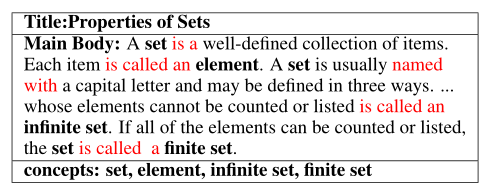


IJCAI 2021

问题：概念提取（与关键词提取不同的是，概念提取是提取全部概念，而不是顶层概念）



目标：从title和main body中提取出一些concepts。

传统的提取方法：

1. 使用规则(手工或语法模式匹配)提取候选短语。

2. 基于监督或非监督方法对其进行排名。

* 在无监督方法中：关键字提取采用不同的技术，如基于图的排序、聚类和语言建模等，将其表述为一个排序问题。
* 在监督方法中：关键词提取被视为一个分类问题，文档中的每个短语都是或不是一个关键词短语。

缺点：主要使用统计特征，无法捕获文本背后的语义。

近年的提取方法：

基于深度神经网络(DNN)的模型被用于解决概念提取问题。

* Encoder-decoder框架直接从文本中提取概念。
* 将概念提取作为序列标注问题。

命名实体提取（NER）vs关键词提取vs概念提取：

NER和概念抽取都可以定义为序列标注问题。

NER的目标是从自由文本中定位和分类命名实体到预定义的类别。（CRF、SVM和手工特征感知模型）

NER试图定位和分类在非结构化文本中提到的命名实体。本文则是提取概念。

与关键词提取不同的是，概念提取是提取全部概念，而不是顶层概念。

问题描述：

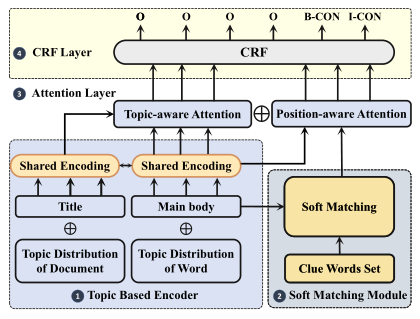
假设已经收集了线索词c={c1,c2,…,ck}。例如：c1=named with

输入：给定一个句子x={x1,x2,…,xn}和标题t={t1,…,tl}

输出：预测标签y={y1,y2,…,yn}

其中，任意yi有S-CON,B-CON,I-CON,E-CON,O这五种标签，分别表示单个词，开始，中间，结尾，和非概念词。

结构：



整体上来看：

基于topic的编码器和注意模块用于收集文档中每个上下文单词的主题和标题的相关信息。

软匹配和position注意模块用于将句子与线索词进行匹配，并模拟线索词与同一句子中其他词之间的位置和语义关系。

CRF用于对输出中的依赖项进行建模。

基于topic的编码器：

众所周知，文档和单词都可以用语义主题的混合表示。一个词在文档的不同主题中扮演着不同的重要角色。文档中的概念词是表示文档主题的关键。因此，关键概念和文件的主题分布应该相似。

使用LDA得到文档的主题分布和单词的主题分布，其中，k是超参，表示文档中的话题数量。

对标题和正文分开考虑，对标题的每个词，先找出它的词嵌入，再和文档的主题分布（不是标题的话题分布，因此只会有一个）串接。对正文的每个词，先找出它的词嵌入，再和单词的主题分布（每个单词都有）串接。将这些内容放到Bi-LSTM中即可。

软匹配模块：

用于对看不见的句子对应的线索词进行匹配，并定位线索词出现的位置。相当于以前的提取过程中的正则表达式匹配，因为正则表达式中的所有同义词和变体都必须显式指定，因此泛化能力较差。为了提高泛化能力，我们预先训练了一个软匹配模块来捕获语义相似的单词。

对一个句子x={x1,x2,…,xn}和线索词查询q。软匹配模块计算每一个xi到q的相似性分数S={s1,s2,…,sn}。通过滑动窗口来提供每个xi更多的语境信息。例如：滑动最大窗口大小为2，则对，有三种语境，它们分别是。每个语境的向量长度是一样的，因为它使用Bi-LSTM Encoder对这些语境里的词进行复合。对q也一样，生成。

相似度计算公式：

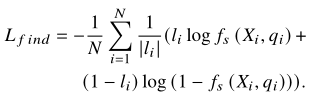




第一个公式计算了每个单词的语境的相似度。第二个公式，将语境信息合并在一起，v是每个语境所占的权重，这个是要训练的。

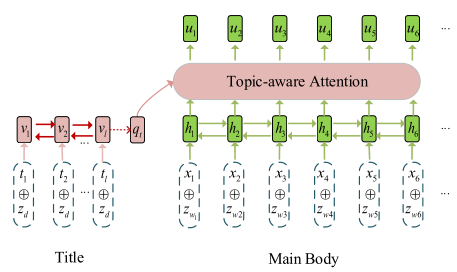
为了学习其中的参数，构造以下数据集：(s,q,l)

其中，我们在训练数据中随机选择连续单词的跨度作为q，选它出现过的句子作为s，l表示是否是在对应位置被解析（和s向量长度一致）。交叉熵损失函数。



训练后，给定一个句子x，我们可以使用它来匹配该句子与线索词，并通过比较相似度得分和阈值来返回线索词在该句子中的位置。

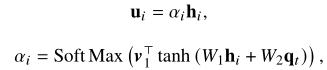
基于Topic的注意模块：



基于Topic的注意模块是从上下文中每个单词的全局信息中聚合相关信息。

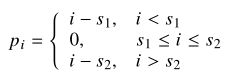
定义查询向量：（把标题的最后一个词的向量作为查询）

注意力系数和输出：

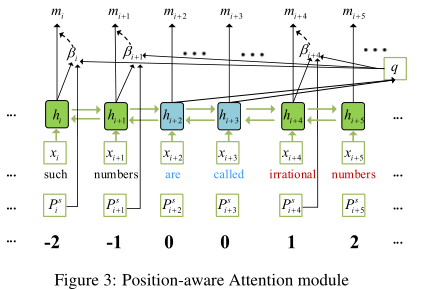


基于Position的注意模块：

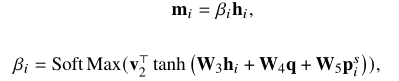
尽管现代序列模型（如LSTM网络）有门控机制来控制每个单词对下一个标记表示的相对影响，但这些机制没有明确地建模线索词和概念词在句子中的位置和语义关系。该模块旨在加强局部注意，并通过结合词位嵌入调整注意权重，从而更加强调上下文周围的线索词。



S1和s2是线索词的开始和结束为止，实际上，pi可以看成是xi到线索词的距离。



Q是所有线索词的隐向量的平均值。对每一个隐状态hi，计算注意力：



CRF条件随机场：

将position注意力和topic注意力求分配的平均值，再和原始词向量串接，输入到CRF求最终的tag（即y）。

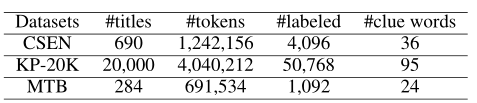
实验：

CSEN：该数据集包含690个计算机科学课程大规模开放在线课程（MOOC）的视频字幕。CSEN没有提供明确的标题。我们用每节课的第一句话作为标题，它总结了本课的内容。

KP-20K：KP20K由567830份来自不同计算机科学领域的高质量科学出版物组成。我们从KP20K中随机选择20000篇文章组成KP-20K。我们收集了与计算机领域相关的概念短语，并在每篇文章中自动注释这些概念短语。

MTB：该数据集包括小学、中学和高中的数学教科书。

我们使用70%作为训练集，10%作为验证集，20%作为测试集。



我们的线索词由三位具有相应背景知识的专家撰写。当注释者观察到固定数量的训练数据时，线索词集的开发被认为已经完成。注释者总共花了不到10个小时的时间写出所有线索词。我们将它们的交点作为最后的线索词集。

Baseline:

3个基于图形的关键词提取方法Text Rank，Topical PageRank（TPR）和Position rank。

3个监督方法：

Joint-layer RNN：结合关键词和上下文信息来执行关键词提取任务。

CopyRNN：基于RNN的生成模型，用于预测科学文本中的关键短语。这是编码器-解码器模型在短语预测任务中的首次应用。

BERT-CRF：临床数据中概念提取的基于BERT的模型。

设备：特斯拉V100 GPU和16个Intel CPU上运行所有实验。

词嵌入：word2vec初始化为300维向量。

Bi-LSTM：输出隐藏层200维。

所有W矩阵：均匀分布(−0. 1,0. 1)。

单词和文档的话题分布：LDA+话题建模工具。

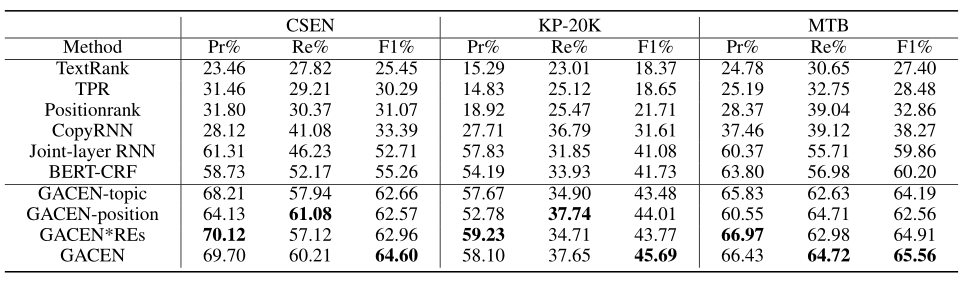
数据集CSEN、KP-20K和MTB中，话题总数分别设置为50、100和50。

最后一层的。

该模型由Adam优化，批量为10，dropout=0.1。

软匹配模块预训练：=0.75，最大窗口=3。20个周期内loss不再减少时终止训练。

结果：



* 非监督使用语料库中的统计信息，并且强烈依赖术语频率，这妨碍了他们的表现。具体而言，TPR的性能优于TextRank，可能是因为TPR还利用了主题信息。但它们的表现都不如Positionrank，Positionrank将单词出现的所有位置的信息合并到一个有偏见的PageRank中。
* 监督方法，GACEN模型在所有数据集上都取得了最好的性能，近10%的改进，这证明了使用主题、标题和线索词的有效性。

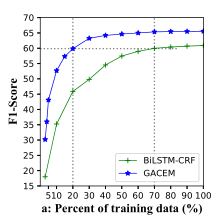
消融实验：

GACEN-TOPIC指去掉了topic模块（recall降低），GACEN-position指去掉了position模块（precision降低），RE表示用常规正则匹配代替软匹配模块（recall降低）。

GACEN能更好地平衡recall和precision。

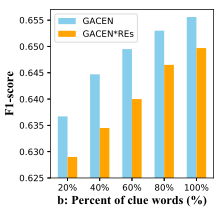
学习效率：

对于不同百分比的训练数据，可以看到，通过使用20%的训练数据，GACEN模型显示出与使用70%训练数据的基础模型相当的性能。模型性能的显著提高反映了标题和线索词在概念提取中的重要性，并证明了收集线索词所产生的轻微额外成本是合理的。



泛化性能：

随着线索词数量的增加，近义的线索词的增加，两种模型之间的性能差距减小。结果表明，软匹配模块能较好地将线索词推广到不可见句，特别是在线索词较少的情况下。



例子：

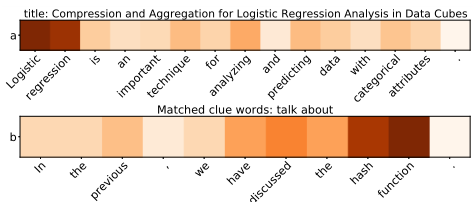


图5的示例，说明主题感知注意和位置感知注意分数有助于GACEN识别概念。

上图显示了topic注意力得分，即“逻辑回归”概念，这是一个注意力得分较高的标题短语。

下图是position注意力模块，“哈希函数”获得了较高的注意分数，并且该句子中的“discuss”一词通过软匹配与“talk about”匹配。

说明：

1.全局信息和基于线索词的模型（如GACEN）可以有效地学习。

2.模型还能提供合理的解释，这是其他神经模型（如Joint-layer RNN和CopyRNN）所缺乏的。