基于 BERT+R-net 的机器阅读理解

1160300607 张开颜
哈尔滨工业大学计算机学院
kaiyanzh@outlook.com

摘要

本次课程任务主要是在 cmrc2017 的封闭域用户提问数据集上进行机器阅读理解。经过调研，选择 BERT fine tuning 模型作为任务的 baseline，F1 值为 86.01%。由于训练集较小，首先进行了数据增强，主要包括回译和 EDA，但是受限于翻译引擎的翻译质量，数据增强的测试效果并不太明显，F1 值为 85.87%。然后在 BERT 模型后附加了简单的网络结构，包括 Context-Query Attention 和 PointerNetwork。当加入 Context-Query Attention 时，F1 值达到 87.17%；而加入 PointerNetwork，效果变差。之后的任务，主要集中在 BERT+R-net 模型上，进行了相关尝试，但是 F1 值降到 85.14%。经过分析，主要原因在于训练 R-net 的同时训练 BERT，导致网络结构很复杂，训练参数过多，出现了过拟合。为了解决过拟合，选择先在 BERT 上进行 fine tuning，然后使用 BERT fixed + R-net 模型，此时效果得到提升，达到 86.71%。最后，根据训练集的特点，进行了答案长度限制和答案择优选择，最终 F1 值达到 89.54%。

关键字：机器阅读理解；BERT；数据增强；R-net；cmrc2017

0 引言

机器阅读理解 (Machine Reading Comprehension) 指让机器阅读文本，然后回答和阅读内容相关的问题。这项技术可以使计算机具备从文本数据中获取知识并回答问题的能力，是构建通用人工智能的关键技术之一。作为自然语言处理和人工智能领域的前沿课题，机器阅读理解研究近年来受到广泛关注。

本报告主要介绍了在中文信息处理前沿知识-机器阅读理解课程中，进行的相关模型调研，模型实现过程以及优化过程和最终结果。

1 相关工作

1.1 相关数据集

在常见的 NLP 任务中，有各种各样的数据集，在某种程度上，数据集对于相关模型的提出和该领域的发展有重要作用。本部分将介绍机器阅读理解任务中常见的数据集, 从而展示机器阅读理解任务的基本模式。训练模型所用的数据可以帮助我们描述机器阅读理解任务的基本形态. 按 answer 的种类划分, 当下常见的数据集大致有 3 类:

1. **文本填词 (Cloze)** 这类数据就是在原文中扣掉一个词, 根据阅读上下文完成填词任务。
2. **完形填空 (Multi Choices)** 这类就像英语完形填空题, 给定一段 passage, 对应的每个 question 会提供 4 个选项。CMU 团队 2017 年发布的 RACE2 就是这类数据的代表。
3. **文本段落 (Spans of words)** 这类数据是现在最流行的, 其中也包含 span of words, human generated 等种类。目前很火的 SQUAD [1] 便属于这种数据集。

1.2 相关模型

随着各种数据集的不断涌出，各种模型也随之被提出，深度学习在阅读理解任务上也得到了充分的发展。近年的深度学习模型在阅读理解任务上的发展，大致如图1

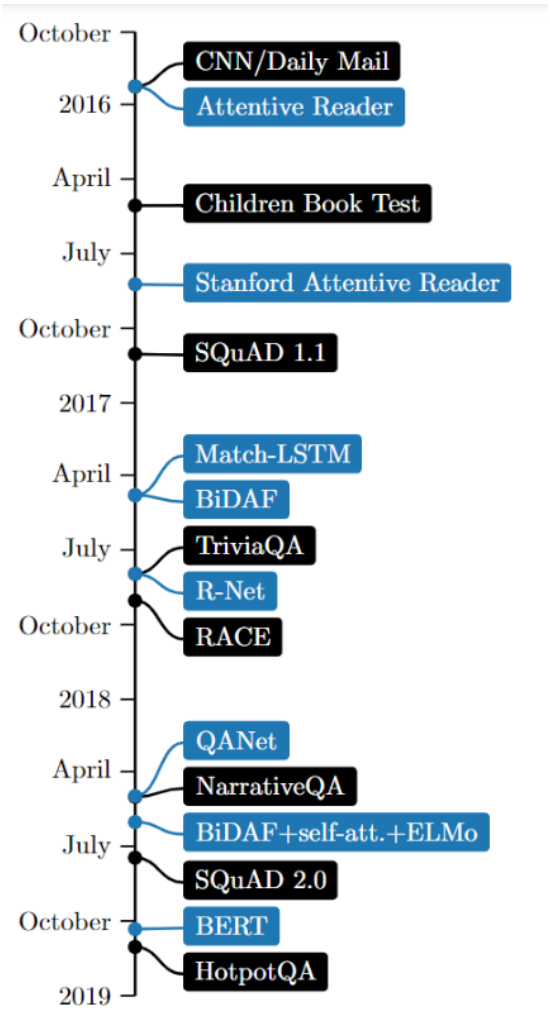


图 1: 近年深度学习机器阅读理解模型发展

1.2.1 神经网络模型开端

2015 年，Google DeepMind 的 Hermann 等人发表《Teaching Machines to Read and Comprehend》[2]，其中提出了三种神经网络模型：Deep LSTM，Attentive Reader，Impatient Reader，作为 baseline。其中各个模型的特点：

1. **Deep LSTM** 将 doc 和 query 进行拼接（doc|||query 或者 query||| doc）实际上视作一个长文本，用多层的 LSTM 来 encode，得到最后的隐藏层状态，进而进行后面的任务。

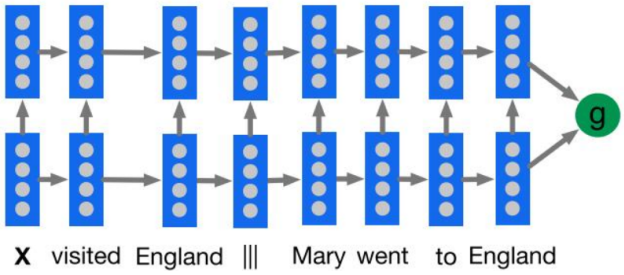


图 2: Deep LSTM

2. **Attentive Reader** 在 LSTM 的基础上，引入 attention 的概念，将 doc 和 query 分开表示。每个部分用双向的 LSTM 来 encode。Query 用两个方向上的 last hidden state 进行表示，doc 中每个 token 用两个方向的 hidden state 表示，而 doc 用每个 token 的加权来表示，其权重即为 attention。

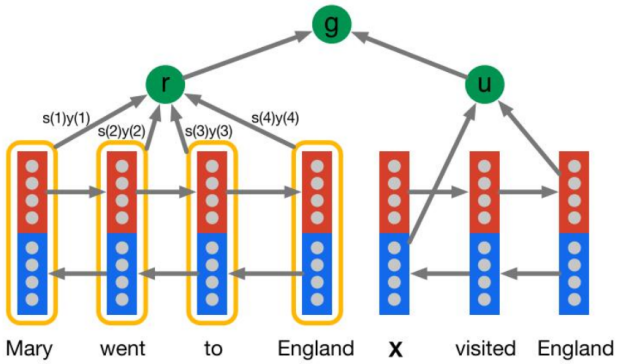


图 3: Attentive Reader

3. **Impatient Reader** 相对于 Attentive Reader 模型，不再将整个 query 考虑为整体，每个 query token 都与 document tokens 有关联。

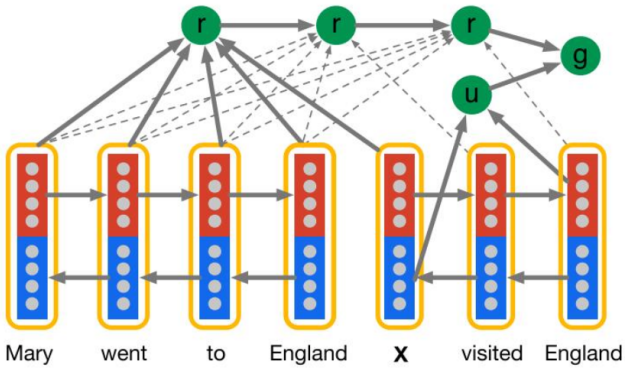


图 4: Impatient Reader

1.2.2 BiDAF

BiDAF(Bi-directional Attention Flow) [3] 在 Attentive Reader 上的主要改进如下：

1. 没有将 context 压缩到一个 fixed-size 向量，而是在每个 time step 上计算 attention。并且允许每层的向量表示能够传递到后续层，减少了信息损失。

2. 采用无记忆注意力机制，即在当前 time step 的注意力并不依赖于之前的注意力的值。

3. 使用了双向注意力机制。计算了 query-to-context（Q2C）和 context-to-query（C2Q）两个方向的 attention 信息，认为 C2Q 和 Q2C 实际上能够相互补充。

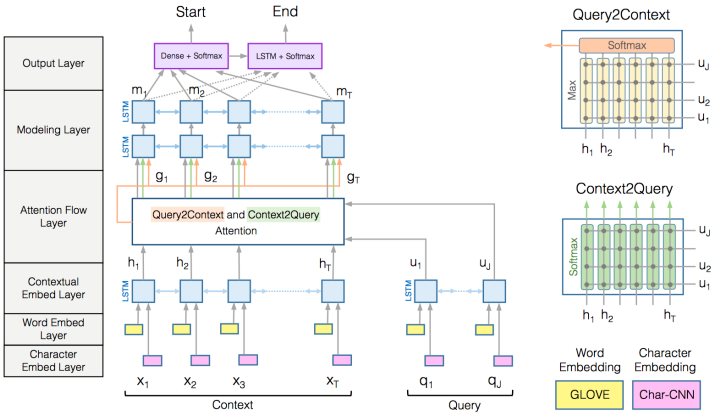


图 5: BiDAF

1.2.3 R-net

R-net [4] 是由微软提出的，在前面模型上的主要改进如下：

1. encoder 中加入 Gated Match-LSTM 模块，可以理解为通过 gate 过滤掉输入中跟问题和答案不相关的部分。

2. 增加了第三层：文章的自匹配注意力层，使得文章内部的词与词之间相互融合，通过自身上下文信息辅助筛选有价值的词，在模型效果提升中起到了很大的作用。

但是 R-net 只使用了单向的 attention，仍然有改进的空间。

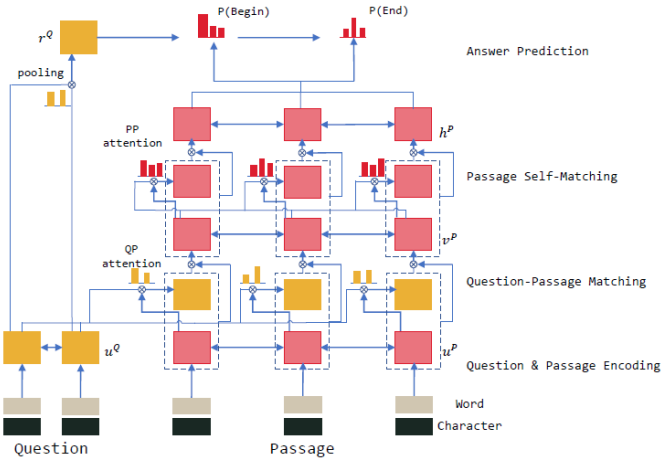


图 6: R-net

1.2.4 QA-net

QA-net [5] 是谷歌提出的一个模型。之前阅读理解的端到端模型主要都是基于 RNN 的，所以在训练和预测上运行较慢，所需资源较多。而 QANet 不需要 RNN 构成，显著地提高了训练速度和预测速度。在前面模型上的主要改进：在 Embedding Encoder Layer 和 Model Encoder Layer 中使用 encoder block, 单个 encoder block 结构自底向上依次包含位置编码 (position encoding)，卷积 (conv) 层，self attention 层和前馈网络 (fnn) 层。卷积能够捕获上下文局部结构，而 self-attention 则可以捕捉文本之间全局的相互作用。因此 QA-net 的特点是训练快，但内存需求较大。

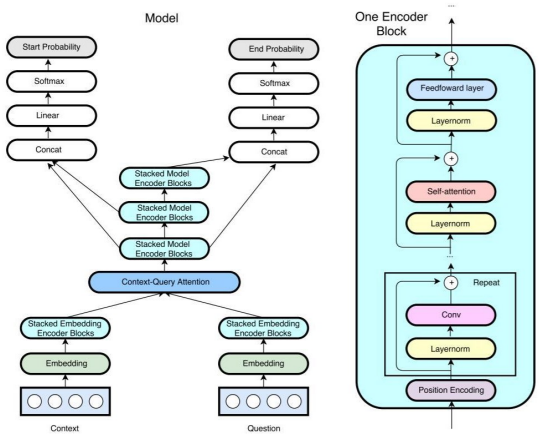


图 7: QA-net

2 模型介绍

2.1 BERT

2.1.1 Bert 模型介绍

BERT 模型 [6] 为谷歌提出的一个预训练的模型，具备广泛的通用性，刷新了很多 NLP 任务的最好性能。预训练，即用大语料训练出语言模型，将其迁移到特定任务，从而进行特征维度的迁移，句子级别信息的迁移。

BERT 模型采用 Transformer Encoder 作为语言模型，完全采用注意力机制来进行 input-output 间的计算，完全抛弃了 RNN/CNN 等结构。BERT 模型主要是在 OpenAI 的 GPT 上发展而来的，主要区别在于采用了 Transformer Encoder，也就是在每个时刻的 Attention 计算都能够得到全部时刻的输入；而 GPT 采用了 Transformer Decoder，每个时刻的 Attention 计算只能依赖于该时刻前的所有时刻的输入，因为 GPT 是采用了单向语言模型。而 BERT 使用了双向语言模型，能够同时获得上下文信息。这里，需要说一下，在最近的 GPT2.0 中，仍然采用的是单向语言模型，其效果依然很好，因此语言模型的单向和双向哪个更好，仍然有待讨论。

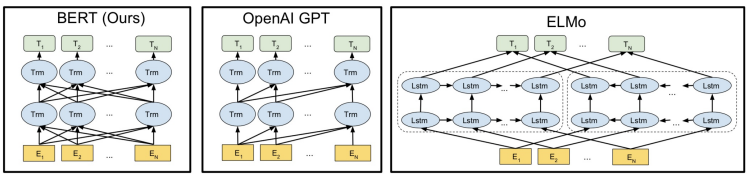


图 8: BERT 与其他预训练模型的对比

2.1.2 BERT 模型的预训练任务

为了能够有利于 token-level 任务（如：序列标注）和 sentence-level 任务（如：情感分析，问答系统等），BERT 模型采用了两个预训练任务:

(1) **Masked Language Model** 随机去掉句子中的部分 token，然后模型来预测被去掉的 token。使用该模型，可以利用双向的信息，即同时利用好前面词和后面词的概率。

(2) **Next Sentence Prediction** 很多需要解决的 NLP tasks 依赖于句子间的关系，而这个关系语言模型是获取不到的，因此将下一句话预测作为了第二个预训练任务。该任务的训练语料是两句话，来预测第二句话是否是第一句话的下一句话。

$$
\begin{aligned}
\text{Input} &= \texttt{[CLS] the man went to [MASK] store [SEP]} \\
&\quad\ \texttt{he bought a gallon [MASK] milk [SEP]} \\
\text{Label} &= \texttt{IsNext} \\
\\
\text{Input} &= \texttt{[CLS] the man [MASK] to the store [SEP]} \\
&\quad\ \texttt{penguin [MASK] are flight ##less birds [SEP]} \\
\text{Label} &= \texttt{NotNext}
\end{aligned}
$$

图 9: Next Sentence Prediction 样例

在训练过程中，将这两个任务的 loss 相加，作为总的 loss 进行优化。

2.1.3 BERT 模型 fine tuning

如图10，原论文给出了几类任务的 fine tuning 方法，主要包括：句子对分类、单句分类、QA 问答、序列标注等任务。在本次阅读理解任务中，主要采用10中 (c) 方法，将问题和文章结合作为模型输入，输出结果为文章中每个单词作为答案边界的概率。

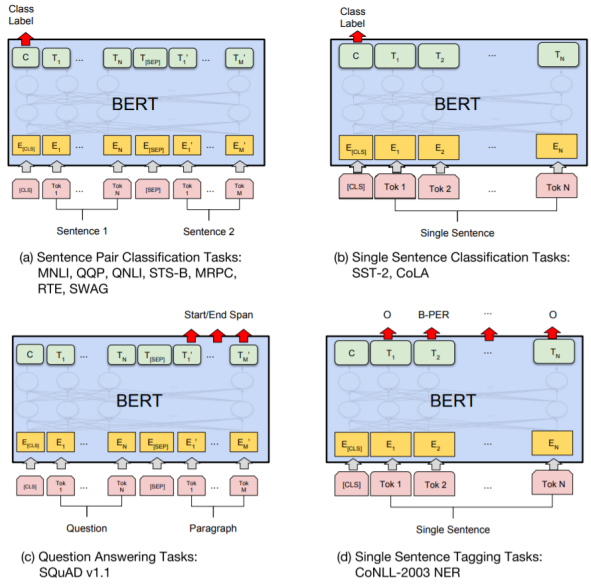


图 10: Bert fine tuning

2.2 BERT+R-net

实验过程中，R-net 部分和论文中的原始实现没有太大差别。只是将 embedding 层用了 BERT 表示，鉴于 BERT 的训练特性，因此在 embedding 时，将问题和文章一起编码，问题放在前面，文章放在后面，形式如下：

$$[CLS]Query[SEP]Context[SEP]$$

由于 BERT 训练时采用了 Predict Next Sentence 的方式，因此，将问题放在文章前面，训练时能够得到更加丰富的信息，有利于之后的答案提取。

而在 R-net 中的 Context 和 Query 的 attention 部分，分别需要 Query 和 Context 的向量表示，此时选择将 BERT 的 embedding 结果复制两份，然后使用 Query 和 Context 的 mask 信息，分别获得 Query 和 Context 的向量表示，然后进行 attention 操作。

3 实验结果及分析

3.1 实验说明

实验配置如图1

表 1: BERT 运行环境配置

环境		配置
笔记本	tensorflow-gpu=1.10	Win10 950M (4G 显存)
服务器	tensorflow-gpu=1.12	K80 (12G 显存)

3.2 数据集说明

本次任务使用的是 cmrc2017 [7] 的数据集，数据集答案主要构成是 (形容词 +) 名词。但是给定数据集已经进行了分词，并且答案是分词中的某个词语，因此鉴于分词器的性能，答案形式不唯一，如图11所示，而这将对后续的模型效果产生一定的影响。

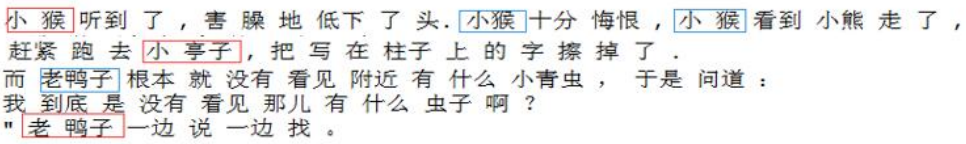


图 11: 数据集举例说明

3.3 实验过程说明

3.3.1 数据增强

实验过程中，尝试了使用回译和 EDA [8] 进行数据增强，具体如下：

1. **回译** 回译进行数据增强的主要过程是：将中文问题进行搜狗翻译 [9] 成英文，然后将获得英文问题，使用有道翻译 [10]，获得翻译之后的中文问题，最后将问题对应的原文章和翻译后所得问题作为一个样本添加到数据集中进行训练。

2. **EDA** EDA 数据增强，主要就是同义词替换、插入、交换和删除。

3.3.2 过拟合

在实验开始，使用了 BERT+R-net 模型，并且在训练 R-net 模型的同时训练 BERT 模型，训练过程中 loss 变化如图12，出现了严重的过拟合现象。

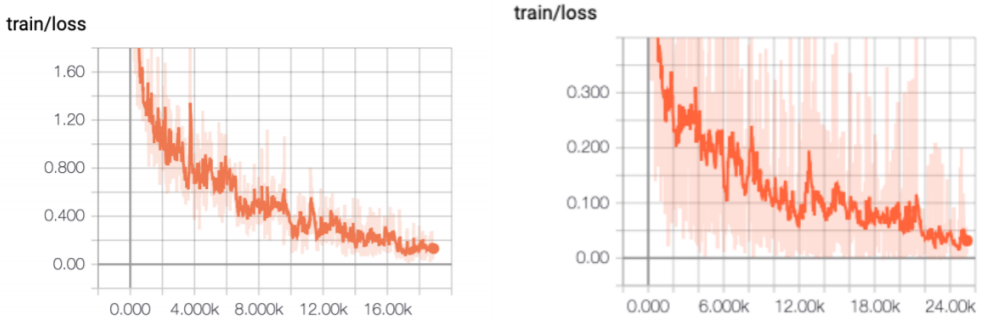
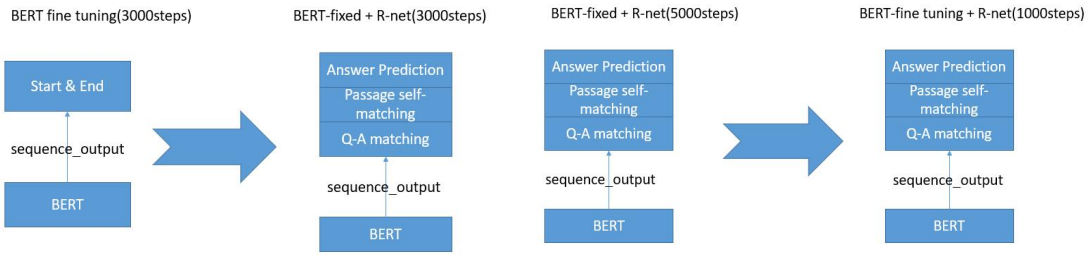


图 12: 训练过程中 loss 变化

主要原因在于网络模型过于复杂，参数过多，因此很容易过拟合。之后采用了以下两种变种模型：

1. 如图13(a)，先使用数据集在 BERT 上进行 fine tuning；然后再使用数据集在 BERT+R-net 上进行训练，此时，将 BERT 固定，使用之前在数据集上 fine tuning 的参数，并且不再进行训练。
2. 如图13(b)，使用数据集训练 BERT+R-net，此时，将 BERT 固定，使用 Google 给出的模型参数进行训练；训练一段时间之后，不再固定 BERT，而是将 BERT 与 R-net 同时进行训练。



(a) BERT+R-net1　　(b) BERT+R-net2

图 13: BERT+R-net 模型变种

3.3.3 后处理

通过分析数据集获得答案长度分布，如图14所示。可以发现长度为 2 的答案最大，并且最长答案长度为 5。因为，R-net 输出的是文章中各个位置作为答案起点和终点的概率，因此，最终答案是随机组合获得的，所以可能出现 top1 答案的长度非常长，缺少限制。为了提高最终测试性能，对 R-net 输出的候选答案进行后处理，选择长度小于 5 的最优答案。

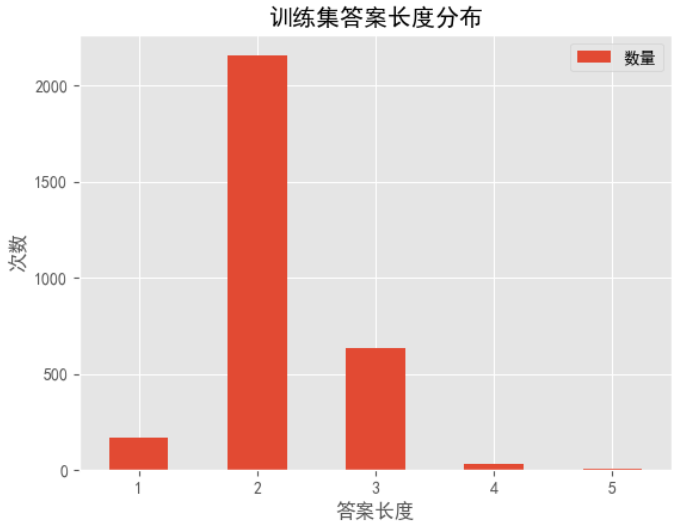


图 14: 答案长度分布

3.4 实验结果及分析

最终实验结果如图2所示。

表 2: 在测试集上结果

模型	召回率 (%)	准确率 (%)	F1 值 (%)
BERT baseline	86.809	88.479	86.941
+DA	86.151	86.841	84.166
+Context2Query	87.365	88.237	**87.752**
+Answer Limit	87.315	88.546	**87.382**
+R-net1	84.762	86.233	85.488
+R-net2	86.908	88.401	86.892
total	88.532	90.776	**89.067**

对实验结果分析如下：

1. **DA** 数据增强，包括回译和 EDA 两种手段。其中 EDA 的例子如图15所示，可以发现结构比较混乱，因为效果较差，因此最终并没有采用这种方法。

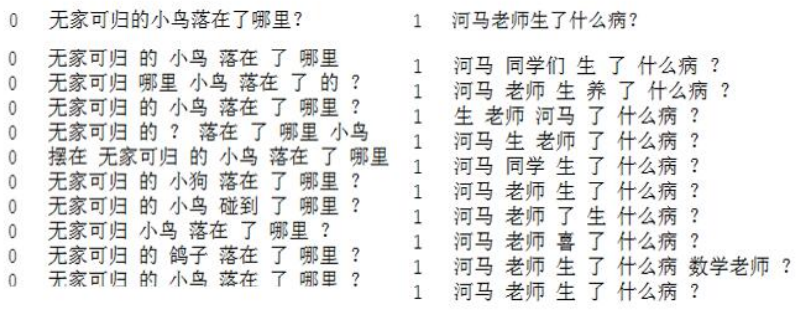


图 15: EDA 举例

实验结果中的数据增强是回译的测试结果。回译的测试效果在 baseline 的基础上下降了，主要在于翻译的效果受限。部分回译结果如表3。可以发现绝大多数的回译结果，都是引入了结构变化，进行语句主干成分的调整。但是，在人名和事件名回译中引入了较大的噪声，会导致之后的答案查找出现较大误差。

表 3: 回译举例

原问题	回译结果
国与国之间会把谁作为人质？	谁将在国与国之间成为人质?
羊和狼的盟约以什么作为保证？	羊和狼结盟的保证是什么?
小兔和谁是好朋友？	谁是兔子的好朋友?
我最好的朋友是谁？	谁是我最好的朋友?
美丽的湖泊是什么构成的？	什么构成了一个美丽的湖?
老鼠发现小朋友小毛在干什么？	当老鼠发现小男孩的头发时，它在做什么?
伍奢的第二个儿子是谁？	吴舍是谁的二儿子?
吕后指定谁为相国？	吕侯任命谁为宰相?
第三次中东战争前夕，洛茨来到了哪里？	在六日战争前夕，罗兹到哪里去了?
树林中有什么？	那里是什么? 在树林里。
山洞里住着一只什么？	那里住着什么? 里面是一个洞穴

同时，回译的方式有待商榷，仅仅对问题进行回译，可能意义不是很大，需要进行更细致的甄别和思考。

2. **Context2Query** 在 BERT baseline 上引入了文章对问题的 attention，主要是对于文章中的每个词求对问题中每个词的 attention，然后通过计算得到的信息获得文章的向量。在这个实验中，测试结果在 BERT baseline 上获得了微小的提升。主要原因在于 BERT 本身使用了大量的 attention 机制，而再加入 attention 机制后，起到的作用不是很大。

3. **Answer Limit** 在对标准答案进行了分析之后，选择对于 top N 的答案进行额外分析，选择在最大答案长度限制内的最优答案最为最终答案。由于 BERT 的最终结果进行了概率组合，因此部分答案可能不是很规范，会很长，因此进行了答案长度限制后，效果得到的提升。

4. **R-net1** 在这个模型中，同时训练了 BERT+R-net，导致网络模型很复杂，因此出现了严重的过拟合，最终测试效果不太好，比 BERT baseline 差。

5. **R-net2** 为了解决模型复杂过拟合的问题，对 BERT+R-net 进行的修改，不进行同时训练。此处选择，先使用训练集对 BERT 进行 fine tuning。然后在 BERT+R-net 模型中，固定 BERT，并且使用 fine tuning 时的模型参数进行初始化，以减少模型训练量。最终测试效果和 BERT baseline 相当。

6. **total** 将最终所有的增益模型进行相加测试，测试结果得到提升。

3.5 可视化应用

在课程中，为了更好的展示实现效果，进行了可视化应用开发。应用主要包括问题获取、问题理解、问题回答三部分。其中，问题获取部分，支持用户进行文章选择，并且使用了讯飞的语音识别 API，进行用户的语音提问到文本的转化；问题理解部分，调用 BERT+R-net 模型，进行阅读理解和答案预测；问题回答部分，将模型的答案在原文中标出，同时调用了讯飞的语音合成 API 进行文本到语音的转换，最终反馈给用户。

图 16: 可视化应用截图

4 结束语

4.1 总结

在本次课程中，主要使用了 BERT 模型完成机器阅读理解任务，在整个过程中，循序渐进，收获颇丰。课程开始，对于机器阅读理解任务知之甚少，也不了解 BERT 模型，花了较长的时间进行技术调研，然后阅读 BERT 代码和相关论文，逐渐增加了对 BERT 的理解，同时也能够使用 BERT 搭建阅读理解任务的 baseline，并且效果比较理想。课程的中间阶段，主要进行了模型的改进。由于在以前的学习中基本都是使用的 Keras 进行的神经网络模型搭建，而课程中使用 Tensorflow 不是很熟练，因此遇到了很多问题，花费了较长时间进行相关资料的查询和学习。幸运的是，最终完成了在 BERT 基础上搭建其他神经网络模型的任务，实现了 BERT+R-net 的模型，并且之后在老师和学长的建议下，进行了相关改进，后续过程中，进一步增加了对于模型的理解，也学习到了很多新知识和技能。虽然最终改进效果没有达到我的预期，但是在这个过程中，实践的经验是是宝贵的，学习到的知识是无价的。

总的来说，通过这次课程，坚定了我在 NLP 道路上前进的信念，明确了前进的方向，对我之后的学习和研究有着指导性的意义。最后，感谢老师们和助教在课程中的指导和建议，同时也感谢我的队友的支持、帮助和信任！

4.2 下一步

(1) **BERT + R-net** 在解决 BERT+R-net 的过拟合问题时，可以选择不同时训练 BERT 和 R-net。课程中，使用了 cmrc 数据集预训练 BERT 模型，之后使用 BERT 模型提取 embedding，输入到 R-net 中进行训练。不同时训练 BERT 和 R-net，可以降低网络模型的复杂度，有效避免过拟合。在以后的学习中可以采取前文中提到的第二种训练方式，即使用 google 提供的 BERT 预训练模型，在训练 BERT+R-net 时不改变 BERT 模型的参数，训练一段时间后，再开启 BERT 模型的参数训练，这种训练方式理论上和前一种方式差不多。

(2) **BERT + +** 在课程中，主要使用了 BERT+R-net 模型。而实际上，目前在 SQUAD [11] 上，高居榜首的是 BERT+AoA，在某种程度上，效果超过了人类。在以后的学习中，可以研究 Attention Over Attention [12] 模型，进行相关的实验。

(3) **ERNIE 模型** BERT 是基于字进行训练的，但是中文一般是以词为单位的。因此，单独使用基于词的训练，可能会失去词的语义知识。在不久以前，百度在 BERT 模型基础了，在中文语料上，基于词进行了训练 ERINE [13]。ERNIE 通过建模海量数据中的词、实体及实体关系，学习真实世界的语义知识。相较于 BERT 学习原始语言信号，ERNIE 直接对先验语义知识单元进行建模，增强了模型语义表示能力。训练数据方面，除百科类、资讯类中文语料外，ERNIE 还引入了论坛对话类数据，利用 DLM（Dialogue Language Model）建模 Query-Response 对话结构，将对话 Pair 对作为输入，引入 Dialogue Embedding 标识对话的角色，利用 Dialogue Response Loss 学习对话的隐式关系，进一步提升模型的语义表示能力。ERNIE 在自然语言推断，语义相似度，命名实体识别，情感分析，问答匹配 5 个公开的中文数据集合上进行了效果验证，ERNIE 模型相较 BERT 取得了更好的效果。在以后的学习中，可以尝试使用 ERNIE 模型。

参考文献

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[2] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[3] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[4] W Wang, N Yang, F Wei, B Chang, and M Zhou. R-net: Machine reading comprehension with self-matching networks. *Natural Lang. Comput. Group, Microsoft Res. Asia, Beijing, China, Tech. Rep*, 5, 2017.

[5] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Yiming Cui, Ting Liu, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. Dataset for the first evaluation on chinese machine reading comprehension. *arXiv preprint arXiv:1709.08299*, 2017.

[8] Jason W Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[9] 搜狗机器翻译. 深智引擎. https://deepi.sogou.com/.

[10] 有道机器翻译. 有道智云. https://ai.youdao.com/gw.s.

[11] Stanford NLP Group. Squad2.0, the stanford question answering dataset. https://rajpurkar.github.io/SQuAD-explorer/. 2019.

[12] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.

[13] Baidu Paddle Paddle. Ernie. https://github.com/PaddlePaddle/LARK/tree/develop/ERNIE. 2019.