(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111611809 A
(43)申请公布日 2020.09.01

(21)申请号 202010455892.X

(22)申请日 2020.05.26

(71)申请人 西藏大学
地址 850000 西藏自治区拉萨市城关区藏大东路10号

(72)发明人 叶家豪 兰萍 杨丹 李文勇 吴志强

(74)专利代理机构 广州粤高专利商标代理有限公司 44102
代理人 张金福

(51)Int.Cl.
G06F 40/30(2020.01)
G06F 40/211(2020.01)
G06F 40/216(2020.01)
G06F 40/126(2020.01)
G06F 16/35(2019.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)
G06K 9/62(2006.01)

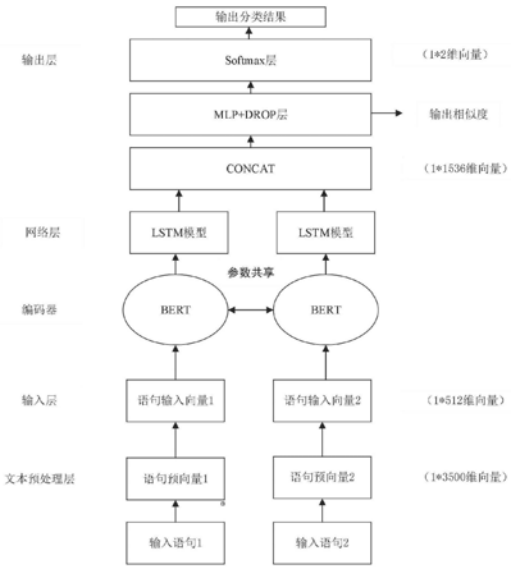
权利要求书3页 说明书12页 附图4页

(54)发明名称

一种基于神经网络的汉语语句相似度计算方法

(57)摘要

本发明提供的一种基于神经网络的汉语语句相似度计算方法,通过构建基于混合语义的编码器,使编码器的收敛速度加快,降低系统对训练语料数量的需求,进而降低系统发生数据过拟合的可能性;再构建语句相似度计算与分类模型实现汉语语句相似度的计算,计算结果准确率高,有效的缓解了现有的基于神经网络的文本相似度方案存在多领域中计算精度不一致的问题。



CN 111611809 A

1.一种基于神经网络的汉语语句相似度计算方法,其特征在于,包括以下步骤:

S1:将待处理的输入语句分别表示为第一语句预向量、第二语句预向量;

S2:获取训练集数据,构建并训练基于混合语义的编码器;

S3:将第一语句预向量、第二语句预向量分别输入两个相同的编码器中,处理得到对应的第一语句编码向量、第二语句编码向量;

S4:构建语句相似度计算与分类模型;

S5:将第一语句编码向量、第二语句编码向量输入语句相似度计算与分类模型中进行拼接、映射和归一化处理,得到类别概率向量;

S6:取类别概率向量中的最大值的类别作为语句的分类结果并取类别概率向量的第二个值作为两个输入语句的相似度,完成汉语语句相似度的计算。

2.根据权利要求1所述的一种基于神经网络的汉语语句相似度计算方法,其特征在于,所述步骤S1具体为:

根据《通用规范汉字表》的一级字表中3500个常用中文字构建语句预向量空间;对输入语句以标点符号作为分割符将语句进行分割,同时为了降低语句的噪音,删除语句中的特殊字符与乱码字符;将3500个汉字以《通用规范汉字表》中顺序编号,输入语句都可以表示为语句预向量空间中的一个向量S;S的表达式为$S=(id_{token1},id_{token2},...,id_{tokeni},...id_{token3500})$,表达式中$id_{tokeni}$表示输入语句是的第i个字的编号,当向量长度大于输入语句长度,超出的部分令id=0;因此得到第一语句预向量、第二语句预向量。

3.根据权利要求2所述的一种基于神经网络的汉语语句相似度计算方法,其特征在于,在所述步骤S2中,所述训练集数据包括中文版的维基百科的汉语条目释义和大型中文问题匹配数据集LCQMC中的训练集和验证集。

4.根据权利要求3所述的一种基于神经网络的汉语语句相似度计算方法,其特征在于,在所述步骤S2中,构建并训练基于混合语义的编码器过程具体为:

构建BERT预训练编码器模型;

使用中文版的维基百科的汉语条目释义对BERT预训练编码器模型进行训练,在BERT预训练编码器模型的基础上进一步构建基于基础语义的预训练编码模型,使预训练编码模型学习文本的基础语义特征,即文本进行编码得到的向量表征中蕴含文本的基础语义;

接着使用LCQMC数据集中的训练集与验证集和LCQMC数据集数据增强后的数据集中的训练集与验证集的数据,对得到的预训练编码模型拼接为下游任务模型后进行训练;目的在于提取文本的上下文信息特征,令文本进行编码得到的向量表征中蕴含文本的不同语句中的具体语义特征,对预训练编码模型进行微调,使下游任务模型更适应下游任务,得到的下游任务模型即为基于混合语义的编码器,即BERT编码器。

5.根据权利要求4所述的一种基于神经网络的汉语语句相似度计算方法,其特征在于,所述步骤S3具体为:

由于BERT编码器的最大输入长度为1*512维的向量,因此截取第一语句预向量或第二语句预向量的前512维向量,即得到输入语句向量S^*,表达为:$S^*=(id_{token1},id_{token2},...id_{token512})$;

由于处理的是文本语句,所以截取前512维的语句预向量不会造成语句的语义丢失;同时,将输入语句向量长度固定为512,输入语句向量在BERT编码器中会自动增添句首与句末

的标志符,输入语句向量会转化成S^*=([CLS],id_{token1},id_{token2},...$id_{token512}$,[SEP]);

接着令BERT编码器输出每一个字的编码,即令下游任务模型的输入转化为512*768维的文本语义向量矩阵,令语句的语义表达更加精细;因此,BERT编码器首先将第一语句预向量、第二语句预向量截取为第一输入语句向量和第二输入语句向量,再将第一输入语句向量和第二输入语句向量的每一个汉字的编码结果顺序,即第一语句编码向量、第二语句编码向量输出至步骤S4构建的语句相似度计算与分类模型中,对超出输入语句长度的编码部分进行补零处理。

6.根据权利要求5所述的一种基于神经网络的汉语语句相似度计算方法,其特征在于,在所述步骤S4中,所述语句相似度计算与分类模型包括两个LSTM模块、一个拼接层,一个全连接MLP层,一个DROPOUT层以及一个SOFTMAX层组成。

7.根据权利要求6所示的一种基于神经网络的汉语语句相似度计算方法,其特征在于,在所述步骤S5中,所述的两个LSTM模块分别对应处理第一语句编码向量、第二语句编码向量;LSTM模块以顺序的方式读取BERT编码器的输出,利用LSTM的记忆网络特性在保留输入文本信息的前提下生成整体语义信息,具体为:

将第一个字的编码结果输入至LSTM模块中作为初始记忆状态C_0;然后LSTM依次读取剩下的字编码作为一个时刻的输入,即输入文本的编码结果在LSTM模块中表示为(C_0,X_1,X_2,...X_t,...X_{511}),其中X_t代表t时刻的输入,每一个X_t首先经过LSTM模块中的遗忘门控单元以确定上一个时刻的记忆状态的重要程度,是否需要遗忘一部分的内容,t时刻遗忘门控单元的计算公式如下:

$$f_t=\mathrm{Sigmoid}(W_f*[h_{t-1},X_t]+b_f) \quad (1)$$

公式(1)中W_f是遗忘门控单元的权重矩阵,h_{t-1}是上一个时刻的输出状态,b_f为遗忘门控单元的偏置系数,σ代表着SIGMOID函数,该函数会的输出是值域为[0,1]的实数,输出越接近1则表示上一个时刻的记忆状态C_{t-1}越重要,保留程度越高,输出为1则C_{t-1}完全保留;输出越接近0则表明上一个时刻的记忆状态C_{t-1}越不重要,遗忘程度越高,输出为0则C_{t-1}完全遗忘;

SIGMOID函数的计算式如下:

$$Sigmoid(x)=\frac{1}{1-e^{-x}} \quad (2)$$

接着X_t经过更新门控单元以确定X_t的重要程度,以确定当前的输入需要更新到C_{t-1}中的程度;更新门控单元计算更新系数i_t与更新记忆状态C_t^*的方式如下:

$$i_t=\mathrm{Sigmoid}(W_i*[h_{t-1},X_t]+b_i) \quad (3)$$

$$C_t^*=tanh\ (W_C*[h_{t-1},X_t]+b_C) \quad (4)$$

$$tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}} \quad (5)$$

公式(3)中W_i是更新门控单元的权重矩阵,b_i是更新门控单元的偏执系数;σ代表着SIGMOID函数,该函数输出的数值表示当前时刻的输入X_t的重要程度,越接近1则表示X_t越重要,则需要更新到当前时刻的记忆单元C_t的程度越高,反之则需要更新到C_t的程度越低;公式(4)中W_C是计算更新记忆状态的权重矩阵,b_C是计算更新记忆状态的偏执系数;tanh层会

生成一个1*768维的向量；

基于计算遗忘门控单元和更新门控单元的计算结果，联合计算出当前时刻的记忆状态C_t，计算式如下：

$$C_t = f_t * C_{t-1} + i_t * C_t^* \quad (6)$$

最后X_t经过输出门控单元，以及根据当前时刻的记忆状态C_t计算出当前时刻的输出状态h_t，计算的公式如下：

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \quad (7)$$

$$h_t = o_t * \tanh(C_t) \quad (8)$$

公式(7)中W_o是输出门控单元的权重矩阵，b_o是输出门控单元的偏执系数，o_t是输出权重系数；

当两个输入文本的编码结果经过LSTM模块的层层更新生成输入文本的语义表达后，将两者的语义表达在拼接层进行拼接，形成一个1*1536维的拼接向量作为MLP层与DROPOUT层的输入；

由全连接MLP层和DROPOUT层对拼接向量进行处理，输出相似度；

最后由SOFTMAX层对得到的相似度进行归一化处理，得到类别概率向量。

8.根据权利要求7所示的一种基于神经网络的汉语语句相似度计算方法，其特征在于，在所述步骤S5中，所述DROPOUT层中的DROPOUT率为0.1。

9.根据权利要求7所述的一种基于神经网络的汉语语句相似度计算方法，其特征在于，在所述步骤S5中，所述拼接向量在MLP层中处理过程如下：

拼接向量首先在MLP层的权重矩阵和偏置的处理下，维度数下降至1*768，完成输入层至隐藏层的转移；

隐藏层向量经过MLP层的激活函数，即SIGMOID的处理维度下降至1*2，完成隐藏层至输出层的转移。

10.根据权利要求9所述的一种基于神经网络的汉语语句相似度计算方法，其特征在于，在所述步骤S5中，全连接MLP层和DROPOUT层的输出经过SOFTMAX函数的处理，得到类别概率向量;SOFTMAX函数的计算式如下：

$$P(S_i) = \frac{e^{g_i}}{\sum_k^n e^{g_k}} \quad (9)$$

其中，i表示文本分类的类别，g_i表示文本类别的值。

一种基于神经网络的汉语语句相似度计算方法

技术领域

[0001] 本发明涉及面向自然语言处理技术领域,更具体的,涉及一种基于神经网络的汉语语句相似度计算方法。

背景技术

[0002] 目前,计算单词或者语句的文本相似度方法可以分为四类[1]Y.Li,D.McLean,Z.A.Bandar,J.D.O'Shea and K.Crockett,"Sentence similarity based on semantic nets and corpus statistics,"in IEEE Transactions on Knowledge and Data Engineering,vol.18,no.8,pp.1138-1150,Aug.2006.doi:10.1109/TKDE.2006.130,分别是基于单词共现计算文本相似度的方法、基于知识库数据计算文本相似度的方法、基于网络引擎计算文本相似度的方法和基于神经网络计算文本相似度的方法。

[0003] 基于单词共现计算相似度的方法[2]C.T.Meadow,B.R.Boyce,and D.H.Kraft,Text InformationRetrieval Systems,seconded.Academic Press,2000是将查询文本作为集合,集合中的元素为字符或者单词,以集合元素的重合程度量化查询文本间的相似度,该类方法计算简洁,但由于无法计算查询文本间的语义相似度,导致该类方法的计算精度低。

[0004] 基于知识库的计算相似度的方法[3]P.W.Foltz,W.Kintsch,and T.K.Landauer,"The Measurement ofTextual Coherence with Latent Semantic Analysis,"Discourse Processes,vol.25,nos.2-3,pp.285-307,1998将单词放于树状知识库中,利用单词子概念之间的最短路径长度,公共节点深度,概念密度等特征量化单词的相似度。知识库的规模、布局及更新速度直接影响该类方法的性能,由于本体知识库的搭建和更新需要语言领域专家的参与,所以知识库存在更新速度慢的缺点,导致该类方法的性能随着时间的推移而逐渐下降。同时,该类方法是基于文本的基础语义计算文本相似度,因此,相同的文本在不同的语句中计算的文本相似度相同,无法基于文本的具体语义计算文本的语义相关性,导致该类方法的计算精度在整体上不足以满足精细的NLP任务的要求。

[0005] 基于网络引擎的计算相似度的方法[4]Cilibrasi R L,Vitanyi P M B.The Google Similarity Distance[J].IEEE Transactions on Knowledge and Data Engineering主要分为基于文本搜索共现页面和基于文本搜索共现窗口两种,不同的搜索引擎会导致不同的单词相似度结果,而且即使查询文本出现在一个页面中共同出现,也无法确定两个文本之间的相干关系,所以这种方法的计算精度难以满足用户的需求。

[0006] 基于神经网络计算相似度的方法利用文本的上下文特征计算文本在具体语句中的具体语义相关性,该类方法计算精度高,但是神经网络模型存在在多领域中计算精度不一致的问题,且当查询文本为字或单词时,由于查询文本缺少上下文信息特征,神经网络模型无法判断文本的具体语义,该类方法性能下降。

[0007] 综上所述,基于知识库的方法和基于神经网络的方法分别具有多领域计算精度一致和计算精度高的优点。但由于基于知识库的方案计算精度低和本体知识库更新速度慢的

缺陷。目前，基于神经网络的文本相似度方案仍存在多领域中计算精度不一致的问题。

发明内容

[0008] 本发明为克服现有的基于神经网络的文本相似度方案存在多领域中计算精度不一致的技术缺陷，提供一种基于神经网络的汉语语句相似度计算方法。

[0009] 为解决上述技术问题，本发明的技术方案如下：

[0010] 一种基于神经网络的汉语语句相似度计算方法，包括以下步骤：

[0011] S1：将待处理的输入语句分别表示为第一语句预向量、第二语句预向量；

[0012] S2：获取训练集数据，构建并训练基于混合语义的编码器；

[0013] S3：将第一语句预向量、第二语句预向量分别输入两个相同的编码器中，处理得到对应的第一语句编码向量、第二语句编码向量；

[0014] S4：构建语句相似度计算与分类模型；

[0015] S5：将第一语句编码向量、第二语句编码向量输入语句相似度计算与分类模型中进行拼接、映射和归一化处理，得到类别概率向量；

[0016] S6：取类别概率向量中的最大值的类别作为语句的分类结果并取类别概率向量的第二个值作为两个输入语句的相似度，完成汉语语句相似度的计算。

[0017] 其中，所述步骤S1具体为：

[0018] 根据《通用规范汉字表》的一级字表中3500个常用中文字构建语句预向量空间；对输入语句以标点符号作为分割符将语句进行分割，同时为了降低语句的噪音，删除语句中的特殊字符与乱码字符；将3500个汉字以《通用规范汉字表》中顺序编号，输入语句都可以表示为语句预向量空间中的一个向量S；S的表达式为$S=(id_{token1},id_{token2},...,id_{tokeni},...id_{token3500})$，表达式中$id_{tokeni}$表示输入语句是的第i个字的编号，当向量长度大于输入语句长度，超出的部分令id＝0；因此得到第一语句预向量、第二语句预向量。

[0019] 其中，在所述步骤S2中，所述训练集数据包括中文版的维基百科的汉语条目释义和大型中文问题匹配数据集LCQMC中的训练集和验证集。

[0020] 其中，在所述步骤S2中，构建并训练基于混合语义的编码器过程具体为：

[0021] 构建BERT预训练编码器模型；

[0022] 使用中文版的维基百科的汉语条目释义对BERT预训练编码器模型进行训练，在BERT预训练编码器模型的基础上进一步构建基于基础语义的预训练编码模型，使预训练编码模型学习文本的基础语义特征，即文本进行编码得到的向量表征中蕴含文本的基础语义；

[0023] 接着使用LCQMC数据集中的训练集与验证集和LCQMC数据集数据增强后的数据集中的训练集与验证集的数据，对得到的预训练编码模型拼接为下游任务模型后进行训练；目的在于提取文本的上下文信息特征，令文本进行编码得到的向量表征中蕴含文本的不同语句中的具体语义特征，对预训练编码模型进行微调，使下游任务模型更适应下游任务，得到的下游任务模型即为基于混合语义的编码器，即BERT编码器。

[0024] 其中，所述步骤S3具体为：

[0025] 由于BERT编码器的最大输入长度为1*512维的向量，因此截取第一语句预向量或第二语句预向量的前512维向量，即得到输入语句向量S^*，表达为：$S^*=(id_{token1},id_{token2},$

...$id_{token512}$)；

[0026] 由于处理的是文本语句,所以截取前512维的语句预向量不会造成语句的语义丢失;同时,将输入语句向量长度固定为512,输入语句向量在BERT编码器中会自动增添句首与句末的标志符,输入语句向量会转化成S^*=([CLS],id_{token1},id_{token2},...$id_{token512}$,[SEP])；

[0027] 接着令BERT编码器输出每一个字的编码,即令下游任务模型的输入转化为512*768维的文本语义向量矩阵,令语句的语义表达更加精细;因此,BERT编码器首先将第一语句预向量、第二语句预向量截取为第一输入语句向量和第二输入语句向量,再将第一输入语句向量和第二输入语句向量的每一个汉字的编码结果顺序,即第一语句编码向量、第二语句编码向量输出至步骤S4构建的语句相似度计算与分类模型中,对超出输入语句长度的编码部分进行补零处理。

[0028] 其中,在所述步骤S4中,所述语句相似度计算与分类模型包括两个LSTM模块、一个拼接层,一个全连接MLP层,一个DROPOUT层以及一个SOFTMAX层组成。

[0029] 其中,在所述步骤S5中,所述的两个LSTM模块分别对应处理第一语句编码向量、第二语句编码向量;LSTM模块以顺序的方式读取BERT编码器的输出,利用LSTM的记忆网络特性在保留输入文本信息的前提下生成整体语义信息,具体为：

[0030] 将第一个字的编码结果输入至LSTM模块中作为初始记忆状态C_0;然后LSTM依次读取剩下的字编码作为一个时刻的输入,即输入文本的编码结果在LSTM模块中表示为($C0,X_1$,X_2,...X_t,...X_{511}),其中X_t代表t时刻的输入,每一个X_t首先经过LSTM模块中的遗忘门控单元以确定上一个时刻的记忆状态的重要程度,是否需要遗忘一部分的内容,t时刻遗忘门控单元的计算公式如下：

[0031] f_t=Sigmoid(W_f*[h_{t-1},X_t]+b_f) (1)

[0032] 公式(1)中W_f是遗忘门控单元的权重矩阵,h_{t-1}是上一个时刻的输出状态,b_f为遗忘门控单元的偏置系数,σ代表着SIGMOID函数,该函数会的输出是值域为[0,1]的实数,输出越接近1则表示上一个时刻的记忆状态C_{t-1}越重要,保留程度越高,输出为1则C_{t-1}完全保留;输出越接近0则表明上一个时刻的记忆状态C_{t-1}越不重要,遗忘程度越高,输出为0则C_{t-1}完全遗忘;

[0033] SIGMOID函数的计算式如下：

[0034] $$Sigmoid(x) = \frac{1}{1 - e^{-x}} \quad (2)$$

[0035] 接着X_t经过更新门控单元以确定X_t的重要程度,以确定当前的输入需要更新到C_{t-1}中的程度;更新门控单元计算更新系数i_t与更新记忆状态C_t^*的方式如下：

[0036] i_t=Sigmoid(W_i*[h_{t-1},X_t]+b_i) (3)

[0037] $$C_t^* = tanh\ (W_C * [h_{t-1}, X_t] + b_C) \quad (4)$$

[0038] $$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5)$$

[0039] 公式(3)中W_i是更新门控单元的权重矩阵,b_i是更新门控单元的偏执系数;σ代表着SIGMOID函数,该函数输出的数值表示当前时刻的输入X_t的重要程度,越接近1则表示X_t越重要,则需要更新到当前时刻的记忆单元C_t的程度越高,反之则需要更新到C_t的程度越低;公

式(4)中W_C是计算更新记忆状态的权重矩阵，b_C是计算更新记忆状态的偏执系数；tanh层会生成一个1*768维的向量；

[0040] 基于计算遗忘门控单元和更新门控单元的计算结果，联合计算出当前时刻的记忆状态C_t，计算式如下：

[0041] $$C_t = f_t * C_{t-1} + i_t * C_t^* \quad (6)$$

[0042] 最后X_t经过输出门控单元，以及根据当前时刻的记忆状态C_t计算出当前时刻的输出状态h_t，计算的公式如下：

[0043] $o_t = \sigma(W_o[h_{t-1}, X_t] + b_o)$ (7)

[0044] $h_t = o_t * \tanh(C_t)$ (8)

[0045] 公式(7)中W_o是输出门控单元的权重矩阵，b_o是输出门控单元的偏执系数，o_t是输出权重系数；

[0046] 当两个输入文本的编码结果经过LSTM模块的层层更新生成输入文本的语义表达后，将两者的语义表达在拼接层进行拼接，形成一个1*1536维的拼接向量作为MLP层与DROPOUT层的输入；

[0047] 由全连接MLP层和DROPOUT层对拼接向量进行处理，输出相似度；

[0048] 最后由SOFTMAX层对得到的相似度进行归一化处理，得到类别概率向量。

[0049] 其中，在所述步骤S5中，所述DROPOUT层中的DROPOUT率为0.1.

[0050] 其中，在所述步骤S5中，所述拼接向量在MLP层中处理过程如下：

[0051] 拼接向量首先在MLP层的权重矩阵和偏置的处理下，维度数下降至1*768，完成输入层至隐藏层的转移；

[0052] 隐藏层向量经过MLP层的激活函数，即SIGMOID的处理维度下降至1*2，完成隐藏层至输出层的转移。

[0053] 其中，在所述步骤S5中，全连接MLP层和DROPOUT层的输出经过SOFTMAX函数的处理，得到类别概率向量；SOFTMAX函数的计算式如下：

[0054] $$P(S_i) = \frac{e^{g_i}}{\sum_k^n e^{g_k}} \quad (9)$$

[0055] 其中，i表示文本分类的类别，g_i表示文本类别的值。

[0056] 上述方案中，本发明利用中文版的维基百科的条目释义和大型中文问题匹配数据集(A Large-scale Chinese Question Matching Corpus，LCQMC)训练基于转变器的双向编码模型(Bidirectional Encoder Representations from Transformers，BERT)，令编码器学习文本的基础语义信息特征和具体语义特征，令文本的编码具有混合语义信息特征，并结合两个长短期记忆网络(long-Short Term Memory，LSTM)模块，一个拼接层、一个全连接层及DROPOUT层搭建下游的相似度计算与分类神经网络模型，利用文本的长度与位置特征计算文本相似度和类别。

[0057] 与现有技术相比，本发明技术方案的有益效果是：

[0058] 本发明提供的一种基于神经网络的汉语语句相似度计算方法，通过构建基于混合语义的编码器，使编码器的收敛速度加快，降低系统对训练语料数量的需求，进而降低系统发生数据过拟合的可能性；再构建语句相似度计算与分类模型实现汉语语句相似度的计算，计算结果准确率高，有效的缓解了现有的基于神经网络的文本相似度方案存在多领域

中计算精度不一致的问题。

附图说明

[0059] 图1为神经网络模型框架示意图；

[0060] 图2为编码器训练步骤流程图；

[0061] 图3为MLP+DROPOUT层框架示意图；

[0062] 图4为模型在训练集中的性能示意图；

[0063] 图5为模型在训练集中的损失函数示意图；

[0064] 图6为模型在验证集中的性能示意图；

[0065] 图7为模型在验证集中的损失函数示意图。

具体实施方式

[0066] 附图仅用于示例性说明，不能理解为对本专利的限制；

[0067] 为了更好说明本实施例，附图某些部件会有省略、放大或缩小，并不代表实际产品的尺寸；

[0068] 对于本领域技术人员来说，附图中某些公知结构及其说明可能省略是可以理解的。

[0069] 下面结合附图和实施例对本发明的技术方案做进一步的说明。

[0070] 实施例1

[0071] 如图1所示，一种基于神经网络的汉语语句相似度计算方法，包括以下步骤：

[0072] S1：将待处理的输入语句分别表示为第一语句预向量、第二语句预向量；

[0073] S2：获取训练集数据，构建并训练基于混合语义的编码器；

[0074] S3：将第一语句预向量、第二语句预向量分别输入两个相同的编码器中，处理得到对应的第一语句编码向量、第二语句编码向量；

[0075] S4：构建语句相似度计算与分类模型；

[0076] S5：将第一语句编码向量、第二语句编码向量输入语句相似度计算与分类模型中进行拼接、映射和归一化处理，得到类别概率向量；

[0077] S6：取类别概率向量中的最大值的类别作为语句的分类结果并取类别概率向量的第二个值作为两个输入语句的相似度，完成汉语语句相似度的计算。

[0078] 在具体实施过程中，在本发明构建的神经网络系统中，首先将输入语句表示成1*3500维的语句预向量，继而在模型的输入层截取前1*512维的语句预向量作为BERT编码器的输入。在编码器的部分，构建基于BERT的孪生神经网络编码器，使编码器的收敛速度加快，降低系统对训练语料数量的需求，进而降低系统发生数据过拟合的可能性。经过训练好的编码器的处理，输入语句1和输入语句2分别被表征成具有混合语义的512*768维的语句编码1和语句编码向量2。语句编码结果经过LSTM模型后，被映射成1*768维的语句整体信息向量，将两个输入语句的整体信息向量拼接成1*1568维的向量作为全连接层和DROPOUT层的输入，全连接层将输入向量映射成一个1*2维的概率向量，最后将此概率向量输入到SOFTMAX层计算归一化概率，得到归一化的1*2维的类别概率向量，取向量中的最大值的类别作为语句的分类结果并取类别概率向量的第二个值作为两个输入句子中的相似度。

[0079] 更具体的,所述步骤S1具体为:

[0080] 首先,根据国家语言文字工作委员会在2013年发布的《通用规范汉字表》的一级字表中3500个常用中文字构建的语句预向量空间;对输入语句以标点符号作为分割符将语句进行分割,同时为了降低语句的噪音,删除语句中的特殊字符与乱码字符;将3500个汉字以《通用规范汉字表》中顺序编号,输入语句都可以表示为语句预向量空间中的一个向量S;S的表达式为S=(id_{token1},id_{token2},...,id_{tokeni},...$id_{token3500}$),表达式中id_{tokeni}表示输入语句是的第i个字的编号,当向量长度大于输入语句长度,超出的部分令id=0;因此得到第一语句预向量、第二语句预向量。

[0081] 更具体的,在所述步骤S2中,所述训练集数据包括中文版的维基百科的汉语条目释义和大型中文问题匹配数据集LCQMC中的训练集和验证集。

[0082] 在具体实施过程中,中文版的维基百科于2002年8月创立,截止2020年,中文版的维基百科拥有110万篇条目,包含了各个地区的华人语料,本发明以3500个常用中文字为基准,爬虫中文版维基百科中的搜索信息,得到常用字的详细释义、参考词组以及分类。该训练集数据目的是提取字的基础释义特征。

[0083] LCQMC数据集是哈尔滨工业大学在自然语言处理国际顶会COLING2018构建的问题语义匹配数据集,其建立的目标是判断两个问题的语义是否相似。该数据集更注重意图匹配而不是某个具体重点词汇的释义。数据集包含了训练集的238766个问题对,验证集的8802个问题对以及测试集的12500个问题对。

[0084] 更具体的,在所述步骤S2中,由于BERT的模型最少层数为12,决定了BERT需要海量的训练数据才能较好地利用提取的特征表征文本数据。为了避免大型神经网络的数据过拟合问题,本发明采取了构建预训练模型、微调、数据增长与添加DROPOUT层等四种防止过拟合的方式。构建并训练基于混合语义的编码器过程具体为:

[0085] 构建BERT预训练编码器模型;

[0086] 使用中文版的维基百科的汉语条目释义对BERT预训练编码器模型进行训练,在BERT预训练编码器模型的基础上进一步构建基于基础语义的预训练编码模型,使预训练编码模型学习文本的基础语义特征,即文本进行编码得到的向量表征中蕴含文本的基础语义;

[0087] 接着使用LCQMC数据集中的训练集与验证集和LCQMC数据集数据增强后的数据集中的训练集与验证集的数据,对得到的预训练编码模型拼接为下游任务模型后进行训练;目的在于提取文本的上下文信息特征,令文本进行编码得到的向量表征中蕴含文本的不同语句中的具体语义特征,对预训练编码模型进行微调,使下游任务模型更适应下游任务,得到的下游任务模型即为基于混合语义的编码器,即BERT编码器。

[0088] 在具体实施过程中,如图2所示,为了解决基于知识库的相似度计算方式无法表征文本在特定语句的具体语义和基于神经网络的相似度计算方式在不同领域中计算精度不相同的问题,本发明将文本的基础语义和上下文特征相结合,令文本的向量表征中同时包含这两种特征,使文本的向量表征一种混合语义,这种方式的向量表征适用于不同领域的文本。

[0089] 更具体的,所述步骤S3具体为:

[0090] 由于BERT编码器的最大输入长度为1*512维的向量,因此截取第一语句预向量或

第二语句预向量的前512维向量，即得到输入语句向量S^*，表达为：$S^*=(id_{token1},id_{token2},...id_{token512})$；

[0091] 由于处理的是文本语句，所以截取前512维的语句预向量不会造成语句的语义丢失；同时，为了使编码器的输出长度固定，使文本的语义精细地表达，将输入语句向量长度固定为512，输入语句向量在BERT编码器中会自动增添句首与句末的标志符，输入语句向量会转化成$S^*=([CLS],id_{token1},id_{token2},...id_{token512},[SEP])$；

[0092] 一般的BERT模型的输出是一个字的长度的编码，即将文本句首标识符[CLS]的编码输出。这种形式的编码输出只是一种文本整体信息的表达，本发明修改了BERT编码器的输出，令编码器输出每一个字的编码，即令下游任务模型的输入转化为512*768维的文本语义向量矩阵，令语句的语义表达更加精细，而不只是一个语句整体的信息表达；因此，BERT编码器首先将第一语句预向量、第二语句预向量截取为第一输入语句向量和第二输入语句向量，再将第一输入语句向量和第二输入语句向量的每一个汉字的编码结果顺序，即第一语句编码向量、第二语句编码向量输出至步骤S4构建的语句相似度计算与分类模型中，对超出输入语句长度的编码部分进行补零处理。

[0093] 在具体实施过程中，舍弃传统BERT编码器只输出文本的句首标识符[CLS]的编码的优势有：

[0094] 传统BERT编码器对输入文本的编码输出是在下游任务模型的约束下所得到的输入文本整体语义表征，而本发明所构建的BERT输出是输入文本所有字的编码，与传统的输出方式相比，本发明所构建的输出方式可以得到输入语句更加具体的语义表征。

[0095] 在传统的BERT编码器中，无论输入文本的长度是多少，编码器输出的都是一个Token长度的编码，这种输出方式无法表达输入文本的长度特征。而本发明构建的输出方式仅需要计算文本编码中非全零元素的行数，即可获得输入文本的长度，所以这种输出方式包含输入文本的长度特征。

[0096] 在传统的BERT编码器中，编码器输出的一个汉字长度的编码，这种输出方式无法表达输入文本的位置特征，本发明构建的输出方式以顺序的方式输入到下游任务模型中，这种输出方式包含了文本的位置特征。

[0097] 更具体的，在所述步骤S4中，由于BERT编码器的神经网络层数是12，为了避免相似度计算系统整体过于庞大导致数据过拟合问题的发生，下游任务模型的层数设计不宜过多。因此所述语句相似度计算与分类模型包括两个LSTM模块、一个拼接层，一个全连接MLP层，一个DROPOUT层以及一个SOFTMAX层组成。

[0098] 在具体实施过程中，本发明使用的LSTM模块的参数如下：单次训练样本数(Batch Size)为64、时刻输入数(N steps)为512、输入维度(Input dimension)为768、隐藏层的维度(Hidden dimension)为768、输出维度(Output dimension)为768。

[0099] 更具体的，在所述步骤S5中，所述的两个LSTM模块分别对应处理第一语句编码向量、第二语句编码向量；LSTM模块以顺序的方式读取BERT编码器的输出，利用LSTM的记忆网络特性在保留输入文本信息的前提下生成整体语义信息，具体为：

[0100] 将第一个字的编码结果输入至LSTM模块中作为初始记忆状态C_0；然后LSTM依次读取剩下的字编码作为一个时刻的输入，即输入文本的编码结果在LSTM模块中表示为$(C_0,X_1,X_2,...X_t,...X_{511})$，其中$X_t$代表t时刻的输入，每一个$X_t$首先经过LSTM模块中的遗忘门控单

元以确定上一个时刻的记忆状态的重要程度，是否需要遗忘一部分的内容，t时刻遗忘门控单元的计算公式如下：

[0101] $f_t=Sigmoid(W_f*[h_{t-1},X_t]+b_f)$ (1)

[0102] 公式(1)中W_f是遗忘门控单元的权重矩阵，h_{t-1}是上一个时刻的输出状态，b_f为遗忘门控单元的偏置系数，σ代表着SIGMOID函数，该函数会的输出是值域为[0,1]的实数，输出越接近1则表示上一个时刻的记忆状态C_{t-1}越重要，保留程度越高，输出为1则C_{t-1}完全保留；输出越接近0则表明上一个时刻的记忆状态C_{t-1}越不重要，遗忘程度越高，输出为0则C_{t-1}完全遗忘；

[0103] SIGMOID函数的计算式如下：

[0104] $$Sigmoid(x)=\frac{1}{1-e^{-x}} \quad (2)$$

[0105] 接着X_t经过更新门控单元以确定X_t的重要程度，以确定当前的输入需要更新到C_{t-1}中的程度；更新门控单元计算更新系数i_t与更新记忆状态C_t^*的方式如下：

[0106] $i_t=Sigmoid(W_i*[h_{t-1},X_t]+b_i)$ (3)

[0107] $$C_t^*=tanh\ (W_C*[h_{t-1},X_t]+b_C) \quad (4)$$

[0108] $$tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}} \quad (5)$$

[0109] 公式(3)中W_i是更新门控单元的权重矩阵，b_i是更新门控单元的偏执系数；σ代表着SIGMOID函数，该函数输出的数值表示当前时刻的输入X_t的重要程度，越接近1则表示X_t越重要，则需要更新到当前时刻的记忆单元C_t的程度越高，反之则需要更新到C_t的程度越低；公式(4)中W_C是计算更新记忆状态的权重矩阵，b_C是计算更新记忆状态的偏执系数；tanh层会生成一个1*768维的向量；

[0110] 基于计算遗忘门控单元和更新门控单元的计算结果，联合计算出当前时刻的记忆状态C_t，计算式如下：

[0111] $$C_t=f_t*C_{t-1}+i_t*C_t^* \quad (6)$$

[0112] 最后X_t经过输出门控单元，以及根据当前时刻的记忆状态C_t计算出当前时刻的输出状态h_t，计算的公式如下：

[0113] $o_t=\sigma(W_o[h_{t-1},X_t]+b_o)$ (7)

[0114] $h_t=o_t*tanh(C_t)$ (8)

[0115] 公式(7)中W_o是输出门控单元的权重矩阵，b_o是输出门控单元的偏执系数，o_t是输出权重系数；

[0116] 当两个输入文本的编码结果经过LSTM模块的层层更新生成输入文本的语义表达后，将两者的语义表达在拼接层进行拼接，形成一个1*1536维的拼接向量作为MLP层与DROPOUT层的输入；

[0117] 由全连接MLP层和DROPOUT层对拼接向量进行处理，输出相似度；

[0118] 最后由SOFTMAX层对得到的相似度进行归一化处理，得到类别概率向量。

[0119] 更具体的，在所述步骤S5中，所述DROPOUT层中的DROPOUT率为0.1.

[0120] 更具体的，如图3所示，实心圆代表模型训练时进行参加训练的数据文本，空心圆代表模型训练时被DROPOUT层随机舍弃的数据文本，DROPOUT层通过随机舍弃模型中的数据

点,形成随机数据训练模型,这种方法降低模型出现数据过拟合情况的可能性。

[0121] 如图3所示,在所述步骤S5中,所述拼接向量在MLP层中处理过程如下:

[0122] 拼接向量首先在MLP层的权重矩阵和偏置的处理下,维度数下降至1*768,完成输入层至隐藏层的转移;

[0123] 隐藏层向量经过MLP层的激活函数,即SIGMOID的处理维度下降至1*2,完成隐藏层至输出层的转移。

[0124] 更具体的,在所述步骤S5中,全连接MLP层和DROPOUT层的输出经过SOFTMAX函数的处理,得到类别概率向量;SOFTMAX函数的计算式如下:

[0125] $$P(S_i)=\frac{e^{g_i}}{\sum_k^n e^{g_k}} \quad (9)$$

[0126] 其中,i表示文本分类的类别,g_i表示文本类别的值。

[0127] 在具体实施过程中,本发明利用中文版的维基百科的条目释义和大型中文问题匹配数据集(ALarge-scale Chinese Question Matching Corpus,LCQMC)训练基于转变器的双向编码模型(Bidirectional Encoder Representations from Transformers,BERT),令编码器学习文本的基础语义信息特征和具体语义特征,令文本的编码具有混合语义信息特征,并结合两个长短期记忆网络(long-Short Term Memory,LSTM)模块,一个拼接层、一个全连接层及DROPOUT层搭建下游的相似度计算与分类神经网络模型,利用文本的长度与位置特征计算文本相似度和类别。

[0128] 实施例2

[0129] 更具体的,在实施例1的基础上,在LCQMC数据集中对中文领域内基于混合语义的神经网络计算文本相似度方案的性能进行了分析。

[0130] 在具体实施过程中,对于神经网络性能包括五个分析指标:正确率(Accuracy)、召回率(Recall)、精准率(Precision)、F1分数(F1-score)以及模型损失函数(Loss)。

[0131] 表1.预测数据的分类

[0132]

	正例(预测分类结果)	反例(预测分类结果)
正例(真实分类结果)	真正例(TP)	伪反例(FN)
反例(真实分类结果)	伪正例(FP)	真反例(TN)

[0133] 在具体实施过程中,表1是基于样本真实分类结果和预测分类结果对测试样本的分类,本发明利用四个分类样本的数量计算正确率、召回率、精准率和F1分数。

[0134] 正确率为预测分类结果正确的样本数占预测样本总数的比例。正确率的计算公式如下:

[0135] $$Accuracy=\frac{TP+TN}{TP+FN+FP+TN} \quad (10)$$

[0136] 召回率为预测分类结果正确的正例样本数占真实正例样本数的比例,召回率体现模型在研究领域内召回目标类别的能力。召回率的计算式如下:

[0137] $$Recall=\frac{TP}{TP+FN} \quad (11)$$

[0138] 精准率为真实正例样本数占预测分类结果为正例的样本数的比例。精准率体现模

型在研究领域内精准捕获目标类别的能力。精准率的计算式如下：

[0139] $$Precision = \frac{TP}{TP + FP} \quad (12)$$

[0140] F1分数为召回率和召回率的调和平均值，F1分数体现了模型的综合能力。F1分数的计算式如下：

[0141] $$F1 - score = \frac{Recall * Precision * 2}{Recall + Precision} \quad (13)$$

[0142] 损失函数体现模型的在研究领域内的预测结果与真实结果之间偏离程度。由于本发明所搭建的神经网络模型计算文本的分类结果，所以使用交叉熵损失函数作为该方案的损失函数，使用的优化器为“Adam”，优化器的学习率为10^{-5}。交叉熵损失函数的计算式如下：

[0143] $$L = \frac{1}{N}\sum_{0}^{N} L_i = \frac{1}{N}\sum_{0}^{N} -[y_i * log(p_i) + (1 - y_i) * log (1 - p_i)] \quad (14)$$

[0144] 公式(14)中N为预测的样本总数，y_i表示样本i的标签，正例取值为1，反例取值为0。p_i表示样本i预测为正例的概率，即样本i之间的文本相似度。

[0145] 在具体实施过程中，如图4所示，随着神经网络模型训练LCQMC中训练集的训练批次数的上升，模型的正确率、F1分数、准确率以及召回率也迅速提升。当训练次数达到3次时，模型的四项性能指标都提升至90％以上。增加模型的训练批次数，四项性能指标稳步提升，当训练批次数达到42时，基于混合语义计算文本相似度的神经网络模型在训练集中性能达到最高，四项性能指标都在95％以上。在这个基础上继续增加模型的训练批次数，四项性能指标基本稳定不变，即模型在训练集中性能不再提升。

[0146] 如图5所示，随着神经网络模型训练LCQMC中训练集的训练批次数的上升，交叉熵损失函数值迅速下降，即模型预测结果与真实结果的偏离程度迅速减少，当模型的训练次数达到3次时，损失函数值减少至50。继续增加模型的训练批次数，损失函数稳步下降，当训练批次数达到42时，损失系数下降到0。继续增加模型的训练批次数，损失函数值产生波动变化，但波动变化量不大，即模型的交叉熵损失函数基本稳定。

[0147] 图6是经过训练集训练后的模型在验证集文本中的性能表现。在训练初期阶段，随着神经网络模型训练LCQMC中验证集的训练次数的上升，模型的正确率、F1分数以及准确率迅速提升，召回率呈下降趋势。当模型的训练次数达到3次时，四项性能指标集中在84％附近。继续增加模型的训练批次数，模型的正确率、F1分数以及准确率性能上升，召回率在波动中上升。当训练次数达到40时，模型的四项性能数值达到87％。继续增加模型的训练次数，正确率与F1分数基本不变，召回率和准确率呈波动变化，但波动幅度逐渐变小，这是由于LCQMC验证集数据为聚集型数据，标签类别一致的数据没有分散，缺少输入的随机性。

[0148] 图7是经过训练集训练后的模型在验证集文本中的损失函数表现。训练初期阶段，随着神经网络模型训练LCQMC中验证集的训练次数的上升，模型的交叉熵损失函数迅速下降。当模型的训练次数达到37时，模型的损失达到最低值，最低值为38。继续增加训练次数，模型的损失函数在40附近波动上升，上升幅度逐渐减小。由于基于混合语义的神经网络模型在验证集中的四项性能指标低于模型在训练集中的四项性能指标，所以图7中损失函数值比图5中的损失函数值高。

[0149] 经过训练后的方案模型在LCQMC数据集中与其它方案模型的性能对比数据如表2。

[0150] 表2.在LCQMC数据集中方案性能对比

[0151]

方案名称	验证集正确率（%）	测试集正确率（%）	测试集 F1 分数（%）	测试集准确率（%）	测试集召回率（%）
WMD	/	0.706	0.734	0.67	0.812
C_WO	/	0.707	0.706	0.611	0.887
S_COS	/	0.703	0.716	0.601	0.889
CBOW	/	0.737	0.774	0.679	0.899
CNN	/	0.728	0.757	0.684	0.846
BILSTM	/	0.761	0.789	0.706	0.833

[0152]

BIMPM	/	0.834	0.85	0.776	0.939
DSSM	/	0.6334	/	/	/
ABCNN	/	0.7992	/	/	/
ESIM	/	0.818	/	/	/
DIIN	/	0.8447	/	/	/
NEZHA	0.8964	0.8618	/	/	/
CHARTEST	0.8443	0.8618	/	/	/
ERNIE	0.897	0.874	/	/	/
本发明方案	0.8861	0.8872	0.8893	0.8549	0.9258

[0153] 如表2所示，本发明提出的基于混合语义的神经网络模型，在LCQMC的测试集中的正确率、F1分数以及准确率高于以往的方案，较之前方案的最优值分别有0.015、0.0393、0.0789的提升量。而模型在验证集的正确率以及测试集的召回率与之前方案的最优值接近。基于2018蚂蚁金服自然语言处理比赛的规则，分类模型最重要的性能指标分别是测试

集的正确率与F1分数。由此可见，本发明提出的基于混合语义的神经网络模型性能优于其它方案的模型，验证了该方案对提高汉语语句相似度计算精度的有效性。同时，模型在LCQMC的验证集和测试集中正确率几乎一致，证实该模型可有效缓解神经网络模型在多领域内精度不一致的问题。

[0154] 显然，本发明的上述实施例仅仅是为清楚地说明本发明所作的举例，而并非是对本发明的实施方式的限定。对于所属领域的普通技术人员来说，在上述说明的基础上还可以做出其它不同形式的变化或变动。这里无需也无法对所有的实施方式予以穷举。凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明权利要求的保护范围之内。

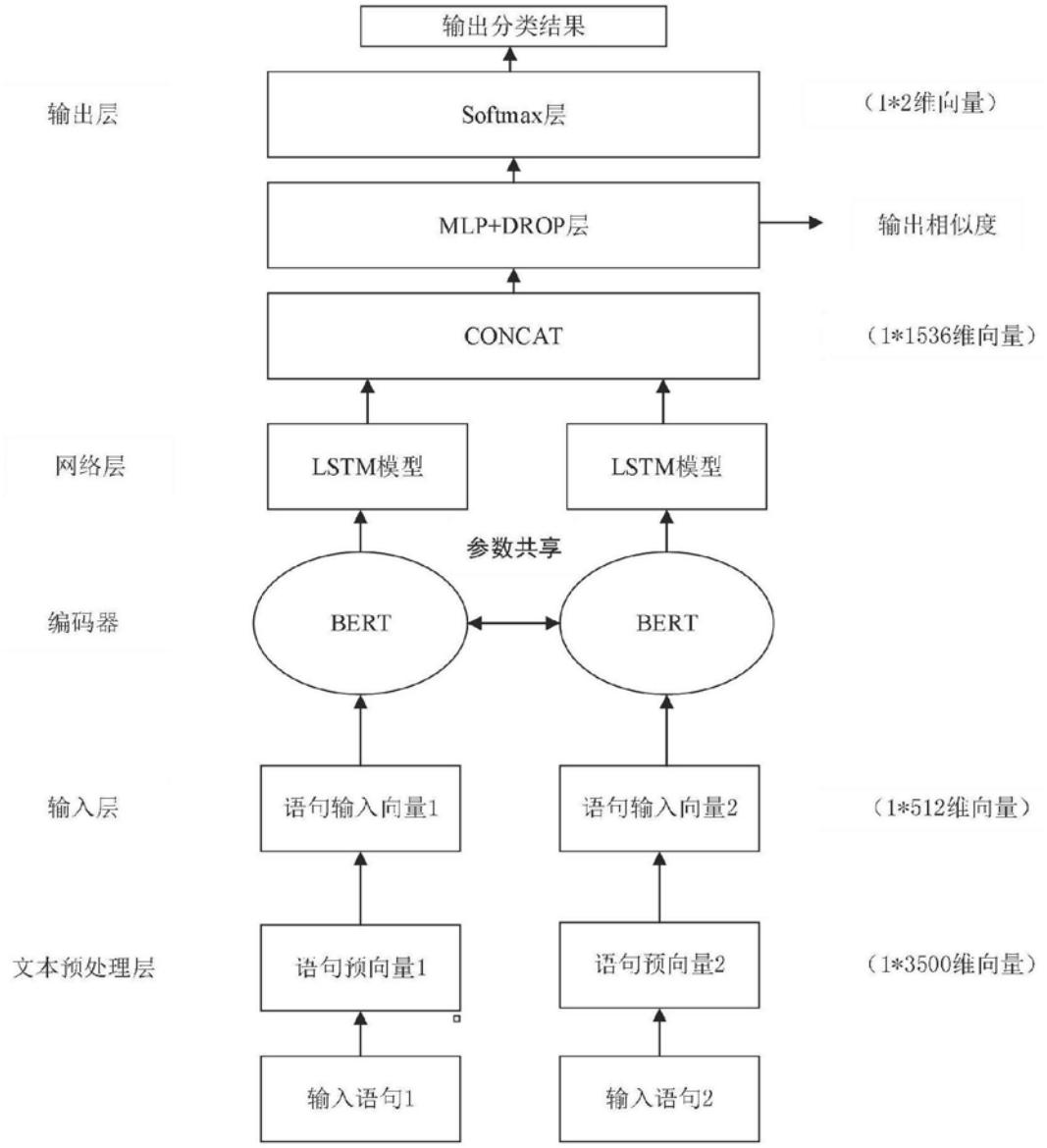

输出分类结果
输出层
Softmax层
（1*2维向量）
MLP+DROP层
输出相似度
CONCAT
（1*1536维向量）
网络层
LSTM模型
LSTM模型
参数共享
编码器
BERT
BERT
输入层
语句输入向量1
语句输入向量2
（1*512维向量）
文本预处理层
语句预向量1
语句预向量2
（1*3500维向量）
输入语句1
输入语句2

图1

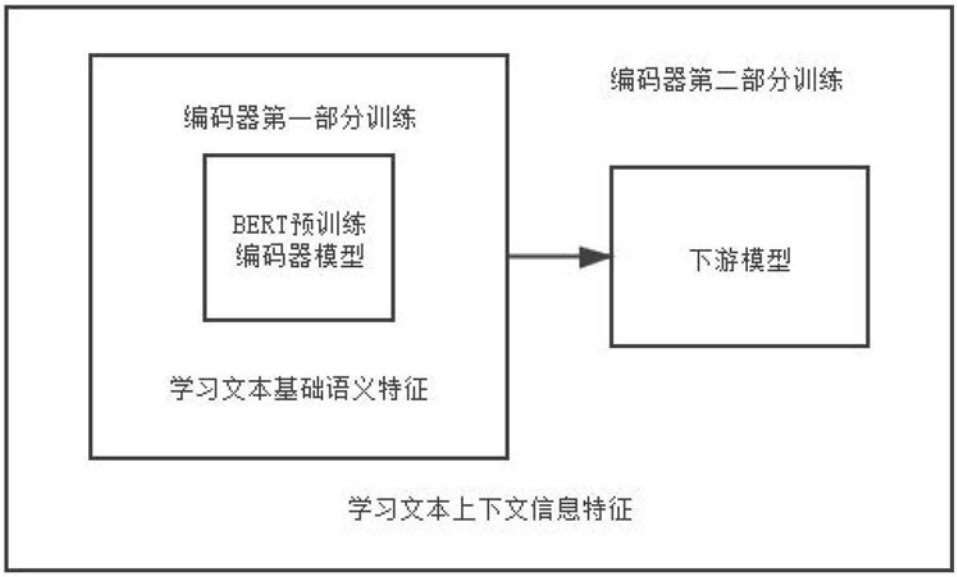

编码器第一部分训练
BERT预训练
编码器模型
学习文本基础语义特征
编码器第二部分训练
下游模型
学习文本上下文信息特征

图2

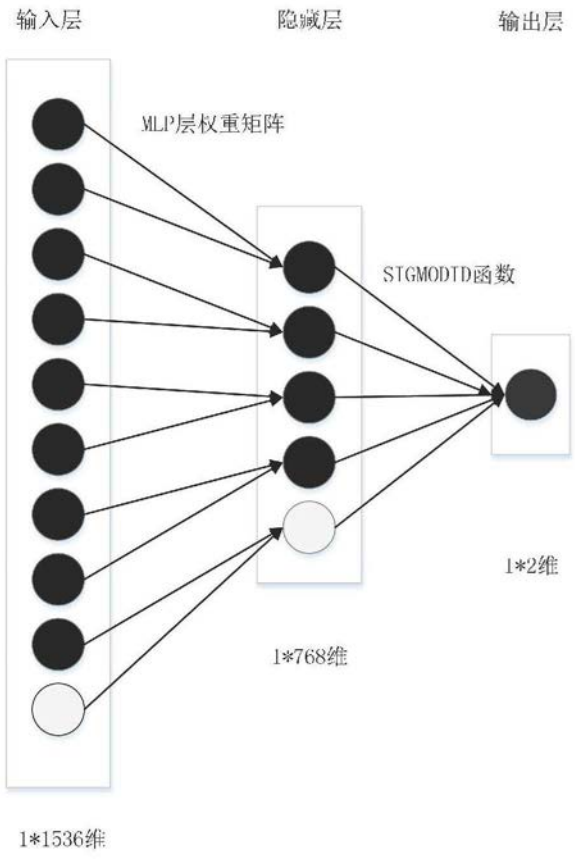

输入层
隐藏层
输出层
MLP层权重矩阵
STGMODTD函数
1*2维
1*768维
1*1536维

图3

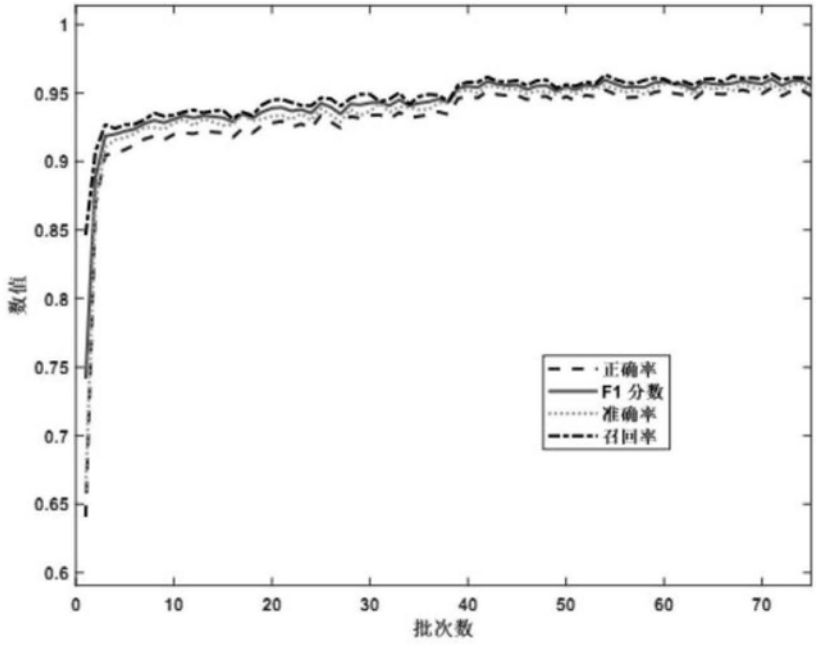

1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
数值
0
10
20
30
40
50
60
70
批次数
正确率
F1 分数
准确率
召回率

图4

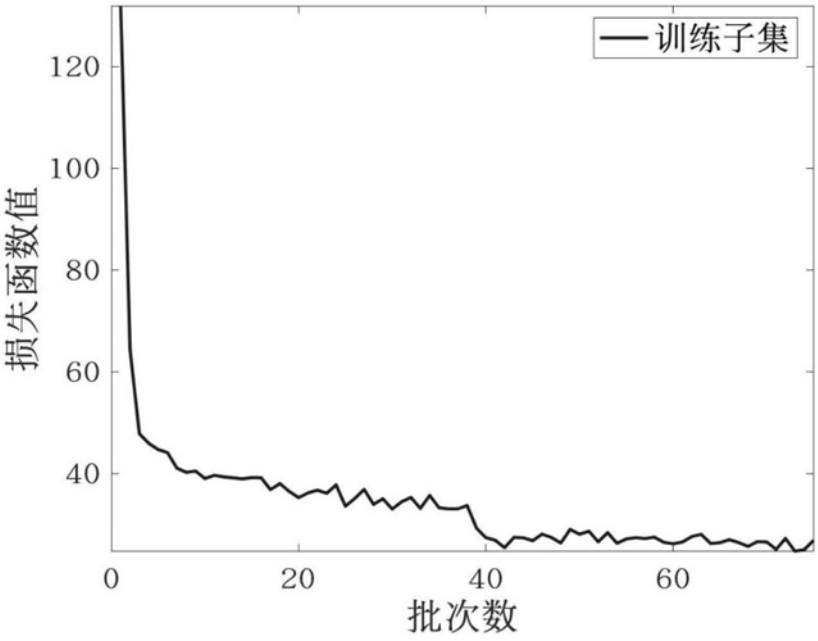

训练子集
120
100
80
60
40
损失函数数值
0
20
40
60
批次数

图5

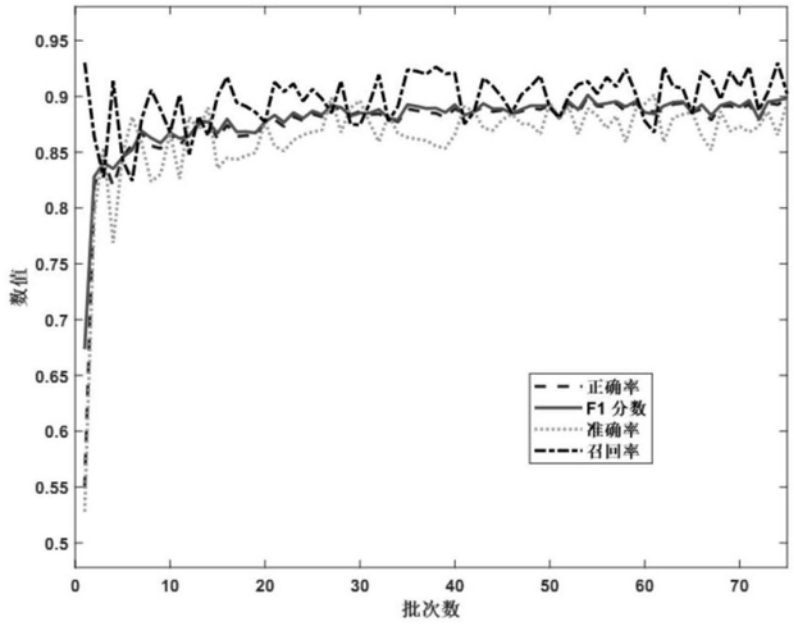

0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
数值
0
10
20
30
40
50
60
70
批次数
正确率
F1 分数
准确率
召回率

图6

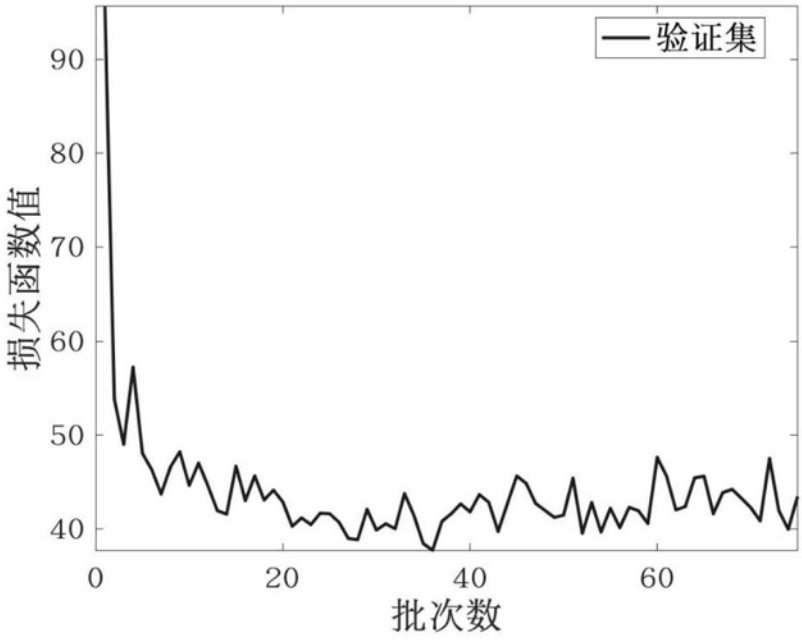

90
80
70
60
50
40
损失函数值
0
20
40
60
批次数
验证集

图7