非对称文本匹配实验报告

郑明钰

北京师范大学人工智能学院

摘　要：非对称文本匹配问题是指两个内容表述和篇幅具有较大差异但语义相似的文本之间的匹配任务，本次实验的任务是根据给定的（question，sentence）样本，判断 sentence 中是否包含 question 的答案,其中 question 和 sentence 在长度和语义表述上差别较大,即非对称性。针对该问题，本文首先采用机器学习的方法，构建两个特征：“文本相似度”和“共享关键词指数”，利用多种模型进行训练和预测，得到的最好的精确度为 0.75。更进一步可以使用深度学习模型如 BERT 进行改善。

关键词：非对称文本匹配，文本相似度，共享关键词指数，机器学习，BERT；

0 引　言

自然语言推断（NLI-Natural Language Inference）是自然语言处理（NLP-Natural Language Processing）的重要组成部分，其目标是在给定一个“前提”（premise）的情况下，判断一个“假设”（hypothesis）是“正确”、“错误”还是“中立”的[1]，在信息检索、问答系统、对话系统等问题中有广泛应用。

本次实验中的任务属于“文本对的分类问题”（sentence pair classification tasks），需要判断 sentence 中是否包含 question 的答案，如果包含则将该（question，sentence）样本标签为 1，否则为 0。更高层次的推断任务还需要根据所给的 sentence 判断 questi-on 的答案是 yes 或 no，或者从 sentence 中选择出包含问题答案的语句范围。

国内外学者对于文本匹配、问题回答等自然语言处理问题有大量的研究，也有很多经典的数据集。 斯坦福大学提出了 QNLI（Question Natural Language Inference）数据集[2]，也就是 SQuAD1.1 版本[3]，以往的数据集针对每一个问题给出的是一系列候选答案，如 MCTest[4]，而 SQuAD1.1 需要的是系统从给出的文本中选择出包含答案的语句范围；SQuAD2.0 在之前的基础上进行了拓展，包含了给定文本并不含有对应问题答案的可能性。Alex Wang 等提出了用于评估模型表现的基准测试和分析平台 GLUE[5]（General Language Understanding Eva-luation benchmark)，可以评估模型在多个任务上的泛化能力，其中包含了多个不同自然语言理解任务的数据集。针对文本匹配任务，中科院计算所 Guo 等实现了一个通用的深度文本匹配的工具 MatchZoo[6]，将许多深度匹配模型如 DRMM、ARC-I、DSSM 等封装成统一的接口，主要解决文档检索、问题回答、会话应答排序、同义句识别等问题。

深度学习在文本匹配领域可以发挥巨大的作用，在 SQuAD2.0 的排行榜上位居前列的无一不是采用了神经网络的模型。对于非对称文本匹配问题，本报告先采用机器学习方法进行解决，然后进一步考虑使用深度学习模型进行尝试。

1 机器学习方法

首先进行特征工程，尝试从所给的（question，sentence）文本中构建特征来进行模型训练。本文构建了两个特征来尝试衡量 sentence 和 question 的相关程度，从而帮助预测 sentence 中是否包含 question 的正确答案，两个特征分别为：“文本相似度”和“共享关键词指数”，下面分别进行叙述。

1.1 特征一：文本相似度

“文本相似度”指两段文本之间在语义上的相似程度，需要先对文本进行向量化，然后可以选择多种方式进行相似度计算，如余弦相似度、Jaccard 相似度等。

本文基于 word2vec[7]词向量化技术进行文本向量化。其一般做法是对文本分词后，提取其关键词，用词向量表示这些关键词，接着对关键词向量求平均或者将其拼接，最后利用词向量计算文本间的相似度。这种做法的缺点是丢失了文本中较为重要的语序信息，包含的语义信息也较为局限，但相比于用词袋模型表示文本仍然是很大的改善。

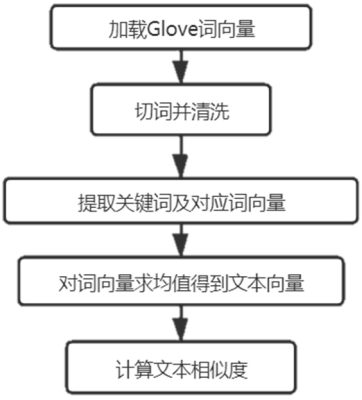

图 1.“文本相似度”特征构建过程

在具体实现上，本文首先提前下载好 50 维的 Glove 词向量文件，然后利用 nltk 中的 tokenizer 对文本进行切词，对分词结果进行进一步清洗，删去停用词和标点符号，然后使用 jieba 中的 extract_tags 方法提取关键词。由于 question 的文本长度远远短于 sent-ence，故在 question 中提取 3 个关键词，在 sentence 中提取 10 个关键词，然后利用 Glo-ve 词向量文件得到关键词的词向量，求均值得到文本向量，然后计算余弦相似度作为 question 和 sentence 的相似度，从而得到特征一：“文本相似度”。计算公式如下，其中 vec1 和 vec2 分别为 question 和 sentence 的文本向量。

$$\text{cosine\_similarity} = \frac{vec1 * vec2}{\|vec1\|\|vec2\|}$$

在得到“文本相似度”特征以后，本文利用机器学习模型进行训练，验证“文本相似度”特征对于解决非对称文本语义匹配的效果，结果如下：

表 1.使用“文本相似度”的分类结果

模型	Logistic 回归	决策树	SVC	RFC	朴素贝叶斯
准确率	0.5588	0.5181	0.5693	0.565	0.5587

可以看出“文本相似度”特征对于帮助判断 sentence 中是否含有 question 的答案有一定作用，但是效果并不是很好。因此考虑增加第二个特征

1.2 特征二：共享关键词指数

“文本相似度”的计算会受到 question 和 sentence 中所有关键词的影响，有些不重要的关键词的词向量会导致文本向量出现误差，导致本来相似性较大的两个文本之间的相似度反而较小，因此本文构建特征“共享关键词指数”，通过 question 中有多少关键词也在 sentence 的关键词列表中来衡量两者的相似程度。

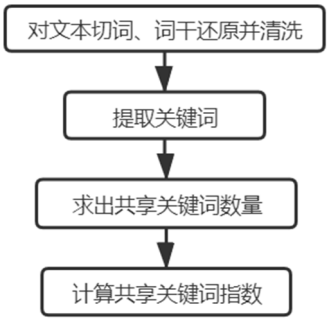

图 2.“共享关键词指数”特征构建过程

为了更好地捕捉共有的关键词，针对 question 提取 m 个关键词，针对 sentence 提取 n 个关键词。设两者的共享关键词有 t 个，则“共享关键词指数”为：

$$\text{common\_index} = \frac{t}{m}$$

对于关键词的提取数量本文多次进行了尝试，观察到更多的关键词数量对于预测结果有明显的积极作用，最终选择从 question 中提取 5 个关键词，从 sentence 中提取 30 个关键词。为了进一步减少 question 和 se

-ntence 的非对称性带来影响，在计算“共享关键词指数”时，又通过 wordnet 考虑 ques-tion 中的关键词的近义词，如果 question 的近义词中有和 sentence 中的关键词相匹配的，则也记入共享关键词内。但是这一优化并没有给结果带来明显的提升。

最终利用“文本相似度”和“共享关键词指数”两个特征进行模型训练，得到了明显优于一个特征进行训练的结果，如下所示：

表 2.最终结果

模型	Logistic 回归	决策树	SVC	RFC	朴素贝叶斯
准确率	0.7463	0.6541	0.75	0.7381	0.7331

机器学习方法相对于深度学习方法，需要人为构造特征，较为耗费精力，特征工程的质量对于最终的结果有着非常重要的影响。在未来优化方向上，可以考虑构造更多更精细的特征，进一步挖掘 question 和 sent-ence 在语义上的联系，如 PranavRajpurkar 在利用机器学习模型在 sentence 中选择答案范围时就使用了匹配的二元词组频率、依存关系树等 7 个特征[2]。

2 深度学习方法：BERT

一般来说，深度学习方法在自然语言处理中的表现明显优于传统的机器学习方法。针对文本匹配问题，有许多的神经网络模型可以使用，如 BERT、DRMM、MatchPyram-id、 MV-LSTM、ARC-I、ARC-II、DSSM 和 CDSSM 等。

BERT（Bidirectional Encoder Represen-tations from Transformers）模型和 word2v-ec 一样都是一个语言表示模型[8]，但其在 NLP 领域的多个方向都取得了更优的结果，包括 QNLI 领域。BERT 模型的本质是通过在海量语料的基础上运行自监督学习方法为单词学习到一个好的特征表示，它以一些文本分词后的序列作为输入，针对每一个输入的词都产生一个向量表示。

BERT 的使用方式一般有两种，“featur-e extraction”方式是指我们仅用 BERT 来从文本中提取特征，将 BERT 的输出作为另一个模型的输入；另一种方式为“fine-tuning BERT”，指在 BERT 模型的上端增加新的层，然后将 BERT 模型和新增加的层作为一个整体进行训练，同时也会对原本 BERT 模型中的权重进行微调（fine-tuning），比如增加一个线性层用于分类等。

以“Fine-tuning BERT”方式使用 BERT，针对不同的任务，需要稍微改变输入的格式，同时使用不同的输出解决问题。Fine-tuning BERT 对应的有 4 种不同的任务形式：“文本对分类问题”、“单段文本分类问题”、“问答系统”和“文本标签”。 非对称文本匹配任务属于“文本对分类问题”，网络结构和模型使用如下，本文采用 BERT-base-uncased 基础模型。

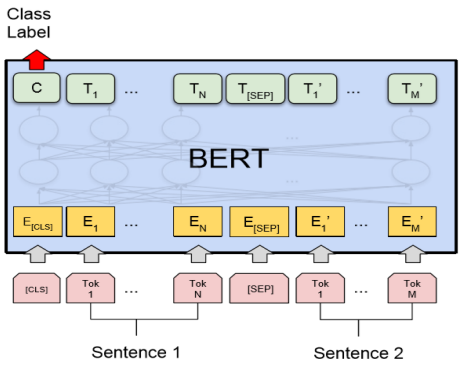

图 3.用于文本对分类的“Fine-tuning BERT”

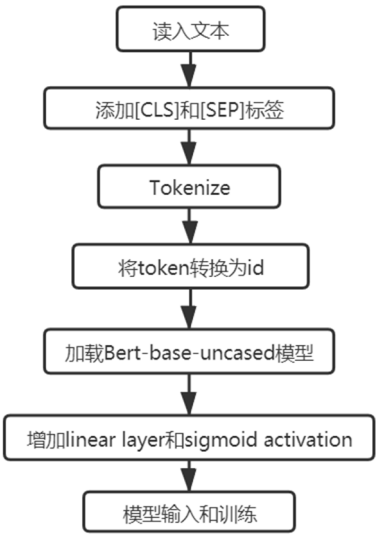

图 4.Bert 模型使用

在具体实现的过程中，我们需要将 que

-stion 和 sentence 先进行 tokenize，在 question 前添加“[CLS]”token，表明为分类任务，在 question 的末尾添加“[SEP]”token，表明将输入的两部分分开，之后再将每一个 token 转换为 BERT 的 tokenizer 词库中对应的 id，然后整个序列作为模型的输入。在模型训练完成之后，将“[CLS]”token 的输出作为线性层的输入，其后再跟随一个 sigmoid 激活函数，完成最终的分类任务。

用于文本对分类的 BERT 神经网络的输入结构如图 5 所示。

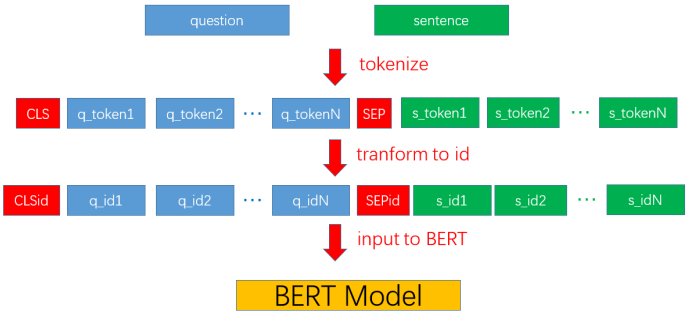

图 5.BERT 的输入处理过程

使用 BERT 神经网络模型进行训练的时间较长，需要较大空间的 GPU 显存，MatchZoo 中关于 BERT 的教程中的训练时间约为 7 小时，本文在进行实验的时候也遇到了“cuda out of memory”的问题，无法完成 BERT 神经网络的训练，期待未来利用服务器进一步进行优化。

3 总结

本文对于非对称文本匹配任务从机器学习和深度学习两个方面进行尝试：

在使用机器学习解决任务的时候，明显体会到特征工程对于机器学习模型的预测结果是非常重要的，但是构建出好的特征需要丰富的自然语言处理的知识，相比之下本文构建的用于机器学习的两个特征较为简陋。

在使用 BERT 神经网络的过程中我参考了 Dima Shulga 博客[9]，相比于机器学习模型，神经网络模型不需要进行庞大的特征工程，只需要将数据处理为正确的形式即可。针对某一个具体问题，对已有的神经网络进行略微修改就可能取得较为满意的结果，在操作上的繁琐程度也较低，但是复杂的神经网络所需的硬件和训练时间却是很大的代价。

本次实验让我收获了更多有关自然语言处理的知识，对于文本匹配任务有了进一步的了解，最重要的是锻炼了代码能力。

参考文献

[1] NLP-progress. http://nlpprogress.com/english/natural_language_inference.html

[2] Rajpurkar, Pranav & Zhang, Jian & Lopyrev, Konstantin & Liang, Percy. (2016). “SQuAD: 100,000+ Questions for Machine Comprehension of Text.” 2383-2392. 10.18653/v1/D16-1264.

[3] The Stanford Question Answering Dataset.https://rajpurkar.github.io/SQuAD-explorer/

[4] Richardson, M. & Burges, C.J.C. & Renshaw, Erin. “MCTest: A challenge dataset for the open-domain machine comprehension of text.” EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. 193-203.

[5] Wang, Alex et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.” ArXiv abs/1804.07461 (2018): n. pag.

[6] Jiafeng Guo et al. “MatchZoo: A Learning, Practicing, and Developing System for Neural Text Matching.” In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’19). Association for Computing Machinery, New York, NY, USA, 1297–1300.

[7] Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

[8] Devlin, Jacob et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” NAACL-HLT (2019).

[9] Dima Shulga. “BERT to the rescue!”. https://towardsdatascience.com/bert-to-the-rescue-17671379687f