山东大学学报(理学版) 2023 年 1 月 第 58 卷 第 1 期
Journal of Shandong University(Natural Science), Vol.58, No.1, 2023

E-mail: xblxb@sdu.edu.cn
http://lxbwk.njournal.sdu.edu.cn
Tel: +86-531-88366917

文章编号: 1671-9352(2023)01-0040-08 DOI: 10.6040/j.issn.1671-9352.1.2021.048

融合分段编码与仿射机制的相似案例匹配方法

赖华[1,2],张恒滔[1,2],线岩团[1,2*],黄于欣[1,2]
(1.昆明理工大学信息工程与自动化学院,云南 昆明 650500; 2.昆明理工大学云南省人工智能重点实验室,云南 昆明 650500)

摘要: 相似案例匹配任务旨在判断2篇裁判文书所描述的案件是否相似,通常被看作裁判文书的文本匹配问题,在司法审判过程中具有重要的应用。现有深度学习模型大多将案例长文本编码为单一向量表示,模型很难从长文本中学习到裁判文书之间的细微差异。考虑到案例文本各部分的内容较为固定,本文提出将案例长文本拆分为多个片断并分别编码,以便获取不同部分的细微特征;同时,采用可学习仿射变换改进相似度打分模块,使模型学习到了更多细微的差异,进一步提高了案例匹配的性能。在CAIL2019-SCM数据集上的实验结果表明,本文提出方法与现有方法相比准确率提升了1.89%。

关键词: 相似案例匹配; 文本匹配; 法律智能; 卷积; 仿射变换
中图分类号: TP391 **文献标志码:** A
引用格式: 赖华,张恒滔,线岩团,等.融合分段编码与仿射机制的相似案例匹配方法[J].山东大学学报(理学版),2023,58(1):40-47.

A similarity case matching method combining segment encoding and affine-mechanism

LAI Hua[1,2], ZHANG Heng-tao[1,2], XIAN Yan-tuan[1,2*], HUANG Yu-xin[1,2]
(1.Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, Yunnan, China; 2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, 650500, Yunnan, China)

Abstract: Similarity case matching (SCM) task is to judge whether the cases described in two judgment documents are similar. SCM is usually regarded as the text matching problem of judgment documents and has important applications in the judicial trial. Existing deep learning models mostly encode long texts of cases into a single vector, and it is difficult for the model to learn the subtle differences between the cases from long texts. Considering that the content of each part of the case text is relatively fixed, this paper proposes to split the long case text into multiple pieces and encode them separately to obtain the subtle features of different parts. At the same time, learnable affine-transformation is used to improve the similarity scoring module, so that the model learn more subtle differences, which further improves the performance of case matching. The experimental results on the CAIL2019-SCM data set show that compared to another model, the accuracy of the method proposed in this paper have increased by 1.89%.
Key words: similarity case matching; text matching; legal intelligence; convolution; affine transformation

0 引言

随着数字化信息的发展,越来越多的裁判文书被以电子文本的形式储存,通过人工检索的方式费时费力,借助机器实现裁判文书的自动匹配能够降低时间和人力成本,加快法院审判进程。相似案例匹配是文本匹配在司法领域的运用,文本匹配在自然语言处理(natural language processing)诸多任务上得到了很好的应

收稿日期: 2021-09-01; 网络出版时间: 2022-11-04 11:32:02
网络出版地址: http://kns.cnki.net/kcms/detail/37.1389.N.20221103.1727.004.html
基金项目: 国家自然科学基金资助项目(61966020);国家重点研发计划资助项目(2018YFC0830104, 2018YFC0830105, 2018YFC0830100);云南省基础研究计划资助项目(202001AT070046)
第一作者简介: 赖华(1966—),男,硕士,副教授,研究方向为智能信息处理、复杂过程建模与控制.E-mail: 405904235@qq.com
* 通信作者简介: 线岩团(1981—),男,博士研究生,副教授,研究方向为自然语言处理、信息抽取、机器翻译.E-mail: xianyantuan@qq.com

用,例如在信息检索、对话与问答、搜索引擎、推荐系统等,这些任务从某种程度上来说都可以看作是文本的匹配任务,通常以文本相似度计算、文本相关度对比等形式呈现,相似案例匹配中的核心技术也是文本匹配。

在利用机器进行文本匹配的初期,主要是通过一些人为设计的特征进行规则的匹配,例如: Hall 等[1] 提出的字符串近似匹配算法利用文本关键词进行字符串相似度计算; Salton 等[2] 利用词频特征得到文本的表示后基于该特征计算相似度; Huang 等[3] 将关键词的语义信息与词频特征相结合后用于相似度计算。以上基于特征进行匹配的方法往往是根据任务需要而人工定义特征,匹配质量与文本自身质量有关,而且这样的匹配方法在很大程度上限制了模型的泛化能力。

后来出现了基于浅层语义空间中的向量表示进行匹配的方法,Niraula 等[4] 和 Wang 等[5] 利用主题模型对文本建模后,通过计算表示向量之间相似度来进行匹配,然而在浅层语义空间进行的匹配在遇到复杂的语义信息时显得效果欠佳。

神经网络被提出后,文本有了更高级的表示(词嵌入[6] 和文档嵌入[7]) ,再加上深度学习的快速发展,通过深度学习模型自动获取融合更多上、下文语义信息的特征来进行匹配成为了主要的方法,具体来说大概可以分为 2 类: 一类是以孪生网络结构为代表的匹配模型,例如 Mueller 等[8] 提出了 Siamese-LSTM 模型,该模型使用孪生的 LSTM 对文本对进行编码后计算二者的相似度; Reimers 等[9] 提出 Sentence-BERT(SBert) 模型,编码器采用了预训练的 BERT[10] ,将编码向量拼接后经过 Softmax 分类输出。另一类是基于文本交互的匹配模型,例如 Wang 等[11] 提出的 BiMPM(bilateral multi-perspective matching) 模型在在匹配之前将 2 段文本进行交互,并提出了 4 种具体的交互策略,最后对交互后的向量进行整合分类。2 类模型在一些短文本匹配数据集[12-13] 上取得了不错的效果。

然而在长文本匹配上,以上方法的效果并不好,长文本具有篇幅较长的特点,并且是在同一案件主题上的相似案例匹配,例如在针对民间借贷的 CAIL2019-SCM[14] 数据集中,最长的案件文本多达 1 062 个字符,平均长度达 679 个字符。将案例长文本编码为单一向量表示,细粒度的匹配信号是非常稀疏的,噪声比较大,难以捕捉到有用的匹配信息。为此本文提出了一种分段的编码方式,在缓解长文本编码效果不佳的同时,对长文本进行了更细粒度的语义信息提取。

分段编码能改善长文本编码问题,但是在分段后需要通过交互来整合信息。裁判文书从结构上来说具有相同的形式,在主题上也是相同的,就民间借贷案件而言,文书中存在大量重复的词汇: 原告、被告、借款、利息、支付等,使得文书之间的差异变得很小。现有的深度匹配模型通过将文本表示整合为某一固定向量后计算相似度以实现匹配,但在聚合过程中由于缺乏更好的交互,导致模型难以学习到细微的差异。针对这一问题,本文基于仿射变换机制设计了一个可学习的打分模块,在文本交互的同时计算文本相似度得分。

基于现有研究,本文提出一种融合分段编码与仿射机制的相似案例匹配方法。主要贡献如下:

1) 提出一种分段处理案件长文本的编码方法。分段编码裁判文书,能够有效缓解长文本编码时数据稀疏导致匹配效果不佳的问题。

2) 提出了一种基于仿射变换的文档交互方法。基于仿射变换构建打分器,通过与查询文书的交互来对候选裁判文书进行相似度得分计算。

3) 在 CAIL2019-SCM 数据集上设计实验,验证了本文方法在民间借贷相似案件匹配上的有效性。结果表明,本文方法比基线模型和一些深度文本匹配模型得到了更好的效果。

1　相关工作

深度匹配模型从提出深度结构的语义模型(deep structured semantic model, DSSM) [15] 到孪生结构的 SBert 模型的出现,这种“双塔式”的深度学习模型在文本匹配任务上取得了不错的效果,它们的结构共有特点是基于孪生网络(siamese network) [16] 设计的,都是通过参数共享的形式对输入的文本进行编码,这种结构在文本匹配过程中具有天然的匹配速度优势,因此本文同样沿用了这种结构。

DSSM 模型主要针对查询文档和候选文档的相似度来建模,整个模型由输入层、表示层和匹配层构成,在表示层利用了 3 层非线性网络进行特征提取。DSSM 特征提取层参数量大,严重降低了匹配速度,所以微软团队提出了 CDSSM(convolutional deep semantic) 模型[17] ,利用卷积神经网络(CNN) 来代替了全连接层

进行特征提取,CNN 不仅能够捕捉到局部的特征,而且在编码速度上也更快。

以上基于孪生网络结构的模型的特点在于分别对文本编码,然后通过相似度计算来完成分类。基于交互的匹配模型是在编码时或特征提取时融入了另一文本的信息,Chen 等[18]提出的 ESIM(enhanced sequential inference model)模型由输入编码、局部增强表示、推理合成 3 个部分构成,在进行推理前利用注意力机制[19]完成两段文本的对齐,以此得到经过注意力机制后的相似性加权表示。受 ESIM 的启发,本文在匹配前利用注意力机制来获得每一篇文本对自身上下文感知的表示。

除了在编码时的交互外,其余交互大多是在获得文本的编码表示后在匹配过程进行交互,Shao 等[20]提出段落交互的相似案例检索模型,在获得裁判文书的段落表示向量后,构造了段落与段落间的交互矩阵,最后利用注意力机制进行交互的整合,但段落交互矩阵是不可学习的,因此本文利用仿射变换机制来进行交互,使得交互的过程有了可学习的参数。

2 方法

2.1 相似案例匹配任务

相似案例匹配(similarity case matching, SCM)任务注重于计算裁判文书之间的相似性,选用以 A、B、C 文档组成的三元组 $\langle A、B、C\rangle$ 作为输入数据,A、B、C 是关于民间借贷案件的 3 篇裁判文书,A 是查询文书,B 和 C 是 2 篇候选的案例文书,三元组可以被表示为: $a=\{a_1,a_2,\cdots,a_q\}$, $b=\{b_1,b_2,\cdots,b_n\}$, $c=\{c_1,c_2,\cdots,c_m\}$,其中 a_i、b_j、c_k 分别表示 3 篇文档中的词,文档 A、B、C 的长度分别为 q、n、m,本文通过预先定义的函数 $\mathrm{Score}_{(x)}$ 来衡量两篇候选文档 B、C 与 A 之间的相似度得分,当 $\mathrm{Score}_b>\mathrm{Score}_c$ 时认为 B 相较于 C 与 A 更相似,标签记为 0,反之标签记为 1,SCM 的目标就是给定三元组输入,模型要能够预测出与 A 更相似的是 B 还是 C。

2.2 模型结构

本文模型主要包含 4 层: 输入层、分段编码层、仿射交互层、输出层,图 1 展示了本文提出的模型总体结构。输入层包含 2 个子层: 词嵌入层和自注意力机制层,分段编码层是由 5 个卷积神经网络组成的编码器构成,仿射交互层利用仿射变换对文档编码后的表示矩阵进行相似度得分计算,相似度得分被送入输出层计算输出标签的概率值,利用概率值计算得到模型的损失。

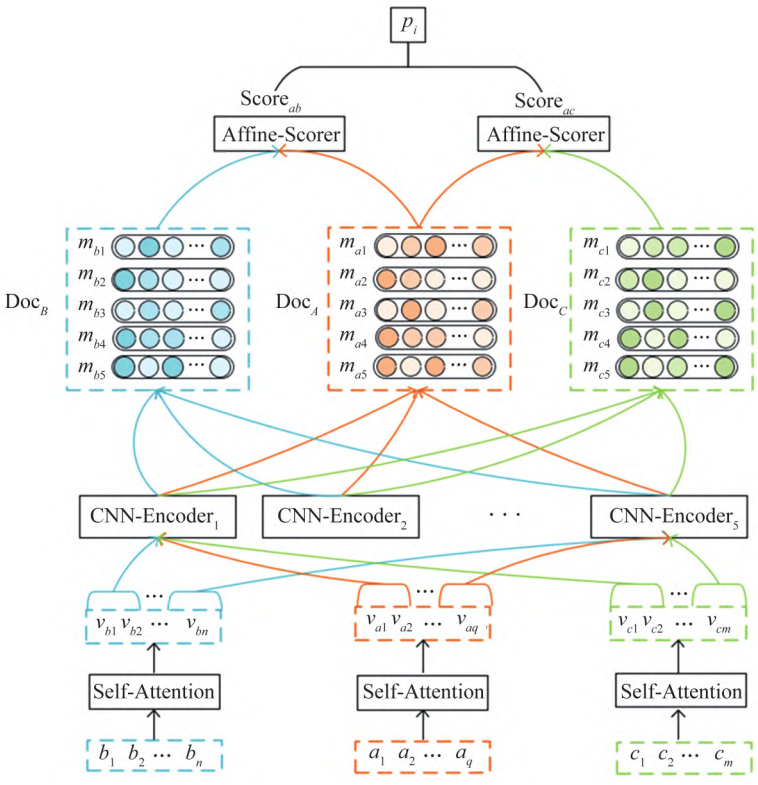

图 1 模型结构图

Fig.1 Model structure

2.2.1　输入层

为了缩减序列的长度,本文使用了词作为文本的最小组成单位,通过对文本的分词处理,这样原本多达 1 062 个字符的文本就能够变成长度为 400 个词的序列,在输入层本文将每一个词映射到一个高维的向量空间,具体地对于长度为 q 的文本 $a=\{a_1,a_2,\cdots,a_q\}$,本文利用预训练的词向量 Word2Vec[6] 得到每一个词 a_i 的固定 d 维的向量表示,整个文档 a 就可以表示为一个矩阵 $\boldsymbol{A}\in\mathbf{R}^{q\times d}$,同样的候选文档 $\boldsymbol{B}\in\mathbf{R}^{n\times d}$和候选文档 $\boldsymbol{C}\in\mathbf{R}^{m\times d}$。

在对文档进行分段编码前,为了尽可能地保证全文的语义信息不因为分段后丢失,本文使用了自注意力机制(self-attention)[21] 对文档中的每一个词进行加权表示,对于文档 a 首先将其词嵌入矩阵 $\boldsymbol{A}$,通过 3 次不同的线性变换转换为 3 个维度同样为 d 的矩阵 $\boldsymbol{Q}$(Query)、$\boldsymbol{K}$(Key)和 $\boldsymbol{V}$(Value),将 3 个矩阵利用公式(1)进行计算得到新的文本矩阵 $\boldsymbol{M}_A$,经过 self-attention 后文档中的每一个词的向量表示都是对文档其余所有词感知的一个向量。

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V}, \tag{1}$$

其中 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\in\mathbf{R}^{q\times d}$,$d$ 表示经过线性变换后词向量的维度,除以$\sqrt{d}$用于避免 $\boldsymbol{Q}$ 与 $\boldsymbol{K}$ 的内积过大,对文档 b、c 本文采用同样方式输入。

2.2.2　分段编码层

分段编码层的输入是来自输入层的文本矩阵 $\boldsymbol{M}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_l\}$,其中 $\boldsymbol{v}_i$ 表示文档中的第 i 个词的向量表示,3 篇文书的输入矩阵具体为 $\boldsymbol{M}_A=\{\boldsymbol{v}_{a1},\boldsymbol{v}_{a2},\cdots,\boldsymbol{v}_{aq}\}$,$\boldsymbol{M}_B=\{\boldsymbol{v}_{b1},\boldsymbol{v}_{b2},\cdots,\boldsymbol{v}_{bn}\}$,$\boldsymbol{M}_C=\{\boldsymbol{v}_{c1},\boldsymbol{v}_{c2},\cdots,\boldsymbol{v}_{cm}\}$,给定文本的输入矩阵 $\boldsymbol{M}$,将其分成 5 段,由公式(2)计算每一段的基本长度 p,再定义一个阈值 mar 用来控制段与段之间重合的词个数,保证段与段之间具有一定的交互信息,根据 p 和 mar 从原始矩阵 $\boldsymbol{M}$ 上可截取到不同大小的 5 个矩阵 $\boldsymbol{M}_i$,$i=1\sim5$。

$$p=\text{len}(\boldsymbol{A})/5,\text{mar}=5。 \tag{2}$$

对于矩阵 $\boldsymbol{M}_i$,分别将其通过对应编码器编码成固定的向量 $\boldsymbol{m}_i$,如公式(3)—(7):

$$\boldsymbol{m}_1=\text{Encoder}_1(\boldsymbol{M}[\boldsymbol{v}_1:\boldsymbol{v}_p]), \tag{3}$$

$$\boldsymbol{m}_2=\text{Encoder}_2(\boldsymbol{M}[\boldsymbol{v}_{p-\text{mar}}:\boldsymbol{v}_{2p+\text{mar}}]), \tag{4}$$

$$\boldsymbol{m}_3=\text{Encoder}_3(\boldsymbol{M}[\boldsymbol{v}_{2p-\text{mar}}:\boldsymbol{v}_{3p+\text{mar}}]), \tag{5}$$

$$\boldsymbol{m}_4=\text{Encoder}_4(\boldsymbol{M}[\boldsymbol{v}_{3p-\text{mar}}:\boldsymbol{v}_{4p+\text{mar}}]), \tag{6}$$

$$\boldsymbol{m}_5=\text{Encoder}_5(\boldsymbol{M}[\boldsymbol{v}_{4p-\text{mar}}:\boldsymbol{v}_l])。 \tag{7}$$

本文使用 CNN 作为编码器,对于矩阵$\boldsymbol{M}_i\in\boldsymbol{R}^{l'\times d}$,$l'$为矩阵 $\boldsymbol{M}_i$ 中词的个数,本文各使用 t 个宽度分别为 h_1、h_2、h_3、h_4,长度与词向量维度一样为 d 的卷积核,以步长 1 对矩阵进行卷积操作,对于某一个宽度 h 的卷积核而言卷积操作具体如公式(8)所示:

$$\boldsymbol{c}_i=f(\boldsymbol{\omega}\cdot\boldsymbol{x}_{i:i+h-1}+b),(i=1,2,\cdots,l'-h+1), \tag{8}$$

其中 $\boldsymbol{x}_{i:i+h-1}$表示矩阵 $\boldsymbol{M}_i$ 第 i 个词到第 $i+h-1$ 个词向量所组成的一个大小为 $h\times d$ 的窗口,$\boldsymbol{\omega}$ 为一个大小为 $h\times d$ 的权重矩阵,$\boldsymbol{b}$ 是一个偏置,f 是一个非线性函数,卷积核通过自上而下移动对矩阵进行卷积得到 $l'-h+1$ 个特征 c。在卷积之后将这些特征拼接后得到最后的特征图 $\boldsymbol{C}=[\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_{l'-h+1}]$,再对特征图进行最大池化后得到特征 c',由于使用了 4 种不同宽度卷积核,每种宽度有 t 个,最终得到的编码后的特征向量 $\boldsymbol{m}_i$ 的维度为 $1\times4t$。将编码得到的向量 $\boldsymbol{m}_i$ 进行拼接后得到一个编码后的矩阵 $\boldsymbol{D}\in\boldsymbol{R}^{5\times4t}$。

2.2.3　仿射交互层

在仿射交互层本文采用了一种基于仿射变换的交互方式计算文本与文本之间的相似得分,本文将其称为仿射打分器(affine-scorer),仿射打分器中在交互时增加了可学习的参数模块,用以缓解现有模型用聚合后的单一向量分类导致的难以学习到细微差异的现象。

对于分段编码层得到的表示矩阵 $\boldsymbol{D}_A$、$\boldsymbol{D}_B$、$\boldsymbol{D}_C$,本文利用公式(9)、(10)计算文本 B 和 C 相较于 A 的相似得分:

$$\text{Score}_{ab}=(\boldsymbol{D}_A\boldsymbol{U}^1)\boldsymbol{D}_B^{\mathrm{T}}+\boldsymbol{D}_A\boldsymbol{U}^2, \tag{9}$$

$$\mathrm{Score}_{ac}=(\boldsymbol{D}_A\boldsymbol{U}^1)\,\boldsymbol{D}_C^{\mathrm{T}}+\boldsymbol{D}_A\boldsymbol{U}^2。\tag{10}$$

其中 $\mathrm{Score}_{ab}\in\boldsymbol{R}^{5\times5}$、$\mathrm{Score}_{ac}\in\boldsymbol{R}^{5\times5}$ 是 2 个得分矩阵,矩阵每一行的数值代表 $\boldsymbol{A}$ 的第 i 段与候选文书的每一段的相似得分。将公式(9)、(10)称为仿射变换是因为相较于传统的仿射分类器 $\boldsymbol{S}_i=\boldsymbol{W}\boldsymbol{x}_i+\boldsymbol{b}$,本文利用了一个变换矩阵 $\boldsymbol{U}^1\in\boldsymbol{R}^{k\times k}$,对文书 $\boldsymbol{A}$ 进行线性变换后代替权重矩阵 $\boldsymbol{W}$,其中 k 为分段编码层的输出维度,对于偏置 $\boldsymbol{b}$ 本文也使用了一个变换矩阵 $\boldsymbol{U}^2\in\boldsymbol{R}^{k\times5}$ 来对 $\boldsymbol{A}$ 进行线性变换后代替,在保留了分类器中存在一定可学习参数的同时与查询文书 $\boldsymbol{A}$ 产生了更多的交互。

2.2.4 输出层

输出层主要由计算文本相似总得分和损失 2 个部分构成。

在计算总得分时先将仿射交互层输出的得分矩阵中所有分数相加作为候选文书的最后得分 Score_b、Score_c。对于打分器输出的分数矩阵 Score_{ab}、Score_{ac},由于本文只关注候选文书与查询文档在每一维度上相似的部分,因此对于得分为负的部分(不相似的部分)通过 ReLU 函数将其置为 0,这样最后的分数之和所代表的就是每一个维度相似得分的总和,具体如公式(11)、(12)所示:

$$\mathrm{Score}_b=\sum_i^5\sum_j^5\mathrm{ReLU}(\mathrm{Score}_{ab}),\tag{11}$$

$$\mathrm{Score}_c=\sum_i^5\sum_j^5\mathrm{ReLU}(\mathrm{Score}_{ac})。\tag{12}$$

对于损失的计算,在计算得到总分数后,通过公式(13)计算概率 p_i,$p_i\in(0,1)$ 作为预测结果输出,其中 $\mathrm{Score}_b>0$、$\mathrm{Score}_c>0$,候选文档 B 相似得分越高 p_i 越大,相反候选文档 C 得分越高 p_i 越小。整个过程采用端到端的方式来完成训练,本文使用交叉熵来作为损失函数,如公式(14)所示:

$$p_i=\frac{\mathrm{Score}_b}{\mathrm{Score}_b+\mathrm{Score}_c},\tag{13}$$

$$L=-\sum_{i=0}^{1}[y_i\log p_i+(1-y_i)\log(1-p_i)],\ y_i\in\{0,1\}。\tag{14}$$

3 实验结果和分析

3.1 数据

CAIL2019-SCM 数据集是由 8 138 个三元组裁判文书构成,三元组包含查询文书 A 和 2 篇候选文书 B、C,一些专业的法律从业者根据裁判文书中预定义的要素对数据集完成了标注,标签 0 代表 2 篇候选文书中 B 与 A 更相似,标签 1 则代表 C 与 A 更相似,所有裁判文书均来自于中国裁判文书网,且全部是关于民间借贷的案件,最终数据的统计情况如表 1 所示。

表 1 数据集统计

Table 1 Dataset statistics

Data set	Label B(0)	Label C(1)	Total
Train	2 596	2 506	5 102
Dev	837	663	1 500
Test	803	733	1 536

官方已将数据集划分为了训练集(train)、验证集(Dev)、测试集(test),分别用于模型的训练、验证以及模型泛化能力的测试,从表 1 中可以看到数据的标签基本是平衡的。

3.2 评价指标和基线模型

本文选用准确率作为评价指标,通过模型计算文档 B 和文档 C 的得分,如果 $\mathrm{Score}_b>\mathrm{Score}_c$ 就将预测结果标签记为 0,认为案例 B 与案例 A 更相似;反之为 1 表示案例 C 与案例 A 更相似。通过与测试数据的真实标签作对比,计算预测结果标签与真实标签一致的个数占比。

在基线模型的选择上,除了官方提供的 3 个基准模型,本文还使用了 2 个较为具有代表性的 ESIM [18] 和 ERNIE-DOC [23] 模型作为基准模型。ESIM 模型在短文本匹配上表现优越,原始的 ESIM 模型是以文本对的形式作为输入,为了适应 CAIL2019-SCM 数据集的三元组结构,本文分别将 A 和 B、A 和 C 以文本对的形式输入,然后计算得到 2 个相似度值,最终通过 Triplet Loss 损失函数进行优化;ERNIE-DOC 是专门为长文本设计的一个文档级预训练模型,结合裁判文书篇幅较长的特点,更适合于 SCM 任务,在利用 ERNIE-DOC 编

码后对编码序列进行池化操作得到特征向量，最终分别计算 A 和 B、A 和 C 的特征向量相似度后，利用 Softmax 函数激活。

3.3 实验设置

本文使用结巴分词对语料进行分词，在分词前通过正则方式现将所有的姓名、地名、公司单位名称提取了出来，然后将这些词传入结巴用户词典以提升分词效果，预训练词向量使用的是 Word2Vec[6] 300 维，整个模型实现基于 Pytorch，详细参数如表 2 所示，批次大小设置为 16，初始学习率为 0.0001，优化器使用 AdamW。

3.4 对比实验

表 3 显示了部分模型在 CAIL2019-SCM 数据集上的实验结果，其中除了前 3 个官方提供的基准模型和 2 个自选基准模型以外，本文还和 CAIL2019-SCM 任务的榜首模型以及目前在该数据集上的 SOTA 模型 LFESM[23] 做了对比。

表 2　参数设置
Table 2　Parameter settings

参数名称	参数值
分段编码层卷积核个数 f	200
卷积核宽度 h	2,3,4,5
分段编码层 mar	5
卷积步长	1
分段编码层 Dropout	0.5
序列最大长度	399

表 3　CAIL2019-SCM 数据集不同方法准确率对比
Table 3　Accuracy comparison of different methods on CAIL2019-SCM　%

模型来源	模型	Dev	test
Baseline	BERT	61.93	67.32
Baseline	LSTM	62.00	68.00
Baseline	CNN	62.27	69.53
Our Baseline	ESIM	63.33	67.84
Our Baseline	ERNIE-DOC	65.60	68.8
Best Score	AlphaCourt	70.70	72.66
State-of-the-Art	LFESM	70.01	74.15
本文模型	Our Model	65.48	76.04

从表 3 中可以看出，在验证集上的表现，本文的模型要好于 3 个官方提供的基准模型和 ESIM，说明了本文算法在相似案例匹配任务上是有效的。在测试集上的准确率上，本文模型比最好的基准模型高了6.51%，比 ERNIE-DOC 提高了 7.24%，其原因在于: 一是这些基于孪生网络设计的基准模型都是通过将文本序列编码成某一固定的向量，然后通过这一特征向量来实现匹配，这种方式会导致原本就很相似的民间借贷裁判文书在固定的特征向量上体现不出差异性，导致模型难以学习到文本之间细微的差异; 二是以字符或词为单位的编码方式使得文本上、下文语义信息丢失，而 ERNIE-DOC 对整篇文本编码又使得裁判文书中丰富的语义信息丢失。虽然与 AlphaCourt 的模型和 LFESM 相比在验证集上表现不佳，但是本文的模型在测试集上的准确率分别提升了 3.38%和 1.89%，这或许是因为 AlphaCourt 的模型和 LFESM 融合了能够体现案件之间差异的一些案件要素特征，导致了模型的泛化能力受到了限制，从表 3 中的结果可以看出，它们从验证集到测试集的提升甚至低于部分基准模型。

通过实验结果的分析可以看出，本文通过将文本分段编码，在保留了一定词与词之间上、下文联系性的同时，又提取到了裁判文书中一定的语义信息，最终通过仿射打分器避免了出现基于某一固定的特征向量完成匹配而效果不佳的问题，整个匹配过程因为完全依赖于模型自动提取特征，使得模型具备了不错的泛化能力，在 CAIL2019-SCM 数据集上取得了不错的效果。

3.5 消融实验

为了进一步探索每一模块对实验效果的影响，本文设置了 4 组实验来观察，实验结果如表 4 所示。

将输入层的自注意力机制(self-attention) 移除后直接将嵌入的文本送入分段编码层，通过分段编码和仿射打分器完成匹配，从结果可以看出在移除自注意力机制后本文的模型在测试集上的准确率下降了 2.35%; 保留了自注意力机制，但是在编码时没有进行分段编码，而是直接利用 CNN 对每一个词

表 4　消融实验准确率对比
Table 4　Accuracy comparison of ablation experiments　%

Method	Dev	Test
本文模型	65.48	76.04
本文模型-Self-Attention	64.78	73.69
本文模型-Encoder(1~5)	61.89	69.79
本文模型-Affine-Scorer	63.57	71.87
本文模型-Encoder[1]	64.51	70.57

注: [1] 分段编码时候共享编码器

编码后通过仿射打分器打分完成匹配,可以看出性能有了明显的下降,说明分段编码在裁判文书匹配上的有效性;保留了输入层和分段编码层的模块,但是不再使用仿射打分器(affine-scorer)打分,而是将分段编码层的输出经过池化后拼接,通过多层感知机预测匹配结果,可以看到在没有使用仿射打分器情况下,测试集表现下降了 4.17%;最后一行不移除任何模块,但是在分段编码时没有再单独为每一段设置编码器,而是共用一个编码器,可以看出准确率下降了 5.47%,说明本文模型在分段编码层编码部分为每一段输入设计单独的编码器,从不同的文本中提取到了不同的特征,这种多角度编码的方式在 CAIL2019-SCM 上是有效的。

3.6 超参数分析

为了探索分段编码层超参数分段数目对实验结果的影响,本文设置了一组实验来验证,图 2 展示了在分段编码层,将文本分为 3、4、5、6 段的实验结果。

如图 2 所示,可以看出将文本分为 5 段时候取得的效果是最好的,本文认为这是与裁判文书的结构有关的,第一段可以把它看作对应裁判文书第一部分诉讼人员的情况,第二段看作对应第二部分诉讼请求,由于第三部分事实描述的长度是最长的,因此将这一部分拆分成 3 段。这种拆分方式与人类在进行长文本理解时的直觉是一致的,实验结果表明这种分段参数设置是合理的。

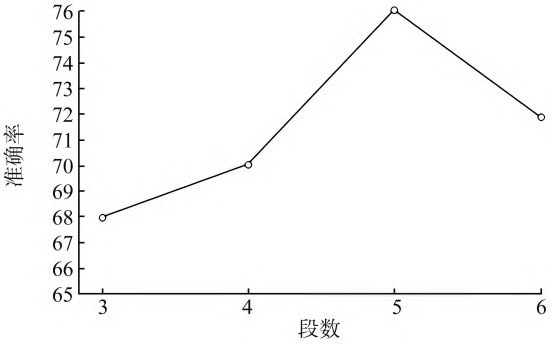

图 2 分段数目对性能的影响

Fig.2 The effect of the number of segments on performance

4 结论

针对 CAIL2019 相似案件匹配任务,本文提出了一种融合分段编码和仿射机制的匹配方法。CAIL2019-SCM 数据集有以下几个挑战:(1) 文本长度过长,是一种典型的长文本;(2) 案件本身情节复杂、要素多,语义信息丰富;(3) 裁判文书用词和格式规范,彼此之间的差异性较小。本文结合裁判文书的构成形式,将文本按一定的长度分成 5 段,基于卷积神经网络为这五段设计了 5 个编码器来捕捉长文本中丰富的语义信息,有效缓解了长文本编码数据稀疏导致难以捕捉有用匹配信息的问题,并基于仿射变换设计了一个打分器来获得衡量相似度的得分,使文本在交互匹配过程中有效学习到了二者的差异性,实验表明本文的模型在该数据集上取得了不错的效果。

参考文献:

[1] HALL P A V, DOWLING G R. Approximate string matching [J]. ACM Computing Surveys (CSUR), 1980, 12(4): 381-402.

[2] SALTON G, BUCKLEY C. Term-weighting approaches in automatic text retrieval [J]. Information Processing & Management, 1988, 24(5): 513-523.

[3] HUANG C H, YIN J, HOU F. A text similarity measurement combining word semantic information with TF-IDF method [J]. Chinese Journal of Computers, 2011, 34(5): 856-864.

[4] NIRAULA N, BANJADE R, ŞTEFĂNESCU D, et al. Experiments with semantic similarity measures based on LDA and LSA [C]//Proceedings of the First International Conference on Statistical Language and Speech Processing. Berlin: Springer, 2013: 188-199.

[5] WANG Z Z, HE M, DU Y P. Text similarity computing based on topic model LDA [J]. Computer Science, 2013, 40(12): 229-232.

[6] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [EB/OL]. (2013-09-07) [2021-07-01]. https://arxiv.org/abs/1301.3781.pdf.

[7] LE Q, MIKOLOV T. Distributed representations of sentences and documents [C]//Proceedings of the 31st International Conference on Machine Learning. Beijing: JMLR, 2014: 1188-1196.

[8] MUELLER J, THYAGARAJAN A. Siamese recurrent architectures for learning sentence similarity [C]//Proceedings of the AAAI Conference on Artificial Intelligence. Arizona: AAAI Press, 2016, 30(1): 2786-2792.

[9] REIMERS N, GUREVYCH I. Sentence-BERT: sentence embeddings using siamese BERT-networks [C] //Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: Association for Computational Linguistics, 2019(1): 3980-3990.

[10] DEVLIN Jacob, CHANG Ming-wei, LEE Kenton, et al. BERT: pre-training of deep bidirectional transformers for language understanding. [C] //Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Long and Short Papers. Minneapolis: Association for Computational Linguistics, 2018: 4171-4186.

[11] WANG Z, HAMZA W, FLORIAN R. Bilateral multi-perspective matching for natural language sentences [C] //Proceedings of Twenty-sixth International Joint Conference on Artificial Intelligence. Melbourne: IJCAI, 2017: 4144-4150.

[12] CHEN Z, ZHANG H, ZHANG X, et al. Quora question pairs [EB/OL]. (2018-05-25) [2021-07-01]. http://static.hongbozhang.me/doc/STAT_441_Report.pdf.

[13] YANG Y, YIH W, MEEK C. Wikiqa: a challenge dataset for open-domain question answering [C] //Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: The Association for Computational Linguistics, 2015: 2013-2018.

[14] XIAO C, ZHONG H, GUO Z. CAIL2019-SCM: a dataset of similar case matching in legal domain [EB/OL]. (2019-09-25) [2021-07-01]. https://arxiv.org/abs/1911.08962.pdf.

[15] HUANG P S, HE X, GAO J, et al. Learning deep structured semantic models for web search using clickthrough data [C] //Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. San Francisco: ACM, 2013: 2333-2338.

[16] CHOPRA S, HADSELL R, LECUN Y. Learning a similarity metric discriminatively, with application to face verification [C] //Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). San Diego: IEEE, 2005: 539-546.

[17] SHEN Y, HE X, GAO J, et al. A latent semantic model with convolutional-pooling structured for information retrieval [C] //Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. Shanghai: CIKM, 2014: 101-110.

[18] CHEN Q, ZHU X, LING Z, et al. Enhanced LSTM for natural language inference [C] //Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017: 1657-1668.

[19] ROCKTÄSCHEL T, GREFENSTETTE E, HERMANN K M, et al. Reasoning about entailment with neural attention [C] //Proceedings of 2016 International Conference on Learning Representations. San Juan: ICLR, 2016.

[20] SHAO Y, MAO J, LIU Y, et al. BERT-PLI: modeling paragraph-level interactions for legal case retrieval [C] //Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. Yokohama: The Association for Computational Linguistics, 2020: 3501-3507.

[21] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: NIPS, 2017: 5998-6008.

[22] DING S, SHANG J, WANG S, et al. ERNIE-DOC: the retrospective long-document modeling transformer [J/OL]. arXiv, 2020, https://arxiv.org/abs/2012.15688.pdf.

[23] HONG Z, ZHOU Q, ZHANG R, et al. Legal feature enhanced semantic matching network for similar case matching [C] //Proceedings of 2020 International Joint Conference on Neural Networks (IJCNN). Glasgow: IEEE, 2020: 1-8.

(编辑: 于善清)