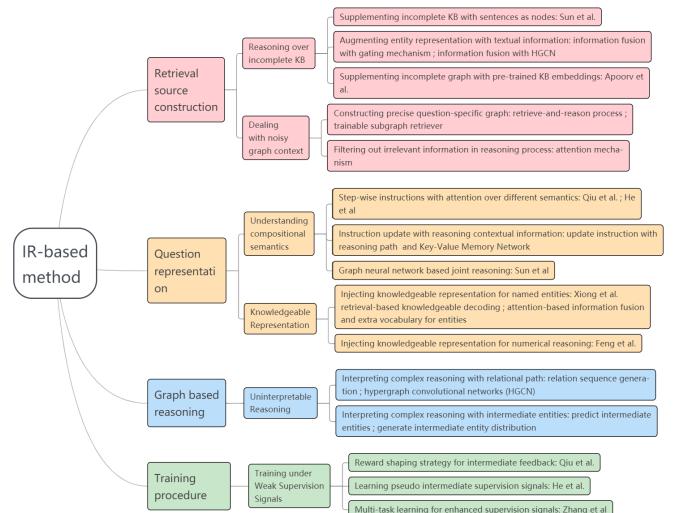


Fig. 5. 基于信息抽取的方法

- 1) 从主题实体  $e_q$  开始，系统首先从 KBS 中提取特定问题的图 (question-specific graph)。理想情况下，该图包括所有与问题相关的实体和关系，分别作为节点和边。在没有显式生成可执行逻辑形式的情况下，基于 IR 的方法对图执行推理。我们表示一个检索源构建模块，将问题和知识库作为输入：

$$\mathcal{G}_q = \text{Retrieval\_Source\_Construction}(q, \mathcal{G})$$

其中  $\mathcal{G}_q$  是从  $\mathcal{G}$  中提取的特定问题图。随着子图的大小到主题实体的距离呈指数增长，并采用了一些过滤技巧来控制图形计算规模。

- 2) 接着，通过问题表示模块（作用是分析问题语义），把输入问题编码成向量，作为后续推理的指导信息。通常，问题  $q$  通过神经网络（例如 LSTM、

GRU 和 PLM) 编码为隐藏向量  $q$ , 然后结合其他方法 (例如注意力机制) 生成一个向量作为指令:

$$\mathbf{i}^{(k)} = \text{Question\_Representation}(\mathbf{i}^{(k-1)}, q, \mathcal{G}_q)$$

这里,  $\{\mathbf{i}^{(k)}, k = 1, \dots, n\}$  是第  $k$  个指令向量推理, 将语义和句法编码形成自然语言问题。多步推理和一步匹配都适用, 从而导致不同的推理步骤  $n$ 。

- 3) 基于图的推理模块, 通过基于向量的计算, 在 question-specific graph 图内, 沿着边传播和聚合邻居实体的信息, 以达到后续对问题和答案进行语义匹配的目的。基于图的推理模块通过基于向量的计算进行语义匹配, 以沿着图中的相邻实体传播和聚合信息。推理状态  $\{\mathbf{s}^{(k)}, k = 1, \dots, n\}$  在不同的方法中具有不同的定义 (例如, 预测实体的分布和关系的表示), 根据推理指令更新:

$$\mathbf{s}^{(k)} = \text{Graph\_Based\_Reasoning}(\mathbf{s}^{(k-1)}, \mathbf{i}^{(k)}, \mathcal{G}_q)$$

其中  $\mathbf{s}^{(k)}$  是所考虑的推理状态作为图上第  $k$  个推理步骤的状态。

- 4) 答案生成模块用于在推理结束时根据推理状态生成答案。这种生成器主要有两种类型:

- a) 实体排名生成器, 对实体进行排名以获得排名靠前的实体作为预测答案,
- b) 文本生成器, 使用词汇  $\mathcal{V}$  生成自由文本答案。该模块可以形式化作为:  $\tilde{\mathcal{A}}_q = \text{Answer\_Generation}(\mathbf{s}^{(n)}, \mathcal{G}_q, \mathcal{V})$  在实体排名范式中,  $\mathcal{G}_q$  中包含的实体是答案预测的候选对象  $\mathcal{A}_q$  在许多情况下,  $\mathcal{A}_q$  是通过选择得分大于预定义阈值的实体获得的, 其中  $(n)$  得分来自  $\mathbf{s}^{(n)}$ 。而在文本生成范式中, 答案是从词汇表  $\mathcal{V}$  中生成的, 作为标记序列。

在训练期间, 实体排名生成器的目标通常是将正确的实体排名高于  $\mathcal{G}_q$  中的其他实体。相比之下, 文本生成器通常经过训练以生成最优答案 (正确实体的名称)。

- 3) PLM 方法:** PLM 在复杂 KBQA 上的应用在大型文本语料库上进行无监督预训练语言模型, 然后在下游任务上微调预训练语言模型 (PLM) 已成为自然语言处理的流行范例。此外, 由于从大规模的广泛数据中获得的强大性能和服务于广泛下游任务的能力, PLM 被公

认为许多任务的“基础模型”, 包括复杂的 KBQA 任务。因此, 最近一些基于 SP 和 IR 的方法已将 PLM 广泛纳入其中。

- 1) 对于基于 SP 的方法, PLM 总是用于同时优化可训练模块 (即问题理解、逻辑解析、知识库基础), 这有助于在 seq2seq 框架中生成可执行程序 (例如 SPARQL)。有了这样一个统一的范例, 可以利用跨任务的可转移知识来缓解低资源场景中的数据稀疏性问题。对于基于 IR 的方法, PLM 有助于精确构建源, 进一步增强统一推理能力。一方面, PLM 提供了强大的表示能力, 可以从 KB 中检索语义相关信息。另一方面, PLM 可以帮助统一问题和 KB 的表示, 这有助于提高推理能力。配备强大的 PLM, 逻辑形式生成模块受益于通过无监督预训练获得的强大生成和理解能力。在统一的 seq2seq 生成框架下, PLM 提供可迁移的知识, 以帮助利用增强逻辑表单生成的 PLM。为了获得可执行程序 (例如, SPARQL), 传统的基于 SP 方法将问题解析为逻辑形式, 并通过 KB grounding 将其实例化。这个过程可以在知识增强的文本生成框架下很好地形式化 (即, 从用户请求到可执行程序)。
- 2) 基于 IR 方法的 PLM, 借助 PLMs 强大的表示能力, 我们可以增强对问题特定图的检索, 减少搜索源构建过程中知识库的不完整性。此外, PLM 提供了一种在统一语义空间中对非结构化文本和结构化知识库信息进行建模的统一方式, 从而改进了特定问题的图推理。PLM 用于精确和统一的推理。受到强大的预训练语言模型的吸引, 一些研究人员对图结构的复杂推理进行了调整, 以进一步参与 PLM。虽然基于 KB 的传统推理依赖于为实体和关系学习的嵌入, 但此类嵌入可能无法识别问题回答上下文的相关部分。为了进一步对问答上下文 (即问答序列) 和知识图进行联合推理, 检索到的子图中的节点表示使用问题、答案和节点表面名称的串联序列的 PLM 编码进行初始化。

#### D. 评估的基准测试

为了全面评估 KBQA 系统, 应考虑多方面的有效测量。考虑到要实现的目标, 我们将测量分为三个方面: 可靠性、鲁棒性和系统-用户交互。

**可靠性:** 对于每个问题，都有一个答案集（一个或多个元素）作为基本事实。KBQA 系统通常预测具有最高置信度分数的实体以形成答案集。如果答案集中存在 KBQA 系统预测的答案，则为正确预测。研究常用的一些经典评价指标：Precision、Recall、F1 值、Hits@1。

对于一个问题  $q$ ，其 Precision 表示正确预测与所有预测答案的比率。它被正式定义为：

$$\text{Precision} = \frac{|\mathcal{A}_q \cap \tilde{\mathcal{A}}_q|}{|\tilde{\mathcal{A}}_q|}$$

$\tilde{\mathcal{A}}_q$  是预测答案， $\mathcal{A}_q$  是基本事实。召回率是正确预测与所有基本事实的比率。它被计算为：

$$\text{Recall} = \frac{|\mathcal{A}_q \cap \tilde{\mathcal{A}}_q|}{|\mathcal{A}_q|}$$

理想情况下，我们期望 KBQA 系统同时具有更高的 Precision 和 Recall。因此 F1 分数最常用于给出综合评价：

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

还会使用 Hits@1 来评估正确预测排名高于其他实体的分数。它被计算为：

$$\text{Hits}@1 = \mathbb{I}(\tilde{a}_q \in \mathcal{A}_q)$$

其中  $\tilde{a}_q$  是  $\tilde{\mathcal{A}}_q$  的前 1 个预测。

**鲁棒性:** KBQA 模型应该在测试时泛化到分布外的问题。然而，当前的 KBQA 数据集主要是基于模板生成的，缺乏多样性。而且，训练数据集的规模受到昂贵的标记成本的限制。此外，由于查询的广泛覆盖和组合爆炸，KBQA 系统的训练数据可能很难覆盖所有可能的用例查询。在基本层面上，假设 KBQA 模型使用从同一分布中提取的问题进行训练和测试，这是大多数现有研究关注的重点。除此之外，强大的 KBQA 模型可以泛化到所用模式项（例如，关系和实体类型）的新组合。为了实现更好的泛化并为用户提供服务，鲁棒的 KBQA 模型应该处理其模式项或域未包含在训练阶段的问题。

**系统-用户交互:** 虽然当前大多数研究都非常关注离线评估，但忽略了用户与 KBQA 系统之间的交互。一方面，在搜索场景中，应考虑用户友好的界面和可接受的响应时间。为了对此进行评估，应收集用户的反馈

并判断系统的效率。另一方面，如果只提供单轮服务，用户的搜索意图很容易被系统误解。所以，评估 KBQA 系统的交互能力很重要。例如，检查他们是否可以提出澄清问题来消除用户查询的歧义，以及他们是否可以回应用户报告的错误。迄今为止，缺乏对系统与用户交互能力的量化衡量，但人工评价不失为一种高效、综合的方式。

## E. 未来的趋势和展望

复杂 KBQA 任务的几个有前途的未来方向：

**不断进化的知识图谱问答系统:** 目前的知识图谱问答系统通常是在线下进行训练然后部署到线上。然而，大部分已有的知识图谱问答系统都会忽略线上部署后对新实例的学习。与此同时，外部知识在不断进化，一个知识图谱问答系统应该能够在线上部署之后不断进化的。目前有少量的工作关注到这一块。比如引入连续学习 (continuous learning) 的框架。当知识图谱问答系统遇到没见过的问题时，他们可以先将其和系统中已有的训练问题进行比对，召回最相似问题的解析式，让用户从中选出有效的解析式加入到训练数据中。再如让用户直接参与到系统中，当用户提出没见过的问题或者包含歧义的问题时，系统给出一些候选的消歧后的问题供用户选择。

**鲁棒的知识图谱问答系统:** 已有的知识图谱问答研究大部分是在一种理想情况下展开的，然而真实场景下，数据可能会严重缺失或者不足。现有少量工作聚焦数据不足时，用元学习 (meta learning) 训练知识图谱问答系统，以及用无监督的双语词典归纳 (bilingual lexicon induction) 技术对低资源 (low-resource) 数据进行增强。与此同时，有一部分工作对分布外异常 (out-of-distribution) 的问题开展了研究。他们提出了 CFQ 和 GrailQA 数据集促进这方面的未来研究。

**会话式 KBQA 系统:** 对话型的知识图谱问答是新兴的任务。在单轮的问答之后，用户会围绕该主题提出一系列问题。目前对于对话型知识图谱问答的研究主要是针对如何判断后续问题的主题展开的。用户提出的问题间将具有某一些联系，而不是独立存在。对历史问题进行记忆和理解有助于回答当前问题。同时，为了减少对话问答带来的搜索空间增大的困难，也有部分工作在这个方向进行了简单的探索。

**更加一般定义的知识图谱:** 由于 KB 的不完整性，研究人员结合了额外的信息（例如文本、图像和人类交

互) 来补充知识库, 这将进一步解决复杂 KBQA 任务的信息需求。有一些任务, 如: 视觉问答 (VQA), 常识知识推理 (Commonsense Knowledge Reasoning) 可以被看作是在一种特殊的知识图谱上做推理。比如有部分工作将阅读理解的任务中的文本看做虚拟的知识图谱, 然后在上面做推理。除了显性地在任务中构建知识图谱, 有部分工作将其他任务看作隐性的知识图谱。近年来较为突出的是, 部分研究者将预训练语言模型作为特殊的知识库, 认为其具有储存知识和推理的能力。

### III. 数据集

NLPCC 全称自然语言处理与中文计算会议 (The Conference on Natural Language Processing and Chinese Computing), 它是由中国计算机学会 (CCF) 主办的 CCF 中文信息技术专业委员会年度学术会议, 专注于自然语言处理及中文计算领域的学术和应用创新。

此次实验使用的数据集来自 NLPCC ICCPOL 2016 KBQA 任务集, 其包含 14609 个问答对的训练集和包含 9870 个问答对的测试集。并提供一个知识库, 包含 650 2738 个实体、587 875 个属性以及 430 637 96 个三元组。知识库文件中每行存储一个事实 (fact), 即三元组 (实体、属性、属性值)。如图 6, 各文件统计如下:

```
训练集: 14609
开发集: 9870
知识库: 43063796
```

Fig. 6. 数据集文件统计

知识库样例如下所示:

```
"希望之星"英语风采大赛 ||| 中文名 ||| "希望之星"英语风采大赛
"希望之星"英语风采大赛 ||| 主办方 ||| 中央电视台科教节目中心
"希望之星"英语风采大赛 ||| 别名 ||| "希望之星"英语风采大赛
"希望之星"英语风采大赛 ||| 外文名 ||| Star of Outlook English Talent Competition
"希望之星"英语风采大赛 ||| 开始时间 ||| 1998
"希望之星"英语风采大赛 ||| 比赛形式 ||| 全国选拔
"希望之星"英语风采大赛 ||| 节目类型 ||| 英语比赛
```

Fig. 7. 知识库样例

问答对样例如 8 所示:

数据集本身存在一些问题例如知识库实体之间会存在歧义, 以“贝拉克·奥巴马”为例, 涉及该实体的问答对如图 9:

在知识库中查询包含该实体的三元组, 结果如图 10:

```
<question id=1> 《机械设计基础》这本书的作者是谁?
<triple id=1> 机械设计基础 ||| 作者 ||| 杨可桢, 程光蕴, 李仲生
<answer id=1> 杨可桢, 程光蕴, 李仲生
=====
<question id=2> 《高等数学》是哪个出版社出版的?
<triple id=2> 高等数学 ||| 出版社 ||| 武汉大学出版社
<answer id=2> 武汉大学出版社
=====
<question id=3> 《线性代数》这本书的出版时间是什么?
<triple id=3> 线性代数 ||| 出版时间 ||| 2013-12-30
<answer id=3> 2013-12-30
=====
```

Fig. 8. 问答对样例

```
<question id=9687> 谁是贝拉克·奥巴马的妻子?
<triple id=9687> 贝拉克·奥巴马 ||| 妻子 ||| 米歇尔·奥巴马
<answer id=9687> 米歇尔·奥巴马
```

Fig. 9. 涉及“贝拉克·奥巴马”的问答对

```
贝拉克·奥巴马(美国现任总统) ||| 别名 ||| 贝拉克·奥巴马
贝拉克·奥巴马(美国现任总统) ||| 姓名 ||| 贝拉克·侯塞因·奥巴马
贝拉克·奥巴马(美国现任总统) ||| 妻子 ||| 米歇尔·奥巴马
.....
贝拉克·奥巴马 ||| 主要成就 ||| 1996年伊利诺伊州参议员 美国第56届、57届总统
贝拉克·奥巴马 ||| 代表作品 ||| 《我相信变革》《我父亲的梦想》《无畏的希望》
贝拉克·奥巴马 ||| 妻子 ||| 米歇尔·拉沃恩·奥巴马
.....
贝拉克·奥巴马(美国第44任总统) ||| 血型 ||| ab
贝拉克·奥巴马(美国第44任总统) ||| 学院 ||| 西方学院
贝拉克·奥巴马(美国第44任总统) ||| 妻子 ||| 米歇尔·拉沃恩·奥巴马
```

Fig. 10. 在知识库中查询包含该实体的三元组

首先, 知识库中存在“贝拉克·奥巴马”的多条实体, 有可能是多数据来源的融合或其他原因, 从而并不能完全保证信息的对齐。我们查看“妻子”这一属性, 发现有的是“米歇尔·拉沃恩·奥巴马”有的是“米歇尔·奥巴马”, 而我们问答对中给出的答案是“米歇尔·奥巴马”。因此当我们的模型检索到正确三元组时, 如图 11:

```
贝拉克·奥巴马(美国第44任总统) ||| 妻子 ||| 米歇尔·拉沃恩·奥巴马
```

Fig. 11. 检索到的正确三元组

除了知识库实体之间会存在歧义, 问题中实体之间也存在歧义, 以“博士来拜”为例, 涉及该实体的问答对如图 12:

```
<question id=249> 博士来拜是什么年代的作品?
<triple id=249> 博士来拜 ||| 年代 ||| 1461年
<answer id=249> 1461年
```

Fig. 12. 问题实体之间的歧义

在知识库中查询包含该实体的三元组，结果如图 13（部分）：问句中的问题是：“博士来拜是什么年代的”

```

博士来拜(曼特尼亞画作) ||| 别名 ||| 博士来拜
博士来拜(曼特尼亞画作) ||| 中文名 ||| 博士来拜
博士来拜(曼特尼亞画作) ||| 类别 ||| 油画, 壁画
博士来拜(曼特尼亞画作) ||| 年代 ||| 1461年
博士来拜(曼特尼亞画作) ||| 作者 ||| 曼特尼亞
...
博士来拜(维登画作) ||| 别名 ||| 博士来拜
博士来拜(维登画作) ||| 中文名 ||| 博士来拜
博士来拜(维登画作) ||| 类别 ||| 油画
博士来拜(维登画作) ||| 年代 ||| 1455年
博士来拜(维登画作) ||| 作者 ||| 维登
博士来拜(维登画作) ||| 属地 ||| 慕尼黑画廊藏
...
博士来拜(达·芬奇画作) ||| 别名 ||| 博士来拜
博士来拜(达·芬奇画作) ||| 中文名 ||| 博士来拜
博士来拜(达·芬奇画作) ||| 类别 ||| 油画
博士来拜(达·芬奇画作) ||| 年代 ||| 1481-1482
博士来拜(达·芬奇画作) ||| 作者 ||| 达芬奇
博士来拜(达·芬奇画作) ||| 现藏 ||| 佛罗伦萨乌菲兹美术馆
博士来拜(达·芬奇画作) ||| 规格 ||| 246 x 243 厘米

```

Fig. 13. 在知识库中查询包含该实体的三元组

作品？“，涉及到”年代“这个属性，而这幅作品被不同时期的很多人创作过，我们无法从当前问句下得到要询问的是哪位艺术家的创作年代。因此该问题的涉及的实体具有歧义性，同样的，当模型检索到我们认为的正确实体和正确属性后，依然有可能会被判定为错误答案。

在知识库中相关实体三元组数量过多的情况下，对检索模型的效果、效率也是个挑战在具有 4300W 条三元组的知识库中，同一个实体会检索出大量（几十、几百条）的相关三元组，而且在存在上述两个歧义性问题的情况下识别的效果和效率都是很大的问题。

#### IV. 模型总体架构

实验主要包含两个核心模块，实体识别模块提取问题中的关键实体，根据候选实体在知识库中查找候选答案，语义匹配模块则用来计算问题和候选答案的匹配程度，最终以匹配度最高的作为答案。流程图如图 14 所示。

KBQA 流程：

- 1) 输入问句。
- 2) 通过实体识别模型 BERT+BiLSTM 检测问句中的实体，得到 mention。
- 3) 通过检索模型在知识库中检索 mention，得到候选集（K 个候选实体的所有三元组）。
- 4) 通过属性映射模型 SimBERT 在候选集中挑选最合适的属性，得到唯一三元组。
- 5) 输出答案。

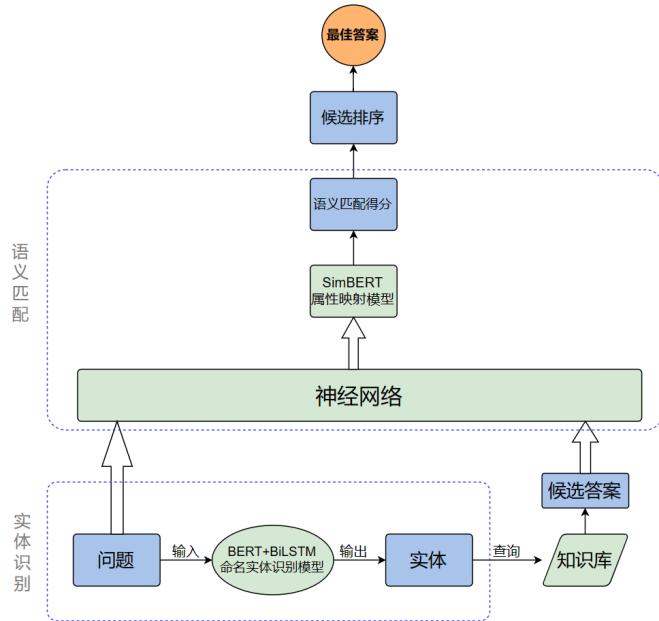


Fig. 14. 模型总体架构

#### V. 命名实体识别

命名实体识别 (Named Entities Recognition, NER) 是 NLP 里的一项很基础的任务，就是指从文本中识别出命名性指称项，为信息抽取、信息检索、机器翻译、问答系统等任务做铺垫。其目的是识别语料中人名、地名、组织机构名等命名实体。在特定的领域中，会相应地定义领域内的各种实体类型。在本实践中，命名实体识别目的是找到问句中询问的实体名称。

首先需要构造 NER 的数据集，根据三元组-Entity 反向标注问题，给数据集中的 Question 打标签。我们这里采用 BIO 的标注方式，将每个元素标注为“B-X”、“I-X”或者“O”：“B-X” (Begin) 表示此元素所在的片段属于 X 类型并且此元素在此片段的开头；“I-X” (Inside) 表示此元素所在的片段属于 X 类型并且此元素在此片段的中间位置；“O” (Outside) 表示不属于任何类型。其中“X”为“LOC”代表地名，“PER”代表人名，“ORG”代表组织机构名。由于本任务无需区分地点名、人名和组织名，只需要识别出实体，因此用 B-ENT, I-ENT 代替其他的标注类型，即 B-ENT 表示实体首字，I-ENT 表示实体非首字。

在实验指导书中，该任务利用 BERT+biilstm 模型对文本 token 序列进行编码表征，再利用一个全连接层对序列每个 token 进行分类，最后利用 Softmax 进行最终标签判断确定。

## A. 各层作用

**bert:** 提供词的嵌入表示，通过大规模训练，得到的结果泛化性更强，因此使用预训练模型，让参数有个比较好的初始化值。

**lstm:** 从这里开始是正式的模型内容，这里是双向 lstm，能够学习句子的上下文内容，从而给出每个字的标注。

## B. Bert

1) **BERT 简介:** BERT 的全称为 Bidirectional Encoder Representation from Transformers，是一个预训练的语言表征模型。它强调了不再像以往一样采用传统的单向语言模型或者把两个单向语言模型进行浅层拼接的方法进行预训练，而是采用新的 masked language model (MLM)，以致能生成深度的双向语言表征。BERT 论文发表时提及在 11 个 NLP (Natural Language Processing, 自然语言处理) 任务中获得了新的 state-of-the-art 的结果。

该模型有以下主要优点：

1) 采用 MLM 对双向的 Transformers 进行预训练，以生成深层的双向语言表征。

2) 预训练后，只需要添加一个额外的输出层进行 fine-tune，就可以在各种各样的下游任务中取得 state-of-the-art 的表现。在这过程中并不需要对 BERT 进行任务特定的结构修改。

2) **BERT 的结构:** 以往的预训练模型的结构会受到单向语言模型（从左到右或者从右到左）的限制，因而也限制了模型的表征能力，使其只能获取单方向的上下文信息。而 BERT 利用 MLM 进行预训练并且采用深层的双向 Transformer 组件（单向的 Transformer 一般被称为 Transformer decoder，其每一个 token（符号）只会 attend 到目前往左的 token。而双向的 Transformer 则被称为 Transformer encoder，其每一个 token 会 attend 到所有的 token。）来构建整个模型，因此最终生成能融合左右上下文信息的深层双向语言表征。经过多层 Transformer 结构的堆叠后，形成 BERT 的主体结构（图 15）：

3) **BERT 的输入:** BERT 的输入为每一个 token 对应的表征（图中的粉红色块就是 token，黄色块就是 token 对应的表征），并且单词字典是采用 WordPiece 算法来进行构建的。为了完成具体的分类任务，除了单词的

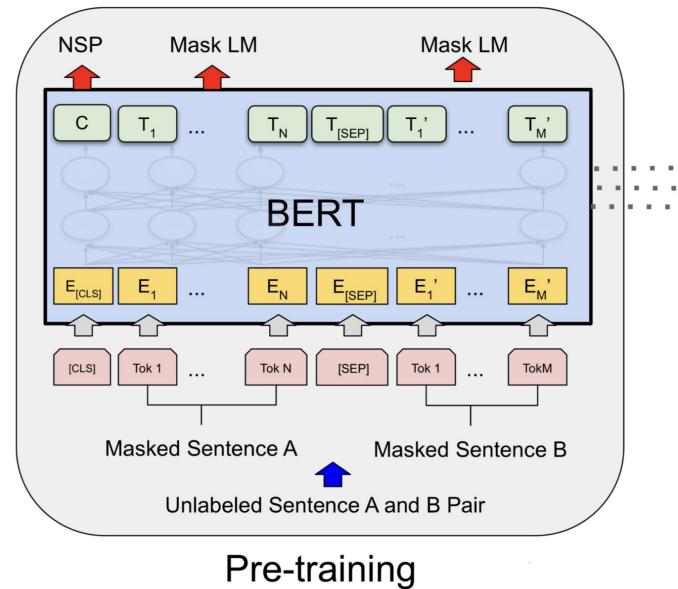


Fig. 15. BERT 的主体结构

token 之外，作者还在输入的每一个序列开头都插入特定的分类 token ([CLS])，该分类 token 对应的最后一个 Transformer 层输出被用来起到聚集整个序列表征信息的作用。

由于 BERT 是一个预训练模型，其必须要适应各种各样的自然语言任务，因此模型所输入的序列必须有能力包含一句话（文本情感分类，序列标注任务）或者两句话以上（文本摘要，自然语言推断，问答任务）。那么如何令模型有能力去分辨哪个范围是属于句子 A，哪个范围是属于句子 B 呢？BERT 采用了两种方法去解决：

1) 在序列 tokens 中把分割 token ([SEP]) 插入到每个句子后，以分开不同的句子 tokens。

2) 为每一个 token 表征都添加一个可学习的分割 embedding 来指示其属于句子 A 还是句子 B。

因此最后模型的输入序列 tokens 为图 16（如果输入序列只包含一个句子的话，则没有 [SEP] 及之后的 token）：

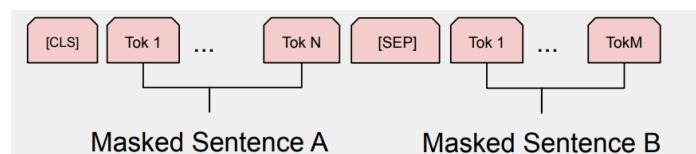


Fig. 16. BERT 的输入

4) **Embedding:** BERT 的 Embedding 由三种 Embedding 求和而成：

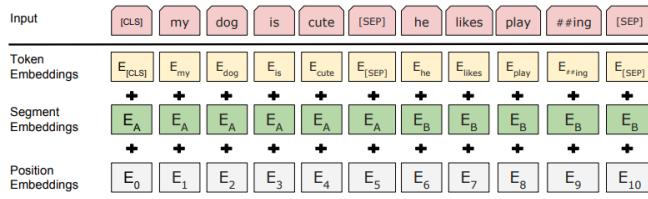


Fig. 17. BERT 的输入

其中：Token Embeddings 是词向量，第一个单词是 CLS 标志，可以用于之后的分类任务。Segment Embeddings 用来区别两种句子，因为预训练不光做 LM 还要做以两个句子为输入的分类任务。Position Embeddings 和之前文章中的 Transformer 不一样，不是三角函数而是学习出来的。

5) **BERT 的输出:** 如图 18,C 为分类 token ([CLS]) 对应最后一个 Transformer 的输出， $T_i$  则代表其他 token 对应最后一个 Transformer 的输出。对于一些 token 级别的任务（如，序列标注和问答任务），就把  $T_i$  输入到额外的输出层中进行预测。对于一些句子级别的任务（如，自然语言推断和情感分类任务），就把 C 输入到额外的输出层中，这里也就解释了为什么要在每一个 token 序列前都要插入特定的分类 token。

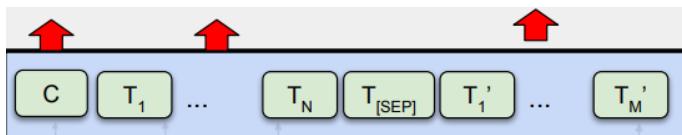


Fig. 18. BERT 的输出

### C. BiLSTM 简介

1) **LSTM 总体框架:** LSTM 的全称是 Long Short-Term Memory，它是 RNN (Recurrent Neural Network) 的一种。LSTM 由于其设计的特点，非常适合用于对时序数据的建模，如文本数据。BiLSTM 是 Bi-directional Long Short-Term Memory 的缩写，是由前向 LSTM 与后向 LSTM 组合而成。两者在自然语言处理任务中都常被用来建模上下文信息。

使用 LSTM 模型可以更好的捕捉到较长距离的依赖关系。因为 LSTM 通过训练过程可以学到记忆哪些信息和遗忘哪些信息。

LSTM 是一种特殊的循环神经网络，可以解决 RNN 的长期依赖问题，其关键就是细胞状态，见下图中贯穿单元结构上方的水平线。细胞状态在整个链上

运行，只有一些少量的线性交互，从而保存长距离的信息流。具体而言，LSTM 一共有三个门来维持和调整细胞状态，包括遗忘门，输入门，输出门，LSTM 的结构如图 19 所示。

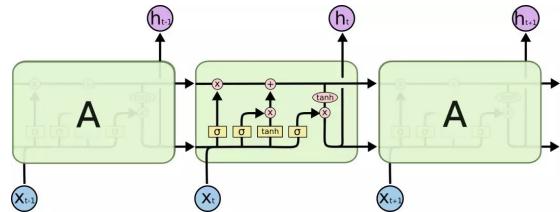


Fig. 19. LSTM 结构

2) **BiLSTM 结构:** 利用 LSTM 对句子进行建模还存在一个问题：无法编码从后到前的信息。在更细粒度的分类时，如对于强程度的褒义、弱程度的褒义、中性、弱程度的贬义、强程度的贬义的五分类任务需要注意情感词、程度词、否定词之间的交互。例如，“这个餐厅脏得不行，没有隔壁好”，这里的“不行”是对“脏”的程度的一种修饰，通过 BiLSTM 可以更好的捕捉双向的语义依赖。

将前向的 LSTM 与后向的 LSTM 结合，便得到了一层 BiLSTM，模型如图 20 所示。

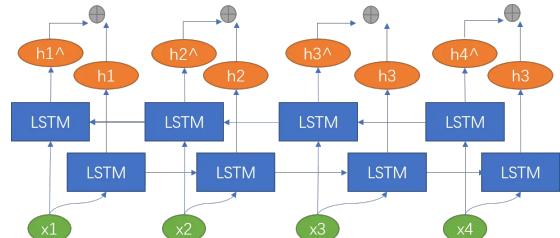


Fig. 20. BiLSTM 结构

3) **BiLSTM-softmax 总体架构:** BiLSTM-softmax 模型主体由双向长短时记忆网络 (Bi-LSTM) 和 softmax 组成，模型输入是字符特征，输出是每个字符对应的预测标签 (图 21)。

4) **模型输入:** 对于输入的自然语言序列，可通过特征工程的方法定义序列字符特征，如词性特征、前后词等，将其输入模型。但多数情况下，可以直接选择句中每个字符的字嵌入或词嵌入向量，可以是事先训练好的或是随机初始化。在此实验中，将 BERT 的序列的输出作为输入。

5) **特征提取:** 在 BiLSTM-softmax 中，一般使用一层的双向 LSTM 是足够的。因此，BiLSTM 对输入 embeddings 的特征提取过程如图 22。

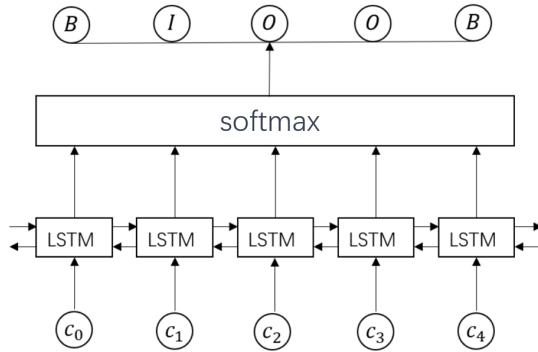


Fig. 21. 整体架构

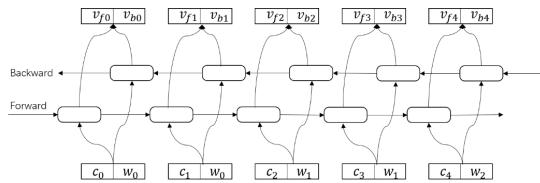


Fig. 22. 特征提取过程

BiLSTM 接收每个字符的 embedding，并预测每个字符的对 5 个标注标签的概率。上图得到的拼接向量维度大小为 [num\_directions, hidden\_size]。为将输入表示为字符对应各个类别的分数，需要在 BiLSTM 层加入一个全连接层，通过 softmax 将向量映射为一个 5 数值的分布概率。

#### D. 标注方法

命名实体识别任务常常转化为序列标注问题，利用 BIO、BIOES 和 BMES 等常用的标注规则对经过分词的文本进行 token 标注。本实验采用的标注规则为 BIO。

假设数据集的实体类别为  $k$  个，以 BIO 作为标注模式，命名实体识别的过程如下：

- 1) 给定一个文本  $text = "w_1 w_2 \dots w_n"$  的 token 序列为： $W = [w'_1, w'_2, \dots, w'_n]$ ,  $n$  为文本序列长度；
- 2)  $W$  经过嵌入层 (Embedding Layer) 获得序列的嵌入表示  $X \in R^{n \times d}$ ,  $d$  表示向量维度；
- 3)  $X$  经过文本编码层 (Encoder Layer) 对 token 序列建模获得序列的隐层表示  $H \in R^{n \times h}$ ,  $h$  表示隐层向量维度；
- 4)  $H$  经过一个全连接的分类层 (Classification Layer) 对每个 token 进行实体标签的预测，得到分类结果  $\text{Logits} \in R^{n \times k}$ ，其中每一行  $\log L_i \in \text{Logits} \in R^k$  表示文本中  $w_i$  为各个实体标签的预

测分数；

#### E. 实体标签判断

在实验指导书中，此任务使用的是 Softmax 实体标签判断，将 Logits 经过 Softmax 计算，将每个  $w_i$  对应概率最大的实体类别作为该  $w_i$  的实体标签，即对  $n$  个 token 部分进行  $k$  分类。

到了这一步，似乎我们通过 BiLSTM 已经找到每个单词对应的最大标签类别，但实际上，直接选择该步骤最大概率的标签类别得到的结果并不理想。原因在于，尽管 LSTM 能够通过双向的设置学习到观测序列之间的依赖，但 softmax 层的输出是相互独立的，输出相互之间并没有影响，只是在每一步挑选一个最大概率值的 label 输出，这样的模型无法学习到输出的标注之间的转移依赖关系（标签的概率转移矩阵）以及序列标注的约束条件，如句子的开头应该是“B”或“O”，而不是“I”等。

softmax 预测实体标签的缺点是：预测实体标签时是独立的，它只由其对应 token 的输出所决定，同一序列中判断预测的多个标签也是独立的，没有关联和影响。

## VI. 属性映射

#### A. 总体架构

属性映射步骤的目的在于找到问句中询问的相关属性，转换成文本相似度问题，采用 BERT 作二分类训练模型。

在识别到了问题中的实体之后，我们根据数据库中保存的三元组，去寻找该实体对应保存的属性。此时存在两种情况。

- 1) 非语义匹配：如果所得三元组的关系 (attribute) 属性是输入问题字符串的子集 (相当于字符串匹配)，将所得三元组的答案 (answer) 属性与正确答案匹配，correct +1。
- 2) 语义匹配：利用 bert 计算输入问题 (input question) 与所得三元组的关系 (attribute) 属性的相似度，将最相似的三元组的答案作为答案，并与正确的答案进行匹配，correct +1。

#### B. SimBERT

SimBERT 是以 Google 开源的 BERT 模型为基础，基于微软的 UniLM 思想设计了融检索与生成于一体的

任务，来进一步微调后得到的模型，所以它同时具备相似问生成和相似句检索能力。

SimBERT 属于有监督训练，训练语料是自行收集到的相似句对，通过一句来预测另一句的相似句生成任务来构建 Seq2Seq 部分，然后前面也提到过 [CLS] 的向量事实上就代表着输入的句向量，所以可以同时用它来训练一个检索任务，如图 23：

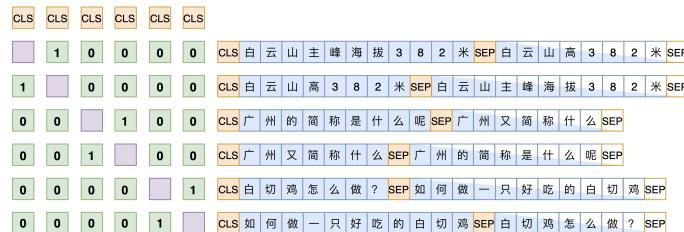


Fig. 23. SimBERT 训练方式示意图

图的左边是语义相似度部分，本质是把 batch 里面不相似的样本作为负样本，然后通过 softmax 函数计算相似样本的相似度。举例说明，上面 6 条数据中，和第 1 条样本 s1 相似的是 s2，其他都不相似，将本身 Mask 掉，那么就会得到 Mask\_1\_0\_0\_0\_0 这样的数据。

图的右边是 Seq2Seq 任务部分，将输入文本和对应的相似文本通过 SEP 进行拼接，比如现在有 sentence1 和 sentence2，那么会将数据转化成 CLS\_sentence1\_SEP\_sentence2\_SEP 这样的方式。然后根据 sentence1 去预测 sentence2。

SimBERT 通过这种训练方式，既可以完成相似文本生成任务，还可以获取文本语义向量完成相似文本检索任务。

## VII. 实验步骤

## A. 命名实体识别

**1) 数据加载:** 首先导入实验环境，导入 tensorflowhub，用于加载 bert 模型，接着导入 bert 分词器等。加载数据后查看训练集和测试集，输出如图 24 所示。

```
train_x, train_y = read_conll_format_file('./kbqa/Data/NER_Data/train.txt')
test_x, test_y = read_conll_format_file('./kbqa/Data/NER_Data/test.txt')

print(train_x[1])
print(train_y[1])
```

Fig. 24. 训练集示例

**2) 构建预处理器:** 预处理器的目的是将词序列转换成 BERT 需要的字符 id、mask id、segments id，以便后续的模型处理。

预处理器步骤如下：

- 1) 加载分词器，按字符切词。
  - 2) 构建 ge\_masks 函数获取 BERT mask id 输入。
  - 3) 构建 get\_segments 函数获取 BERT segments id 输入，第一句用 0 表示，第二句用 1 表示。
  - 4) 构建 get\_ids 函数获取 BERT 字符 id 输入。
  - 5) 构建 transform 函数将词序列转换成 BERT 需要的字符 id、mask id、segments id。
  - 6) 构建 conll\_transform 函数批量转换训练数据。
  - 7) 输入数据统一 padding 成统一长度，以矩阵的形式送入模型。

词序列转换完成后的测试输出如图 25 所示。

Fig. 25. 词序列转为 id

3) 模型构建: 定义模型类, 使用 BERT+BiLSTM 架构。直接使用 pytorch 已经实现的函数, 设置好 bert 层, 后面通过 dropout 非线性层随机失活, 然后加上双向 LSTM, 双向的隐藏层是将两个方向的直接拼接, 因此每个的长度设置为总的隐藏层输出长度的一半; 然后接线性层。

模型构建步骤如下：

- 1) 初始化预处理器，配置参数。
  - 2) 定义 bert 模型输入为字符 id, mask id, segment id。
  - 3) 加载 bert 并转化为 keras 层。
  - 4) 取 bert 的序列输出 sequence\_output, 后接 bilstm。
  - 5) 模型训练并验证。
  - 6) 对模型进行测试评估。

4) 模型预测: 经过上述步骤后, 命名实体识别模型搭建训练完成, 模型预测的结果如图 26 所示。

Fig. 26. 实体识别输出结果示例

## B. 属性映射

属性映射目的在于找到问句中询问的相关属性，转换为文本相似度问题，即采用 BERT 作二分类问题。

构造用于分类的训练集和测试集，构造测试集的整体关系集合，通过提取和去重，获得若干关系集合；每个 sample 由“问题句 + 关系属性 + label”构成，原始数据中的关系属性的 label 为 1；从关系集合中随机采样五个属性作为 Negative Samples，label 为 0，示例如下：

**1) 数据加载:** 一个 sample 由“问题 + 关系 + Label”构成，原始数据中的关系值置为 1；从 RelationList 中随机抽取五个属性作为 Negative Samples；加载数据查看训练集和测试集，输出如图 27 所示。

```
x_train, y_train = load_data("./kbqa/Data/Sim_Data/train.txt")
x_valid, y_valid = load_data("./kbqa/Data/Sim_Data/dev.txt")
x_test, y_test = load_data("./kbqa/Data/Sim_Data/test.txt")

x_train[0], y_train[0]
```

Fig. 27 训练集示例

2) 构建预处理器: 预处理器的目的是处理转换用户输入，把文字转换成对应的 id。

预处理器步骤如下：

- 1) 加载分词器，按字符切词。
  - 2) 构建 `ge_masks` 函数获取 BERT mask id 输入。
  - 3) 构建 `get_segments` 函数获取 BERT segments id 输入，第一句用 0 表示，第二句用 1 表示。
  - 4) 构建 `get_ids` 函数获取 BERT 字符 id 输入。
  - 5) 构建 `transform` 函数将句对转换成 BERT 需要的字符 id、mask id、segments id。
  - 6) 构建 `batch_transform` 函数批量转换训练数据。
  - 7) 进行测试，把字符编码成对应的字符 id、mask id 以及句段 id。

句子进行字分割和将句对转换为 id 的输出结果如图 28 所示。

Fig. 28. 句对转换为 id 输出结果

3) 模型构建: 定义模型类, 使用 BERTsim 架构。

模型构建步骤如下：

- 1) 初始化预处理器，配置参数。
  - 2) 定义 bert 模型输入为字符 id, mask id, segment id。
  - 3) 加载 bert 并转化为 keras 层。
  - 4) 模型训练并验证。
  - 5) 构建 predict\_similarity 函数预测两个句子的相似度
  - 6) 输入数据 padding 成统一长度。
  - 7) 对模型进行测试评估。

**4) 模型预测:** 经过上述步骤后, 属性映射模型搭建训练完成, 模型预测的结果如图 29 所示。

```
bert_sim = BertSim(bert_sim_config)
bert_sim.restore('kbqa/output_sim/') #只有restore()方法没有load()方法
bert_sim.predict_similarity("《机械设计基础》这本书的作者是谁?", "作者")

1/1 [=====] - 2s 2s/step
0.99578923
```

Fig. 29. 属性映射模型预测结果

### C. 问答系统

通过整合命名实体识别和属性映射两个步骤，来完成一个基于知识库的问答系统。命名实体识别通过输入问题，使用 BERT 模型得到问题中的实体，在知识库中检索出包含该实体的所有知识组合。属性映射在包含实体的知识组合中，进行属性映射寻找答案，又可分为非语义匹配和语义匹配。

1) 系统构建: 通过定义问答类来构建问答系统。

- 1) 初始化知识库，实体识别模型，语义匹配模型。
  - 2) 通过实体识别模型找出问题的实体。
  - 3) 通过属性映射模型寻找与实体相关的答案。
  - 4) 若非语义匹配，则一个知识三元组的关系属性是输入问题的子集（相当于字符串匹配），该三元组对应的答案匹配为正确答案。
  - 5) 若语义匹配，即可转化为分类问题，利用 BERT 模型计算输入问题与知识三元组的相似度，将最

相近的三元组对应的答案匹配为正确答案。

6) 进行知识库方式问答测试和 FAQ 问题答案对式  
问答测试测试。

知识库方式问答系统测试结果如图 30 所示。

Fig. 30. 知识库方式问答系统测试结果

FAQ 问题答案对式问答测试结果如图 31 所示。

Fig. 31. FAQ 问题答案对式问答测试结果

A. 本实验中 *BERT* 模型的输入输出是？

输入：BERT 接收三个输入，输入字符的 id, 代表是否有效字符的 mask id, 代表句子之间关系的 segment id。

输出：BERT 有两个输出，一个是序列的输出，对应的是每个输入序列的输出，另一个是代表整个输入文本的输出。

#### B. 属性映射模型中如何处理句对的输入？

把两个句子拼成一个句子输入，中间用‘[SEP]’分割。

C. 实体识别模型和属性映射模型在问答系统中的作用？

实体识别模型识别出问句中的实体，属性映射模型判断问句和知识三元组的属性的匹配程度，如果实体命中且属性匹配度大于阈值则可返回相应答案。

## IX. 开放题

本实验中，只处理问题中的实体出现在知识库三元组的情况，假如问题中的实体不在三元组中，但表达的是同一个实体的情况呢（如红楼梦 vs 石头记）？

要解决这个问题，可以考虑使用实体链接技术，将问题中的实体链接到知识库中的实体，以便更好地利用知识库中的信息来回答问题。

实体链接是将自然语言文本中的实体链接到知识库中的对应实体的过程。在实体链接中，首先对输入文本进行命名实体识别，提取出文本中的实体，然后通过实体消歧技术，将每个实体与知识库中的实体进行匹配。实体消歧可以使用各种技术，例如基于实体描述的相似度计算、上下文相关的实体识别和实体链接等。

一旦问题中的实体被链接到知识库中的实体，就可以利用知识库中与该实体相关的信息来回答问题，即使知识库中的信息并未直接涉及到该实体的名称。

实体链接可以结合命名实体识别和属性映射技术一起使用，以提高问答系统的精确性和效率。在属性映射中，可以根据链接到知识库中的实体，来寻找与该实体相关的属性值，并将最相关的答案返回给用户。

因此，将实体链接技术引入到问答系统中，可以更全面地利用知识库中的信息来回答问题，同时提高系统的准确性和覆盖率。

## X. 实验改进

在复现的基础上，除了前文提到的实体链接技术，我们还在实验中增加了以下改进：

**CRF**: 即条件随机场 (conditional random field) 的简称，是一种鉴别式机率模型，常用于标注或分析序列资料。在进行实体标签判断时，由于原始句法约束，LSTM 没有学习到原始的句法约束。由于 CRF 的预测效果比 Softmax 好，因此我们在此处做出了改进，将实验指导书原本的 Softmax 实体标签判断改进成 CRF 实体标签判断，使用条件随机场 CRF 层来限制句法要求，从而加强结果。

CRF 是以标签路径为预测目标，可以在 Logits 基础上为最终的预测标签序列添加一些约束，以确保预测

的实体标签序列是有效的，这些约束可以由 CRF 层在训练过程中从训练数据集自动学习。CRF 可以通过数据学习标签转移关系和一些约束条件，通过转移特征考虑输出 label 之间的顺序性，确保预测结果的有效性。CRF 层将 BiLSTM 的 Emission\_score 作为输入，输出符合标注转移约束条件的、最大可能的预测标注序列。

**Roberta**: 如图 32 所示，我们还将 BERT 替换为 Roberta。Roberta 是一种预训练语言模型，其全称为 Robustly optimized BERT approach，是由 Facebook AI Research 团队在 BERT 模型的基础上进行改进而来。Roberta 和 BERT 一样，都是使用 Transformers 架构进行训练，但是在训练过程中采用了一些新的技巧，以提高模型的性能和泛化能力。

Roberta 相比于传统的 BERT 模型，具有更好的语言建模和表示能力，在实体识别和知识库检索等任务中具有更好的性能表现。在基于知识库的问答系统中，使用 Roberta 可以提高系统的准确率和鲁棒性，并实现更高效、更智能的问答服务。

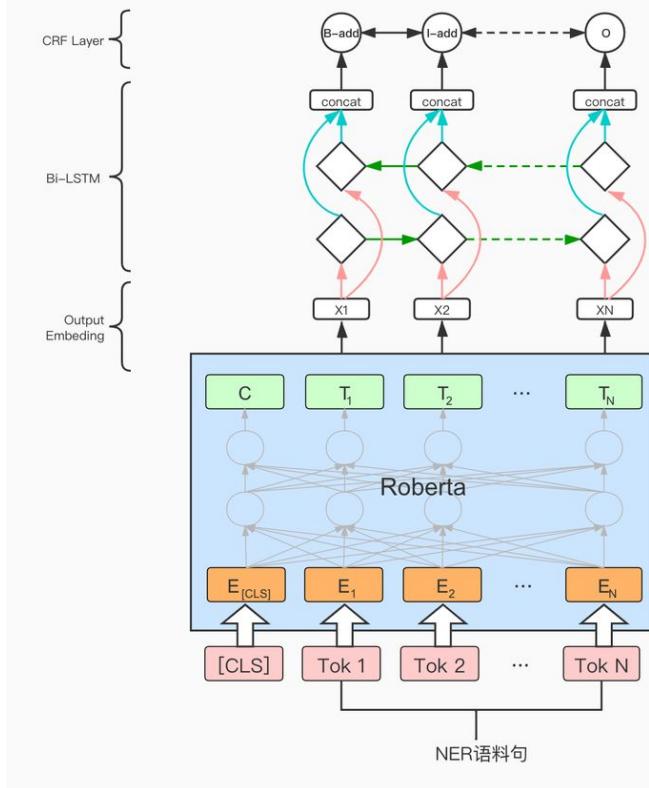


Fig. 32. 使用了 CRF 与 Roberta 的模型架构

另外，我们还希望实现多轮对话与人类反馈等功能，这将会是我们的期末目标，我们期望能够实现一个

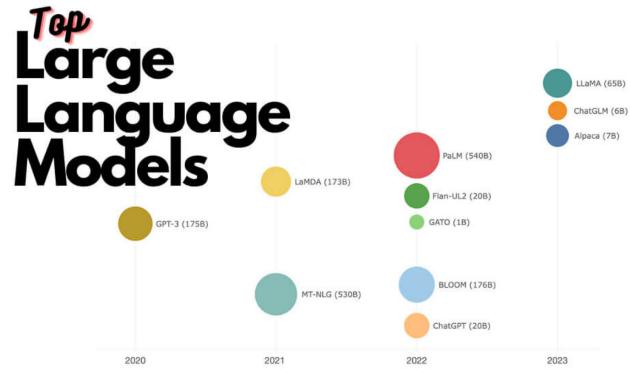


Fig. 33. LLMs

更加智能的问答系统。

## XI. 实验总结

根据实验数据和实验结果，我们得出以下结论：

- 1) 实体链接技术结合 CRF 和 Roberta 模型的引入，能够显著提高我们自动问答系统的效率和准确率。
- 2) 实验中使用的数据集 nlpcc\_kbqa\_2016 有助于我们对实验结果的分析和评估，但在实际使用中，还需要满足更多领域的实际数据，构造更为全面的知识库，才能更好地提高实际问答的效果。
- 3) 在使用该问答系统过程中，我们发现准确性和可靠性有待改进。其中，在命名实体识别的环节中，一些人名和地名等实体无法被准确识别，造成答案不准确的情况；在语义匹配环节中，一些比较复杂的问题，系统往往不能给出准确的答案。因此，在实际使用中仍需进行更多优化。

综上所述，通过该次实验，我们了解了知识库自动问答系统的基本架构和实现方法，并对问题的解决方式和应用场景有了更深入的了解。同时，在查阅相关论文后我们也意识到该系统还需要持续优化和完善，以适应更广泛的应用场景。

随着 GPT4 的公开、Meta AI 的 llama 模型权重开源……各式各样的问答模型层出不穷，我们期待在期末项目也能实现一个简易的问答系统 demo。