基于BERTCA的新闻实体与正文语义相关度计算模型

向军毅[1,2], 胡慧君[1,2],毛瑞彬[3] ,刘茂福[1,2]✉
1. 武汉科技大学计算机科学与技术学院, 武汉, 430065
2. 智能信息处理与实时工业系统湖北省重点实验室, 武汉, 430065
3. 武汉大学信息资源研究中心, 武汉, 430072
liumaofu@wust.edu.cn

摘要

目前的搜索引擎仍然存在“重形式，轻语义”的问题，无法做到对搜索关键词和文本的深层次语义理解，因此语义检索成为当代搜索引擎中亟需解决的问题。为了提高搜索引擎的语义理解能力，提出一种语义相关度的计算方法。首先标注金融类新闻标题实体与新闻正文语义相关度语料1万条，然后建立新闻实体与正文语义相关度计算的BERTCA(Bidirectional Encoder Representation from Transformers Co-Attention)模型，通过使用BERT预训练模型，综合考虑细粒度的实体和粗粒度的正文的语义信息，然后经过协同注意力，实现实体与正文的语义匹配，不仅能计算出金融新闻实体与新闻正文之间的相关度，还能根据相关度阈值来判定相关度类别，实验表明该模型在1万条标注语料上准确率超过95%，优于目前主流模型，最后通过具体搜索示例展现该模型的优秀性能。

关键词： 语义相关度计算 ； BERT模型 ； 协同注意力机制 ； 语言模型

Semantic Relevance Computing Model of News Entity and Text based on BERTCA

Junyi Xiang[1,2], Huijun Hu[1,2],Ruibin Mao[3] ,Maofu Liu[1,2]✉
1. School of Computer Science and Technology,
Wuhan University of Science and Technology, Wuhan, 430065
2. Hubei Province Key Laboratory of Intelligent Information Processing and
Real-time Industrial System, Wuhan, 430065
3. Center for Studies of Information Resources, Wuhan University, Wuhan, 430072
liumaofu@wust.edu.cn

Abstract

At present, the search engines cannot understand the semantic deep meaning of keywords and texts, and still face the problem of more attentions to form than semantics. In order to improve the ability of semantic retrieval in contemporary search engines, this paper proposes the semantic relevancy computing method. The corpus with the entity and text semantic relatedness in 10,000 financial news headlines has been constructed firstly by manual annotation, and then the BERTCA (Bidirectional Encoder Representation from Transformers co-attention semantic relevancy computing) model has been established using this corpus. This model has taken both fine-grained entity semantic information and coarse-grained text by BERT pre-trained language mode into consideration. Through the co-attention mechanism, this model can obtain the semantic matching between the entity and text, and it can not only calculate the degree of correlation between entity and text, but also determine the degree of correlation according to the semantic relevancy. The experimental results show that the accuracy of the proposed model has achieved more than 95% on the constructed corpus and is better than the state-of-the-art models.

第十九届中国计算语言学大会论文集，第288页-第300页，海南，中国，2020年10月30日至11月1日。

Keywords: semantic relevance computing , BERT , co-attention mechanism , language model

1 引言

当代互联网飞速发展，人们可以方便地通过搜索引擎获取符合需求的信息。其中，金融新闻的检索占比尤为重要。对关心市场变化的投资者而言，金融类新闻、公告和资讯等数据有着极其重要的参考价值。然而金融数据来源广泛、种类繁杂，直接定位关键信息非常困难。因此，通过实体对海量的金融类数据进行精确有效且快速地定位 (MacAvaney et al., 2019; Ai et al., 2018; Wang et al., 2018; Xiong et al., 2018)，仍然面临着重大挑战。虽然用户一直迫切需要精准检索，但庞杂的无关结果仍是困扰用户的一个顽疾。

目前，大型搜索引擎公司为了提高搜索引擎的效率，主要的解决方案是对每个入库的网页进行分析，得到每个网页的关键词或者标题等部分重要信息，然后再将用户查找的关键词切分，与每个网页的关键词进行匹配，再过滤垃圾结果，最后根据网站的整体评价、网页质量、内容质量、资源质量等指标进行重排。这样就会因为牺牲了全文语义信息，导致搜索引擎的准确度下降，例1是使用搜索引擎搜索“万科”金融新闻的实例：

例1：
标题： 宝能系增持华润旗下东阿阿胶，万科股权事件重演
正文： 万科股权之争稍稍有些停息，“宝能系”又盯上了华润旗下的另一家上市公司东阿阿胶。据东阿阿胶2016年半年财报,宝能旗下的前海人寿二季度再度增持东阿阿胶,目前,持股比例已增至4.17%,逼近举牌线,成为了该公司的第三大股东。同时,华润也在二季度增持了东阿阿胶4.6%的股份,借此巩固其控股地位。分析人士指出,宝能系意图不明,华润显然感到了压力,以防“宝能系”再度搅局,万科事件重演...

在检索阶段，将实体“万科”与新闻标题进行匹配，得到一系列标题中包含“万科”的新闻。但是通过观察返回的结果不一定与该实体相关，新闻正文的核心内容和部分实体语义相关度较小。该新闻标题实体包含“宝能”、“华润”、“东阿阿胶”与“万科”，然而新闻正文主要报道了新闻标题的前半部分，即“宝能系增持华润旗下东阿阿胶”，新闻标题的后半部分“万科股权事件重演”基本上没谈及，新闻正文的主要内容与“万科”本身关系不大，而搜索引擎无法判断这一点，在检索“万科”的时候也会将该条新闻返回，。从金融债券大数据的统计情况来看，这样的新闻数量大约占搜索引擎返回结果的1/5。从用户的角度出发，这样的结果无法准确找到体现所关注公司价值和风险的信息，最终降低用户体验。因此，解决实体与正文的“语义鸿沟”仍然是一项艰巨的任务和挑战。

为了让搜索结果中的高匹配结果排名靠前，采用排序学习(Learning to Rank,LTR)方法对搜索结果进行重排是非常好的选择。目前，使用机器学习方法来进行排序学习取得了非常好的效果 (Burges, 2010)，排序模型依靠多种特征，比如关键词匹配特征、BM25特征、词频、逆文档频率以及PageRank特征等等，对人工相关性标注数据进行学习，从而实现结果排序。使用深度学习方法进行排序也进行了探索 (Köppel et al., 2019)。然而，目前的开源的排序学习数据集都是通过对特征项进行拟合，而并非使用自然语言，例如：在LETOR4.0 (Qin et al., 2010)数据集中，每一条查询文档对包括46个特征。丰富特征确实可以得到相对优秀的结果，但是仍然没法避免关键语义信息丢失的问题。因此，本文通过众包的形式，构建了自然语言形式的新闻实体与正文的语义相关度数据集1万条。之所以使用新闻实体，这是由于金融领域搜索引擎的日志中占比80%以上都是对某个公司进行检索，并且搜索引擎的检索对象也是标题。通过该数据集，模型可以学习到实体与正文的语义相关度，然后通过计算实体与正文的相关度，对搜索结果进行排序，这样既缩短了检索时间，又能达到语义检索的效果。

语义相关度的应用非常广泛，对于该方面的研究也非常火热，目前，语义相关度计算的方法主要分为机器学习方法和深度学习方法两类。机器学习方法主要有TF-IDF、BM25、simHash、VSM等，在粗粒度的文本上达到了可观的效果，但是在对文本的深层次语义理解上存在缺陷。为此，研究人员运用深度学习方法来进一步提高语义理解能力，

主要有包括以DSSM为代表的深层深度学习网络，这些方法都取得了很好的效果，但是仍然存在长距离语义缺失、语义偏移等问题，并且将语义相关度与排序学习综合起来的模型较少。

为了解决上述问题，本文提出基于BERTCA的语义相关度的计算模型BERTCA，通过模型计算标题实体与正文的语义相关度，然后通过语义相关度对搜索结果进行重排。该方法将语义相关度计算方法与排序学习方法结合，充分利用了文本的语义信息，实现了搜索引擎的语义检索功能。标题与正文的语义相关度可以预先计算存入数据库中，因此搜索引擎的速度并不会降低。BERT的多层多头自注意力的堆叠结构使得模型能够在每个层次注意不同的信息，语义偏移问题得到了显著地缓解，而语义交互层可以让实体与正文的语义信息得到充分交互，即使是长距离的文本，也会得到足够的重视，将两者结合起来，就会得到了更好的语义相关度的计算结果。

2 相关工作

近年来，随着深度学习的突破，神经网络模型不仅在图像处理、语音识别领域展现出了较好的性能，也在自然语言处理领域展现出了很大的优势。针对语义相关度计算方法，国内外学者提出了很多研究方法和模型，大体可以分为以下3类 (庞亮et al., 2017)：

(1) 单语义文档表达的计算模型：简单地使用全连接层、卷积神经网络或循环神经网络将文本表达成一个稠密向量，然后直接计算两个向量间的相关度。 Huang et al. (2013)探索了一种潜在语义模型DSSM，该模型是一个具有深层结构的潜在语义模型，它会将查询和文档投射到一个常见的低维空间中，这样子很容易将给定的文档和查询的相关性计算为它们之间的距离； Kim et al. (2019)提出密集连接协同注意力循环神经网络模型(DRCN)，将循环神经网络中的隐含层与协同注意力层进行拼接，这样原始信息就会在最底层到最顶层之间一直保留，而在循环神经网络每一个块中，使用协同注意力的方式得到两个句子之间的交互信息，因为参数量的迅速增加，影响模型训练，因此用自编码器进行压缩表示。

(2) 多语义文档表达的计算模型：此方法认为单一粒度的语义信息无法精细地描绘出文本所有内容，因此建立了多语义表示，让两段文本进行交互，挖掘文本交互后的模式特征，综合得到文本间的相关度。 Wan et al. (2016)提出了MV-LSTM来解决模型无法捕获上下文相关的局部信息的问题，利用长短时记忆神经网络(Long Short-Term Memory，LSTM)获取上下文语义向量，通过k-Max池化和多层感知器变换，最后聚合这些不同位置的句子表示之间的交互，最终产生语义相关度得分；(3) 直接计算模型：为了进行更深层次的语义信息交互，应该考虑不同层次的交互信息，更精细的处理句子中的联系，再用深度神经网络挖掘交互后的模式特征，综合计算文本之间的相关度。 Lu and Li (2013)提出了DeepMatch网络，通过使用文档主题生成模型(Latent Dirichlet Allocation，LDA)模型获取两个文本的共现情况，从而得出两个文本的匹配分数； Xu et al. (2019)提出了一种文本匹配的多层次匹配网络(MMN)，该网络利用多个词的表示来获得多个词的水平匹配结果，从而进行最终的文本水平评分。

深度学习模型使词向量的具有线性的语义信息，但是仍然没有解决语义偏移问题，研究者在这个方面进行了大量的探索，其中预训练语言模型表现的非常好。 Devlin et al. (2018)提出了一种新的语言表示模型BERT，它是由Transformers (Vaswani et al., 2017)的双向编码器表示，与其他的语言表示模型不同，BERT会利用未标记文本来预训练深层双向表示，这种双向表示不仅会对上文进行编码，还会对下文进行编码。另外还有许多BERT模型的变体：ALBERT (Lan et al., 2019)、XLNet (Yang et al., 2019) RoBERTa (Liu et al., 2019)。其中XLNet是一种泛化的自回归预训练模型，通过最大化所有可能的因式分解顺序的对数似然，学习双向语境信息，用自回归本身的特点克服BERT的缺点，并融合了最优自回归模型Transformer-XL (Dai et al., 2019)的思路，在各项任务上超过BERT创下的记录。

排序学习 (Liu, 2011; Yin et al., 2016; Wang et al., 2017)是一个监督学习，通过对每一个查询文档对进行特征抽取，训练排序模型，使得输出与标签相符。常见的排序学习方法一般分为三类：单文档型(Pointwise)，文档对型(pairwise)，文档列表型(Listwise)。

单文档方法只对单个文档进行处理，将文档转换为特征向量，根据训练数据得到的模型对其进行打分，再将所有文档按照得分结果进行排序。主要包括以下算法：Pranking, OC SVM，McRank等。

基金项目：深圳证券信息有限公司联合研究计划(No.2018002)；全军共用信息系统装备预先研究项目（31502030502）；

文档对方法将相关性得分转换为文档对关系，根据标注信息，A的得分为2，B的得分为4，C的得分为1，可以得到B>A,B>C,A>C的比较关系，这样就把排序问题转化成了二分类问题，模型通过对任意两个文档之间的关系进行分类，得到全集的排序关系。主要包括以下算法：LambdaMART (Wu et al., 2010)、RankNet、Ranking SVM等。

文档列表方法的输入为一个文档序列，通过构造合适的度量函数来优化排序，得到排序模型。主要包括以下算法：AdaRank、SVM MAP21、Soft Rank。

3 ETSR语料构建

针对ETSR语料集的构建，爬取多个金融网站的近10年新闻，制定了实体与正文语义相关度标注规范，标注了20,000条实体与正文语义相关度语料，语料集构建整体流程如图1所示。

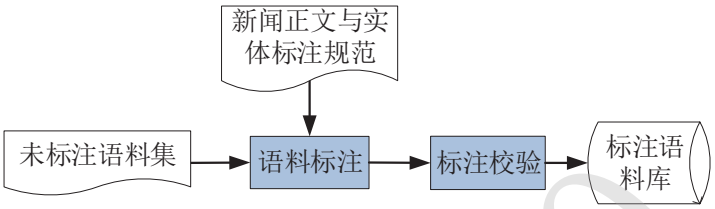

Figure 1: 语料标注流程图

3.1 语料获取

本文使用开源的Scrapy爬虫框架，对21世纪经济报道[1]、财新网[2]、每经网[3]、生意社[4]、人民网[5]、新浪上市公司[6]和腾讯上市公司[7]共7个网站爬取了45,000多条金融类新闻。由于新闻门户网站存在大量抄袭或者搬运的状况，剔除重复新闻后余下的41,000条新闻。

为了后面的语料标注的方便，首先对新闻标题进行了命名实体识别，剔除了不存在公司名、人名、组织名的新闻，保留了20,000条新闻作为未标注语料集。

3.2 语料标注

对于语料标注工作，采用众包的方式进行。同一个新闻文本，至少5位以上标注者同时标注，采用少数服从多数原则，由人工校验后，得到最终的标注语料库

人工标注内容包括实体标注和相关度标注两项。为了制订实体标注规范和相关度标注规范，共经过了8轮的交叉验证，共5,000条语料，剩余的15,000篇新闻将通过众包形式进行标注。

实体标注规范是对新闻标题中的实体进行标注，标题中可能包含多个实体或者嵌套实体，还有一些不确定的实体，其中标题实体为命名实体工具识别出来的实体，关键词实体为那些可能是实体的关键词，例如：“微信”不是一个公司名，只是腾讯公司推出的一个为智能终端提供即时通讯服务的免费应用程序。

(1) 全模式：兼顾最长原则和最短原则，标出所有可能的实体。

例1：
标题：“恒大健康轮值董事长张三”
标注：依次根据公司、职称、人名标注“恒大”、“恒大健康”、“恒大健康轮值董事长”、“张三”等四个实体

(2) 标题实体优先于关键词实体的原则：当且仅当标题中的实体无法覆盖正文内容或者不足以作为新闻正文关键词时，加标关键词实体。

[1]21世纪经济报道：http://www.21jingji.com/channel/finance
[2]财新网：http://finance.caixin.com/
[3]每经网：http://finance.nbd.com.cn/
[4]生意社：http://www.100ppi.com/kx/
[5]人民网：http://finance.people.com.cn/
[6]新浪上市公司：http://vip.stock.finance.sina.com.cn/
[7]腾讯上市公司：https://finance.qq.com/stock/

例2：
标题：“滴滴整改重组，顺风车何去何从”；
标注：优先标注“滴滴”，但是该实体无法覆盖正文内容，随后再加标“滴滴顺风车”；

语义相关度标注则是标注出标题实体与新闻正文是强相关、弱相关、不相关。语义相似度标注的原则是依据正文与实体的相关程度来界定。(1)对于报道当前标注新闻的媒体机构，如果正文没有特意介绍，一般为不相关。(2)新闻正文讲述发生在子公司身上的事情时，母公司(短名字)一般为弱相关，例如“网易云音乐”和“网易”。(3)有出现在正文的标题实体一般为不相关。

例3：
标题：TCL进军互联网电视，意欲赶超小米、乐视。
标注：新闻正文中几乎没有提及小米、乐视，二者皆为不相关。

由于人工标注对命名实体识别的公司人名实体有部分调整，因此语料标注完成后，还需要剔除标题中既没有关键词实体也没有标题实体的新闻，最终挑选了10,000条高质量的标注语料进行实体与新闻正文的语义相关度计算。

3.3 语料分析

所有语料标注结果存为JSON格式，并以UTF-8格式编码，标题最长为57个字符，其中0~35区间占比为94.8%，正文最长有10000个字符，其中0~2000个字符占比为87.2%，可见正文文本都是偏长的。但是经过阅读接近100条数据，发现正文的前200个字符能涵盖新闻正文大部分内容，因此在后续的处理中，可以合理的利用这个特点。

id: 5c39b52be65e992d1c4d74f3
title: 鲁亿通30亿收购比特币芯片商
title_entity: {”鲁亿通”: ”强相关” }
content: 比特币价格今年再现疯狂，22天涨幅超过60%;更加土豪的是,上市公司鲁亿通拟以30亿元收购一家主营业务为比特币“矿机”制造芯片的公司。重组...

Figure 2: 标注语料示例

然后，我们对10,000条语料中不同相关度的实体进行统计，统计结果如表1所示。

Table 1: 实体相关度统计结果

强相关	弱相关	不相关	总计
12,589	1,801	1,142	15,532

从表1统计结果可以看出，强相关数据偏多，弱相关和不相关数据相对偏少。这也证实了引言中的大数据统计结论，弱相关与不相关的数据占比大约1/5，从标注数据统计分析的角度验证了前述的不相关或者弱相关新闻大约占搜索引擎返回结果的1/5的假设。

4 语义相关度计算模型BERTCA

语义相关度计算模型BERTCA的整体结构如图3所示，包含动态语义编码层、上下文编码层、语义交互层和解码层四个部分，上下文编码层是LSTM (Xingjian et al., 2015)，解码层是Fusion LSTM (Liu et al., 2016)。模型的输入是实体和新闻正文两段文本，通过BERT模型得到文本的动态语义编码后，然后通过LSTM对文本进行上下文编码，然后通过协同注意力的语义交互，最后通过解码层得到两者之间的语义相似度得分。

定义输入实体为$\boldsymbol{X} = \{x_1, x_2, \ldots, x_N\}$ 和新闻正文$\boldsymbol{Y} = \{y_1, y_2, \ldots, y_M\}$ ，其中x_i和y_j分别是实体$\boldsymbol{X}$和新闻正文$\boldsymbol{Y}$中的第i和第j个字，N和M为实体X和新闻正文Y的长度。

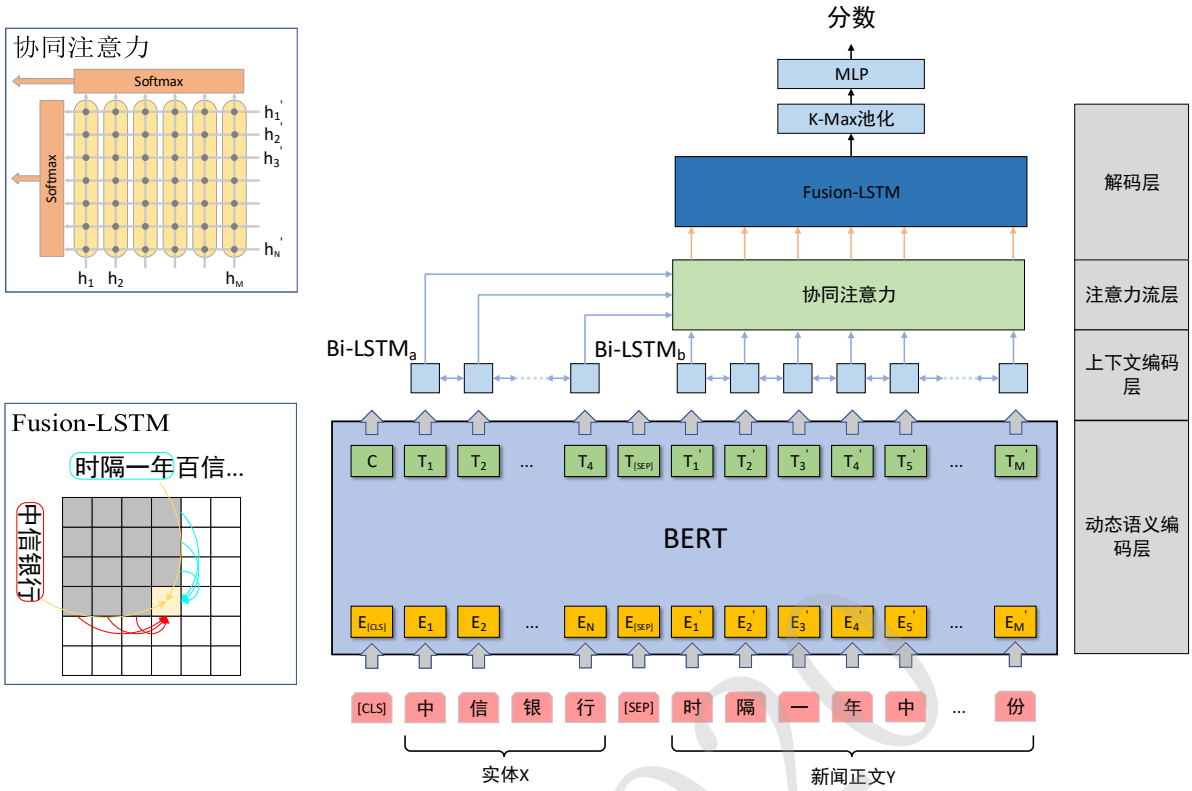

Figure 3: BERTCA语义相关度计算模型

4.1 动态语义编码层

本文通过使用预训练语言模型 (Cui et al., 2019)来构建文本动态语义编码。由于谷歌官方发布的预训练语言模型中，中文是以字为粒度进行切分，没有考虑中文分词的问题，因此该模型使用了全词遮罩(Whole Word Masking)技术来进行训练。

首先，将每个词转换成BERT模型输入的格式，包括索引、遮罩、位置和文本类型编码，这些编码会在BERT内部的嵌入层转换为向量形式，其中索引编码会转换成字向量$E_i = Emb(x_i)$ 和$E'_j = Emb(y_j)$。

其次，这些向量通过多层的编码结构，如公式(1)

$$Layer_{output} = Layer(x + \text{SubLayer}(x)) \tag{1}$$

其中x可以代表上述任何一种编码的向量形式。这样，让模型就拥有了多层结构相同但权重不同的自注意力，每一个注意力头都能关注到不同的特征，模型整体就会关注到更多的特征，如公式(2)(3):

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Con(h_1, \ldots, h_h)\boldsymbol{W}^o \tag{2}$$

$$head_i = Attention\left(\boldsymbol{Q}\boldsymbol{W_i}^Q, \boldsymbol{K}\boldsymbol{W_i}^K, \boldsymbol{V}\boldsymbol{W_i}^V\right) \tag{3}$$

为了解决深层次神经网络中出现的退化问题，每一个Encoder都加入了残差网络和层归一化，如公式(4)(5)。

$$LN(x_i) = \alpha \times \frac{x_i - \mu_L}{\sqrt{\sigma_L^2 + \varepsilon}} + \beta \tag{4}$$

$$FFN = \max(0, x\boldsymbol{W_1} + b_1)\boldsymbol{W_2} + b_2 \tag{5}$$

代码中的注意力机制采用了缩放点积，$\boldsymbol{Q}$表示查询，$\boldsymbol{K}$为键字，$\boldsymbol{V}$为键值，都是输入的字向量。其核心思想是计算一句话中每个字与其他字之间的关联程度，并利用这种关联程度来调整字在句子中的权重，这个字向量不仅蕴含了自己本身的意思，还蕴含了与它相关联的字的关系，因此它能根据上下文对字的表征进行调整，如公式(6)。

$$Attention\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)=softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{T}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{6}$$

整个动态语义编码层的输出是每个字的语义向量。其中[CLS]表示的是整个句子的语义向量，T_i表示实体$\boldsymbol{X}$第i个字的动态语义表示，T'_j表示的是新闻正文$\boldsymbol{Y}$中第j个字的动态语义表示。

最后，将E_i与T_i、E'_j和T'_j分别进行拼接得到实体向量$d_i=[E_i,T_i]\,(1\le i\le N)$、新闻正文向量$d'_j=\left[E'_j,T'_j\right]\,(1\le j\le M)$。

4.2 上下文编码层

上下文编码层采用了双向长短时记忆网络(Bi-LSTM)来对输入文本序列进行编码，因为实体与新闻正文是不同的文本，因为两个网络参数并不共享，t位置的隐含状态输出为：

$$h_t=\left[\overrightarrow{LSTM_a}(d_t,h_{t-1})^T,\overleftarrow{LSTM_a}\right](d_t,h_{t-1})^T] \tag{7}$$

$$h'_t=\left[\overrightarrow{LSTM_b}(d'_t,h'_{t-1})^T,\overleftarrow{LSTM_b}\right](d'_t,h'_{t-1})^T] \tag{8}$$

其中，$(\cdot)^T$表示转置操作，$\overrightarrow{LSTM}(\mathrm{d_t},\mathrm{h_{t-1}})^{\mathrm{T}}$和$\overleftarrow{LSTM}](d_t,h_{t-1})^T$分别表示长短时记忆网络在$t$时刻的正向输出和反向输出，然后将网络的正向输出与反向输出拼接起来，$\mathrm{h_i}\,(1\le i\le N)$为实体语义向量，$\mathrm{h'_i}\,(1\le i\le M)$为新闻正文的语义向量。

4.3 语义交互层

语义交互层是实现实体与正文的相互理解的重要机制，本文主要采用Co-attention，它是注意力机制的一种变体，不仅要给阅读的新闻正文生成一个注意力权重，还要给实体也生成一个注意力权重，然后利用两者的注意力权重，得到经过修正后的语义向量。

首先，计算两段文本的相关性矩阵L，然后根据新闻正文的每一个字计算实体中每一个字的注意力分数A，同理，可以根据实体的每一个字计算新闻正文中每一个字的注意力分数A'：

$$\boldsymbol{L}=\left(\boldsymbol{H'}\right)^{T}\boldsymbol{H}\left(\boldsymbol{L}\in R^{MN}\right) \tag{9}$$

$$\boldsymbol{A}=\mathrm{softmax}\left(\boldsymbol{L}\right) \tag{10}$$

$$\boldsymbol{A'}=\mathrm{softmax}\left(\boldsymbol{L}^{T}\right) \tag{11}$$

然后，通过$\boldsymbol{A'}$来对实体的语义向量进行加权，得到经过注意力修正后的实体向量$\boldsymbol{F}$，再利用实体语义向量$\boldsymbol{H}$和修正后的实体语义向量$\boldsymbol{F}$，经过新闻正文注意力分数$\boldsymbol{A}$得到修正后的实体语义向量$\boldsymbol{G}$，同理，可以得到修正后的新闻正文语义向量$\boldsymbol{G'}$：

$$\boldsymbol{F}=\boldsymbol{H}\,\boldsymbol{A'}\left(\boldsymbol{F}\in R^{l*M}\right) \tag{12}$$

$$\boldsymbol{G}=\left[\boldsymbol{H'},\boldsymbol{F}\right]\,\boldsymbol{A}\left(\boldsymbol{G}\in R^{2l*N}\right) \tag{13}$$

$$\boldsymbol{F'}=\boldsymbol{H'}\,\boldsymbol{A}\left(\boldsymbol{F}\in R^{l*N}\right) \tag{14}$$

$$\boldsymbol{G'}=\left[\boldsymbol{H},\boldsymbol{F'}\right]\,\boldsymbol{A'}\left(\boldsymbol{G}\in R^{2l*M}\right) \tag{15}$$

其中，$[\cdot,\cdot]$表示两个矩阵在第1维进行拼接。$\boldsymbol{G}$和$\boldsymbol{G}'$，作为下一层的输入，分别是实体与新闻正文的协同语义表示，是经过相互理解后的文本语义表示。

4.4 解码层

在解码层，使用了由 Liu et al. (2016)提出来的DF-LSTMs(Deep Fusion LSTMs)网络，它包含了两个独立的LSTMs (Wu et al., 2017)来挖掘更高层次的文本语义信息，对实体与新闻正文进行多次递归融合。

定义$g_{1:i}$表示实体的子序列$\{g_1, g_2, \ldots, g_i\}(1 \leq i \leq N)$，$g'_{1:j}$表示新闻正文的子序列$\{g'_1, g'_2, \ldots, g'_j\}(1 \leq j \leq M)$，$I_{i,j}$表示子序列$g_{1:i}$和$g'_{1:j}$之间的相互作用，

$$h_{i,j} = \mathrm{h}^{g}_{i,j} \oplus h^{g'}_{i,j} \tag{16}$$

其中，$I^{g}_{i,j}$表示经过新闻正文LSTM的隐藏层编码的实体输出编码，$I^{g'}_{i,j}$表示经过实体LSTM的隐藏层编码的新闻正文输出编码。整个网络推演的过程如下：

$$h^{g}_{i,j}, c^{g}_{i,j} = LSTM\left(h_{i,j}, c^{g}_{i-1,j}, g_i\right) \tag{17}$$

$$h^{g'}_{i,j} c^{g'}_{i,j} = LSTM\left(H_{i,j}, c^{g'}_{i-1,j}, g'_i\right) \tag{18}$$

其中，$H_{i,j}$是包含过去信息的隐藏层，此方法可以看成Grid LSTMs (Kalchbrenner et al., 2015)的一种变种。

为了计算两段文本交互匹配信息，选用了Cosine(余弦相似度)、Bilinear(双线性变换)和Tensor Layer(张量变换)三种相似性度量方法来进行综合评定分数，给定两个向量u和v，三个相似度评分$s\,(u,v)$分别为:

$$Cosine : s\,(u,v) = \frac{u^T v}{u \cdot v} \tag{19}$$

$$Bilinear : s\,(u,v) = u^T \boldsymbol{M} v + b \tag{20}$$

$$TensorLayer : s\,(u,v) = f\left(u^T \boldsymbol{M}^{[1:c]} v + \boldsymbol{W}_{uv} \begin{bmatrix} u \\ v \end{bmatrix} + b\right) \tag{21}$$

其中，$\boldsymbol{M}$是两短文本交互的权重矩阵，$\boldsymbol{M}^i, i \in [1, \ldots, c]$是张量参数的一个切片，$\boldsymbol{W}_{uv}$和$b$是线性部分参数，$f\,(z) = \max\,(0, \mathrm{z})$。Cosine和Bilinear函数输出形式为矩阵，而Tensor Layer函数输出的形式为张量。Cosine相似度计算是一种常用做法，而Bilinear能够考虑不同维度之间的关联信息，因此相比Cosine方法能够捕获更复杂的交互信息。Tensor Layer在建模两个向量之间相互关系表现出了较大的优越性，且能够退化为Bilinear和点积相似度度量方法。

需要通过K-max池化层 (Shu et al., 2018)来整合这三种语义交互信息来得到最终的匹配得分。对于Cosine和Bilinear，经过K-Max池化层后可以得到K个数值，并以降序排列，组成一个新的向量q 。

对于Tensor Layer，每个张量切片返回K个数值形成一个向量，所有张量切片返回向量拼接在一起生成向量q。把向量q输入多层感知机(MLP)以获取更深层次交互向量表示r，然后用一个线性变换输出匹配得分值s，

$$s = W_s f\,(W_r q + b_r) + b_s \tag{22}$$

其中，W_r、W_s分别是权重矩阵，b_r和b_s为偏置向量，$f\,(\cdot)$是tanh函数。

5 实验结果分析

5.1 实验设置

由于网上开源的LTR数据集并没有给出相应的查询文本，而是给出若干个特征项，很难用作做对比数据集，因此本文选用文本匹配数据集STS-B、QQP、MRPC作为对比数据集。同时，本文提出的BERTCA模型将与DRCN (Kim et al., 2019)、ESIM (Chen et al., 2016)、BERT (Devlin et al., 2018) 、XLNetCA (Yang et al., 2019)、MV-LSTM (Wan et al., 2016)、ALBERTCA (Lan et al., 2019)、RoBERTaCA (Liu et al., 2019)进行对比实验，XLNetCA的意思就是在XLNet模型的基础上增加语义相关度计算模型，其他模型以此类推。

在实验参数方面，BERT中的隐藏层维度为1024，隐藏层激活函数为GELU，注意力层数为24层，注意力层dropout率为0.1，注意力头数为16个，词表大小为21128，最大位置编码为512。所有LSTM隐含层维度设为320，全连接网络MLP的隐含层维度设为128、32，dropout率为0.15，Tensor Layer中张量的c设为8，K-Max层的卷积核的宽度为2*2。

训练参数batchsize设为64，AdaW (Loshchilov and Hutter, 2017)优化算法的初始学习率设为1e-5，迭代训练次数epoch=300。语句序列的长度分别为ETSR512，其中实体为10，新闻正文为499，STS-B取128，MRPC取128，QQP取128。

所有任务都转化为排序任务，模型要使正样本排名比负样本高，因此选用精准度P@1和平均倒数排名MRR(Mean Reciprocal Rank)作为评估指标：

$$P@1 = \frac{1}{N}\sum_{i=1}^{N}\delta\left(r\left(S_Y^{+(i)}\right)=1\right) \tag{23}$$

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r\left(S_Y^{+(i)}\right)} \tag{24}$$

其中，N是排序列表的长度，$S_Y^{+(i)}$指第i个排序列表中的正样本语句，$r(\cdot)$表示排序列表中语句的排名，δ是一个指示函数，即$\delta(true)=1$,$\delta(false)=0$。

5.2 结果分析

Table 2: 在ETSR、STS-B、QQP及MRPC数据集上的评估结果

模型	ETSR		STS-B		QQP		MRPC	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
DRCN	0.812	0.842	0.723	0.772	0.739	0.783	0.739	0.798
ESIM	0.845	0.873	0.693	0.750	0.715	0.763	0.754	0.801
MV-LSTM	0.864	0.896	0.741	0.785	0.703	0.754	0.772	0.830
BERT	0.864	0.882	0.870	0.927	0.864	0.935	0.843	0.901
XLNetCA	0.863	0.899	0.874	0.930	0.882	0.931	0.882	0.925
ALBERTCA	0.899	0.925	0.905	0.936	0.867	0.904	0.874	0.907
RoBERTaCA	0.902	0.933	0.902	0.931	0.876	0.913	0.905	0.933
BERTCA-Cosine	0.943	0.974	0.931	0.968	0.922	0.948	0.932	0.966
BERTCA-Bilinear	0.942	0.972	0.903	0.963	**0.938**	**0.962**	0.898	0.962
BERTCA-TL	**0.954**	**0.988**	**0.919**	**0.969**	0.882	0.951	**0.933**	**0.973**

本模型的参数量为370M，在参数量为330M的BERT基础上只增加了40M的参数量。经过计算，在8个Tesla P100 16G上的运行一个批次大约需要16分钟，而在XLNetCA、RoBERTaCA等模型的上大约需要20分钟，时间相差较大。表2显示了本文所提出的模型在ETSR、STS-B、QQP和MRPC数据集上与其他模型的对比结果，BERTCA-Cosine、BERTCA-Bilinear、BERTCA-TL分别代表计算语义相似度时所采用的三种不同计算方法。其中DRCN为单语义文档表达的计算模型，ESIM、MV-LSTM和BERTCA为多语义文档表达的计算模型，其他的都是直接计算模型。

从该表中可以发现BERTCA模型在三种相关度计算方式下结果略有差异，TL的效果普遍偏好，猜测该方法能够捕获两个文本不同位置特征向量之间的差异，相比于Cosine和Bilinear方

Table 3: 去除解码层、语义交互层层、上下文编码层在ETSR上的消融实验

模型	P@1	MRR
BERTCA-TL	**0.954**	**0.988**
-Decoder	0.923	0.967
-Coattention	0.889	0.925
-Context	0.912	0.953

法能够学到更有意义的交互信息。我们还发现Bilinear相似度计算方法在QQP数据集上的效果比较好，经观察发现QQP的两段文本的长度集中在5~30之间，并且文中的句式较为简单，词表前2000个词可以覆盖95%的句式，句式的变化较少导致最终形成的语义向量变化平顺，因此语义信息可能记录很完整，经过Bilinear计算后，得到了较好的结果。通过对比BERTCA模型与其他模型的效果，BERTCA-TL除了在MRPC上的效果略逊色于BERTCA-Bilinear以外，其他数据集上，达到了最优效果，说明本文提出的基于BERTCA模型在ETSR、STS-B、QQP和MRPC这3个数据集上具有明显的提升。

除此之外，为了更好地衡量不同网络参数对实验结果的影响，本文也进行了对比实验，通过选用不同参数量的预训练模型[8]，对模型的效果进行了验证。通过观察表4可以发现，虽然ALBERT或者XLNet对BERT模型进行了一定程度的改进，但是最终呈现出来的效果却不是最好的。得益于ALBERT的参数共享和词向量分解机制，ALBERT的参数量大大减少，运算速度大大提高，然而却破坏有效的语义信息，效果有所衰减，XLNet的训练模式则是词生成，语义理解的能力还有待提高。

Table 4: 不同参数量下的模型效果

模型	P@1	MRR
BERT-chinese-wwm-ext	**0.864**	**0.882**
BERT-chinese-wwm	0.834	0.850
BERT-chinese-base	0.776	0.798
ALBERT-chinese-Large	0.815	0.841
ALBERT-chinese-base	0.782	0.806
XLNet-chinese-mid	0.842	0.869
XLNet-chinese-base	0.822	0.846

本文还对语义解码层的Fusion-LSTM隐含层不同维度进行了分析。通过表5可以发现，当维度在160以下时，Fusion-LSTM的模型效果普遍偏低，而当维度到达了256维，模型呈现出了震荡的趋势，在320维的时候，到达了顶点，因此本文选用了320维作为最终模型的参数。

Table 5: 不同参数量下的模型效果

维度	P@1	MRR	维度	P@1	MRR
32	0.443	0.462	288	0.934	0.959
64	0.863	0.882	**320**	**0.954**	**0.988**
96	0.873	0.899	352	0.931	0.951
128	0.867	0.891	384	0.949	0.964
160	0.892	0.916	416	0.951	0.974
192	0.921	0.941	448	0.933	0.948
224	0.914	0.935	480	0.943	0.962
256	0.945	0.961	512	0.922	0.941

在ETSR数据集上，与最优的BERTCA-TL模型对比结果显示，去除解码层模块之后，模型的整体性能有明显的下滑，在P@1和MRR指标上分别下降了0.031和0.021，在没有解码层的情况下，语义交互的结果没法很有效的展现，我们曾将解码层替换成单纯的MLP网络，实验效果反而下降了，可见解码层在整个模型中的作用。另外，去除了语义交互层后，解码层得到了保留，但是在P@1和MRR指标上依然下降了0.065和0.063，可以发现实体与正文之间的语义交互层的重要性。对比“-Coattention”和“-Context”，“-Context”的实验结果反而有一定的提升，这是因为语义交互层的缺失，导致整体效果的下降，也从侧面表示了上下文编码层在整体结构上的提升偏小，而语义交互在整体结构上的重要性偏大。

[8] https://huggingface.co/models

(a) 无BERTCA　　(b) 有BERTCA

Figure 4: 基于语义相关度排序实例

为了更加形象的展示模型的成果，本文利用Solr开源搜索引擎框架对爬取的新闻进行可视化展示，如图4。该搜索引擎将搜索关键词与新闻标题进行比对，返回匹配结果，以关键词“360 公司”为例，搜索引擎返回了“东风小康风光360”、“经典落幕Xbox360正式停产！”这两条与“360 公司”关键词看似很匹配的结果，其实用户是想对360公司进行搜索，因此所有与360公司的有关的新闻应该在搜索结果的前列。

下面将BERTCA模型与搜索引擎结合，对搜索结果进行重排。首先通过训练好的BERTCA模型计算“360”与正文的语义相关度相关度，第一条新闻“360手机新掌门首次亮相否认360放弃手机业务”与“360”的语义相关度为0.964，说明实体与正文是密切相关度的，而在第3条新闻中，虽然标题中包含“360”，但是其实它是想表达“Xbox360”，在计算“360”与该条新闻的语义相关度时候，BERTCA模型发现了这种差别，语义相关度的结果为0.744，相较于上一条新闻，具有较大的差距。

6 结束语

针对搜索引擎存在“重形式，轻语义”，无法对搜索关键词和文本进行深层次语义理解的问题，本文提出了一种基于BERTCA的语义相关度计算模型，该模型融合了BERT动态语义编码方法和深度语义信息交互的协同注意力方法，并将排序方法与语义相关度方法进行结合，在ETSR、STS-B、QQP和MRPC这四个数据集上进行了对比实验，结果证明本文提出的模型能有效提升语义相关度的计算结果，并对搜索引擎的语义理解能力有较大的帮助。

参考文献

Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 135–144.

Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. 2015. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 6:19–34.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6586–6593.

Marius Köppel, Alexander Segner, Martin Wagener, Lukas Pensel, Andreas Karwath, and Stefan Kramer. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 237–252. Springer.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Pengfei Liu, Xipeng Qiu, Jifan Chen, and Xuan-Jing Huang. 2016. Deep fusion lstms for text semantic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1034–1043.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tie-Yan Liu. 2011. *Learning to rank for information retrieval*. Springer Science & Business Media.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in neural information processing systems*, pages 1367–1375.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374.

Bo Shu, Fuji Ren, and Yanwei Bao. 2018. Investigating lstm with k-max pooling for text classification. In *2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pages 31–34. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*, volume 16, pages 2835–2841.

Liang Wang, Sujian Li, Yajuan Lü, and Houfeng Wang. 2017. Learning to rank semantic coherence for topic segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1340–1344.

Sheng Wang, Zhifeng Bao, J Shane Culpepper, Zizhe Xie, Qizhi Liu, and Xiaolin Qin. 2018. Torch: A search engine for trajectory data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 535–544.

Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270.

Wei Wu, Houfeng Wang, and Sujian Li. 2017. Bi-directional gated memory networks for answer selection. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 251–262. Springer.

SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, volume 8, pages 802–810.

Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards better text understanding and retrieval through kernel entity salience modeling. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 575–584.

Chunlin Xu, Zhiwei Lin, Shengli Wu, and Hui Wang. 2019. Multi-level matching networks for text matching. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 949–952.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, et al. 2016. Ranking relevance in yahoo search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–332.

庞亮, 兰艳艳, 徐君, 郭嘉丰, 万圣贤, and 程学旗. 2017. 深度文本匹配综述. 计算机学报, 40(4):985–1003.