结合主题分割和自动文摘的演示文稿生成方法

王 鑫　李 宁　田英爱
(北京信息科技大学计算机学院，北京 100101)

摘　要　通过演示文稿传播学术成果是一种常见做法，然而手工制作演示文稿过于繁琐。本文以学术论文为原本，提出一种结合主题分割和自动文摘的演示文稿生成方法。该方法首先在论文章节结构的基础上对正文进行主题分割，构建演示文稿层次结构，再利用自动文摘抽取论文中的重要文本，基于主题生成演示文稿。实验证明，该方法生成的演示文稿不仅体现论文的行文逻辑，在 ROUGE-1、ROUGE-2、ROUGE-L 三个指标上也有所提高。

关键词　演示文稿生成　主题分割　自动文摘　ROUGE 指标

中图分类号　TP391　　**文献标志码**　A

A PRESENTATION GENERATION METHOD COMBING TOPIC SEGMENTATION AND AUTOMATIC SUMMARIZATION

Wang Xin　Li Ning　Tian Yingai
(College of Computer Science, Beijing Information Science and Technology University, *Beijing 100101, China*)

Abstract　It is a common practice to disseminate academic results through presentations, however, presentation slides are often made by hand and it is a tedious task. Taking academic papers as the source, this paper proposed a presentation slides generation method combining topic segmentation and automatic abstracting. First the main text is segmented by topic based on chapter structure, then the hierarchical structure of presentation is constructed, finally the important text in the paper is automatically abstracted, as the result, the presentation slides are generated based on the topic. Experimental results show that this method not only reflects the writing logic of the paper, but also improves the ROUGE-1,ROUGE-2 and ROUGE-L index.

Keywords　Slides generation　Topic segmentation　Automatic text abstraction　ROUGE metric

0　引　言

在学术研究中，通过演示文稿展示最新的研究成果是一种常见做法。然而演示文稿的制作耗时费力，如何根据学术论文自动生成适合的演示文稿具有较高的应用价值。

演示文稿自动生成的研究始于 20 年前，早期采用篇章分析的方法进行演示文稿内容的选择。Masao 等通过规则化方法从文档中获取主题词或短语，然后采用 GDA tagset 对文档进行半自动注释以推断句子之间的语义关系，并选择和主题相关的句子作为演示文稿的内容[1]。Shibata 等将文本中的子句和整句视为基本篇章单元，并确定彼此之间的篇章关系，再根据规则判定主题和其下的文本[2]。这些方法与文档的领域、风格无关，但需要识别特定语种（如英语，日语）的篇章结构，在其他语言中无法直接应用。

目前主流的演示文稿生成方法主要是通过自动文摘技术从学术论文中选择重要文本，根据文本所属章节或所包含的关键词将其放置到合适位置。Wan 等人利用支持向量回归(Support Vector Regression, SVR)模型判断论文中整句的重要性，根据整数线性规划算法(Integer Liner Programming, ILP)选择重要文本，然后按其所属章节进行演示文稿生成[3]。Edward 等人将演示文稿视为长篇问答任务(Long-Form Question Answering Problem, LFQA)，在用户输入幻灯片标题后，采用信息检索和自动文摘技术实现对应标题下的文本生成[4]。

尽管上述方法取得了一定的效果，但更侧重于内容的提取，对学术论文和演示文稿的结构分析不够深入。本文认为从学术论文中生成演示文稿应当满足以

收稿日期: xxxx-xx-xx。国家自然科学基金项目(61672105)。**王鑫**，硕士生，主研领域：文档信息处理。**李宁**，教授。**田英爱**，高级实验师。

下三个基本要求：

1) 简洁性：演示文稿的正文通常以简短的文本和图表来展现内容，不宜长篇大论。

2) 全面性：不同于新闻等文体，学术论文普遍遵循规范 IMRAD 结构[5]，对应的演示文稿应忠实反映论文的内容。

3) 条理性：幻灯片上的正文通常以层次结构出现。上一层常是概括性的内容，下一层进一步展开叙述。这种层次结构应该反映论文的行文逻辑。

从上述研究发现，自动文摘能够自动地从文本中提炼出反映原文中心内容的简洁连贯的短文，内容上满足了简洁性[6]。然而，在处理学术论文时，该技术无法保证全面性，生成的演示文稿中可能并未涉及原文的部分章节。此外，以往将选择的文本按其所属章节或所包含的名词短语放置到演示文稿中的做法，无法体现论文的行文逻辑。

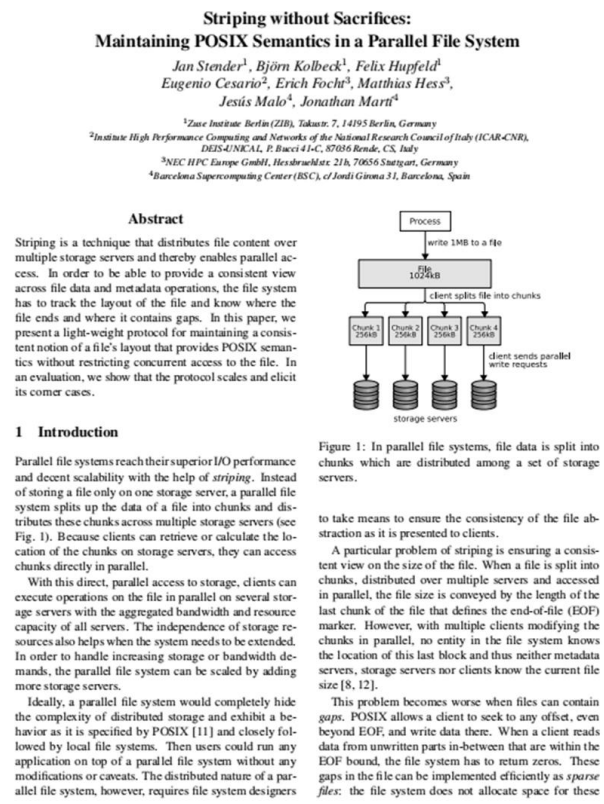
针对上述问题，本文在以往研究的基础上，提出一种主题分割与自动文摘相结合的演示文稿生成方法。本文采用与以往研究相同的数据集进行实验对比，以验证本文的方法。数据来自公开出版的计算机论文及其作者制作的演示文稿。

1 相关技术

1.1 演示文稿生成

近年来，部分学者使用机器学习方法研究演示文稿的自动生成。Wang 等人将短语视为文本单元，通过随机森林模型判断短语的重要性，并使用贪心算法进行短语选择[7]。Sefid 等人则采用神经网络模型和 ILP 进行整句重要性的判断和选择，并在此基础上结合学术论文的结构特点，引入窗口参数使得摘要句尽可能散布在论文各个章节，提高了演示文稿的全面性[8]。基于机器学习的方法提高了演示文稿的生成效果，但该方法侧重于文本的内容，对学术论文和演示文稿结构的分析不够深入。

学术论文的每个章节各有其行文逻辑，如引言的顺序通常是研究背景、研究目的和研究方法，从论文到演示文稿的转换应当保留这种行文逻辑。以往的研究在组织演示文稿时主要采用两种方法：1）根据摘要句所属章节进行划分；2）根据摘要句包含的名词短语进行划分，以名词短语作为第一层的内容，包含该名词短语的摘要句作为第二层。第一种方法会将所有文本处于第一层，导致条理不清。第二种方法则会导致同一名词短语下的文本来自各个章节，不能反映原文的行文逻辑。本文以 PS5K 数据集中的一篇论文(4493/stender.pdf)来说明第二种方法存在的问题[8]，如图 1 所示：



Striping without Sacrifices:
Maintaining POSIX Semantics in a Parallel File System

Jan Stender[1], Björn Kolbeck[1], Felix Hupfeld[1]
Eugenio Cesario[2], Erich Focht[3], Matthias Hess[3],
Jesús Malo[4], Jonathan Martí[4]

[1]Zuse Institute Berlin (ZIB), Takustr. 7, 14195 Berlin, Germany
[2]Institute High Performance Computing and Networks of the National Research Council of Italy (ICAR-CNR), DEIS-UNICAL, P. Bucci 41-C, 87036 Rende, CS, Italy
[3]NEC HPC Europe GmbH, Hessbruehlstr. 21b, 70656 Stuttgart, Germany
[4]Barcelona Supercomputing Center (BSC), c/ Jordi Girona 31, Barcelona, Spain

Abstract

Striping is a technique that distributes file content over multiple storage servers and thereby enables parallel access. In order to be able to provide a consistent view across file data and metadata operations, the file system has to track the layout of the file and know where the file ends and where it contains gaps. In this paper, we present a light-weight protocol for maintaining a consistent notion of a file's layout that provides POSIX semantics without restricting concurrent access to the file. In an evaluation, we show that the protocol scales and elicit its corner cases.

Figure 1: In parallel file systems, file data is split into chunks which are distributed among a set of storage servers.

1 Introduction

Parallel file systems reach their superior I/O performance and decent scalability with the help of *striping*. Instead of storing a file only on one storage server, a parallel file system splits up the data of a file into chunks and distributes these chunks across multiple storage servers (see Fig. 1). Because clients can retrieve or calculate the location of the chunks on storage servers, they can access chunks directly in parallel.

With this direct, parallel access to storage, clients can execute operations on the file in parallel on several storage servers with the aggregated bandwidth and resource capacity of all servers. The independence of storage resources also helps when the system needs to be extended. In order to handle increasing storage or bandwidth demands, the parallel file system can be scaled by adding more storage servers.

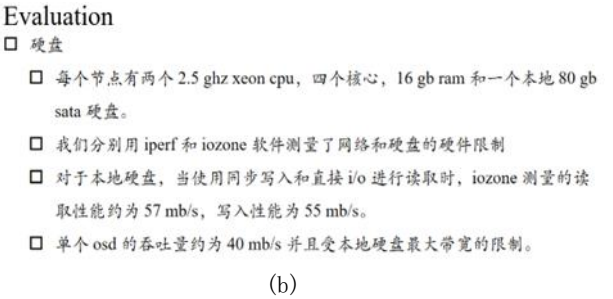
Ideally, a parallel file system would completely hide the complexity of distributed storage and exhibit a behavior as it is specified by POSIX [11] and closely followed by local file systems. Then users could run any application on top of a parallel file system without any modifications or caveats. The distributed nature of a parallel file system, however, requires file system designers to take means to ensure the consistency of the file abstraction as it is presented to clients.

A particular problem of striping is ensuring a consistent view on the size of the file. When a file is split into chunks, distributed over multiple servers and accessed in parallel, the file size is conveyed by the length of the last chunk of the file that defines the end-of-file (EOF) marker. However, with multiple clients modifying the chunks in parallel, no entity in the file system knows the location of this last block and thus neither metadata servers, storage servers nor clients know the current file size [8, 12].

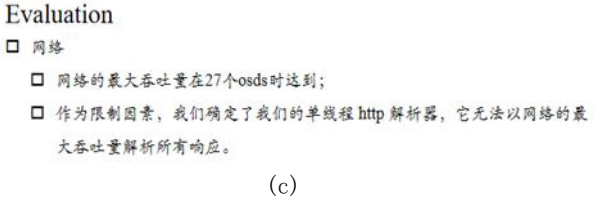
This problem becomes worse when files can contain *gaps*. POSIX allows a client to seek to any offset, even beyond EOF, and write data there. When a client reads data from unwritten parts in-between that are within the EOF bound, the file system has to return zeros. These gaps in the file can be implemented efficiently as *sparse files*: the file system does not allocate space for these

1

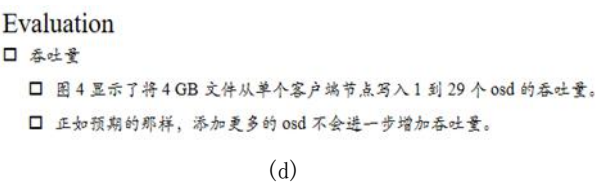
(a)



(b)



(c)



(d)

图 1 a 为部分原文，b、c、d 为 Sefid 提出的摘要模型生成的演示文稿[9](翻译为中文)

图 1 是一篇关于并行文件系统的论文(原文详见 https://github.com/oblialbum/presentation-generation)。演示文稿(b)中的 4 句话都包含了 hard disk 关键词，

从原文中可以看出，前 3 句在描述实验的硬件参数，而第 4 句则在描述实验结果。(b)尽管遵循了原文结构，但打乱了正文的行文逻辑。(c)，(d)以及(b)中的第 4 句源于同一章节且都在描述实验结果，但因为包含的关键词不同被划分到不同的层次结构中，也未遵循原文的行文逻辑。

1.2 主题分割

主题分割指根据主题相关性，识别长文本中具有独立意义的短文本单元之间的边界，使得单元之间的主题相关性最小，而单元内部的主题相关性最大。本文通过对标题下的正文进行主题分割来构建幻灯片的层次结构。目前的主题分割算法多为无监督算法，主要由三部分组成：文本单元的表示；单元间相似度衡量；边界识别策略。其中，边界识别策略是按照一定的规则判断两个相邻单元是否属于同一主题。无监督的主题分割算法可以分为以下三类：

1) 基于语言特征的方法：该方法认为特定的语言现象，比如提示短语、停顿、标记、韵律特征、指代、句法及词汇的形态同化等与主题的转变隐含某种必然联系[10]。

2) 基于词汇聚集的方法：该方法假定相同、相似或者语义相关的词汇倾向于出现在同一片段内，文本中词汇的变化能够反应主题的转变[11]。

3) 基于主题模型的方法：该方法相信合适的概率统计模型能够为主题边界的估计提供可靠依据，主要通过主题模型优化文本单元的表示来提高分割效果[12]。

上述方法各自存在不足之处，基于语言特征的方法依赖于具有明显形式化特征的文本，移植性较差；基于词汇聚集的方法未考虑到词语之间的联系，分割效果不佳；基于主题模型的方法忽略了词语本身和词语之间的语义信息。

近年来部分学者结合 Word2Vec，进一步改进了主题分割中文本单元的表示，获得了更好的分割效果，但 Word2Vec 难以获取长文本的上下文特征[13,14]。Kenton 等人提出 Bert 预训练模型，可以更好地获取长文本向量[15]。

2 一种新的演示文稿自动生成方法

为了提高演示文稿的全面性和条理性，本文在分析学术论文和演示文稿组织结构的基础上，提出了一种主题分割与自动文摘相结合的演示文稿自动生成方法。该方法首先对论文的章节进行识别，获取标题的显式逻辑层次；其次对各标题下的正文进行主题分割，构建幻灯片的层次结构，展现论文的行文逻辑；最后对主题分割的结果进行关键词识别，将关键词和对应的摘要句分别作为幻灯片正文的第一层和第二层。

该方法整体框架如图 2 所示，各部分内容在后续小节阐述。

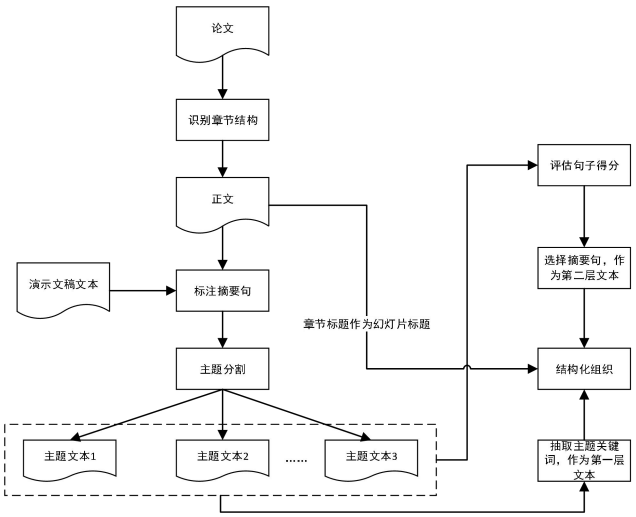


图 2 演示文稿生成方法框架图

2.1 面向自动文摘的主题分割方法

考虑到主题分割的结果将在自动文摘和演示文稿生成时应用，分割后的粒度不宜过小。因此，本文提出了一个适用于自动文摘的主题分割算法，该算法根据相邻句间的相似度判断主题边界，再按照规则合并篇幅较短的主题，算法的流程图如图 3 所示：

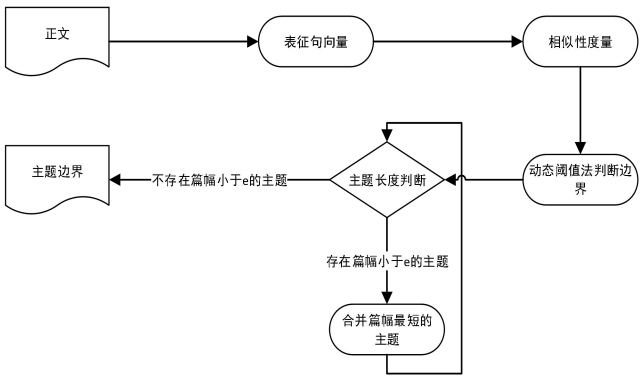


图 3 改进的主题分割流程图

其中，e 表示主题的最小篇幅，篇幅小于 e 的主题将被合并，具体的分割流程如下：

1) 对于标题下的正文，通过 Stanford Core NLP 工具进行分句。

2) 通过 Bert 模型对整句进行文本向量化，计算相邻句间的相似度，获得相似度列表 $\text{simList}=\{\text{sim}_1,\text{sim}_2,...,\text{sim}_{N-1}\}$，其中 sim_i 表示句子 i 和下一句之间的语义相似度，N 表示正文所包含的句

子数。

3) 找到(2)中 simList 中的局部最大值（极大值点，即 $\mathrm{sim}_i > \mathrm{sim}_{i-1}$ 且 $\mathrm{sim}_i > \mathrm{sim}_{i+1}$ ）和局部最小值（极小值点，即 $\mathrm{sim}_i < \mathrm{sim}_{i-1}$ 且 $\mathrm{sim}_i < \mathrm{sim}_{i+1}$ ），将局部最小值点对应的整句作为候选主题边界。根据公式(1)计算每个候选边界的深度得分：

$$depthScore_i = \frac{left_{sim}^{\max} + right_{sim}^{\max}}{2*sim_i} - 1 \quad (1)$$

其中，sim_i 为 simList 中的局部极小值点，$left_{sim}^{\max}$ 和 $right_{sim}^{\max}$ 分别代表该点的前驱和后继中距离该点最近的局部最大值点，$depthScore_i$ 表示当前候选边界的深度得分。设定阈值 θ，若 $depthScore_i > \theta$,则认为该点对应的整句为主题边界。

4) 判断是否存在篇幅小于 e 的主题，如果存在则将篇幅最短的主题与其左右主题中篇幅较短的主题合并，重复此本步骤，反之则结束。

具体的分割算法流程如下：

Algorithm：TopicSegment
Input：*text* as the body of paper, *t* as the threshold which aid to judge topic boundaries
Output：*Topic Boundaries,* a point list where adjacent point determines a topic area

```
1  Function TopicSegment(text, t)
2      Let sentence-list ← be the result of Stanford-Core-NLP-Parser of text ;// split text into sentence list
3      Let similarity-list ←be the list of cosine similarity of adjacent sentences
4      Let min-list be the local minima point list representing candidate Topic Boundaries;
5      Let max-list be the local maximum point list;
6      min-list ← null;
7      max-list ← null;
8      For each similarity sim_i in similarity-list
9          If sim_i < sim_{i+1} and sim_i < sim_{i-1} Then
10             Add i into min-list;
11         End If
12         If sim_i > sim_{i+1} and sim_i > sim_{i-1} Then
13             Add i into max-list;
14         End If
15     End For
16     For each point i in min-list
17      /*Calculate the score according to formula 1*/
18         depth-score ← getDepthScore(i, similarity-list, min-list, max-list )
19         If depth-score < t then
20             Remove i from min-list;
21         End if
22     End for
23     For each topic in min-list
24         If the length of topic is less than 5 sentences Then
25             Merge shortest topic with adjacent topic
26     End for
27     Output min-list;
28 End Function
```

尽管该算法在合并过程中会降低主题划分的召回率，但对于自动文摘而言，将一个篇幅较短的主题合并到另一个主题中，对摘要的结果并不会产生很大的影响。本文根据实验设定合并的篇幅为 e =5 句。

2.2 面向文本选择的自动摘要方法

分割后的主题和对应的摘要将用来训练模型。为了提高演示文稿的全面性，模型生成的摘要应当覆盖论文的各个主题，也就需要在标注阶段尽可能地将摘要句散布在论文的各个主题中。Sefid 等人所采用的 BertSum 摘要模型采用最大化 ROUGE-2 分数的贪心算法进行摘要句的标注[9,16]，无法满足这一要求，本文选择文献[8]中提出的方法进行标注。该方法同样使用贪心算法，不同之处在于选择的范围从全文变为包含连续 N 个句子的窗口，每个窗口互不重叠。每次选择最大化 ROUGE-1 分数的句子，不断迭代直到达到预期的摘要比例，或无法继续提高 ROUGE-1 分数。

BertSum 是一个基于 Bert 的抽取式摘要模型。该模型对输入文本的所有句子前添加[CLS]标识并用区间段嵌入来区分多个句子，使用 Bert 获取句向量，通过多个 Transformer 层来预测每个句子的得分，最后根据句子得分选择摘要。

原始 Bert 模型的最大处理长度为 512 token，BertSum 通过添加更多的位置编码将其扩展，但需要消耗较多的 GPU 内存。本文根据论文的章节结构和 3.1 中的主题分割算法，对原始论文进行分割，降低了 GPU 的显存消耗。本文按照 Sefid 等人的方法将 PS5K 数据集划分为训练集，验证集，测试集，分别包含 4500、250 和 250 论文-演示文稿对[9]。数据集中的每篇文档都会使用主题分割算法进行分割，分割后的主题和对应的摘要会作为新的数据集用来训练 BertSum 模型。BertSum 的大部分参数与原文相同，具体如下：

使用 Bert 作为 encoder，双层 transformer 作为 decoder，dropout 为 0.2，学习率 $lr = 2e^{-3} \times \min\left(step^{-0.5}, step \times warmup^{-1.5}\right)$，最大输入文本长度为 1024 token，隐藏层状态大小为 768，采用 Adam 优化器（$\beta_1 = 0.9, \beta_2 = 0.999$），输出层为 sigmoid 函数。模型训练目标为最小化预测标签和实际标签的二元交叉熵损失。

2.3 演示文稿生成

主题分割算法的文本向量化工作通过

Sentence-Bert 完成[17]。Sentence-Bert 是 Bert 模型的变种，基于孪生网络结构构建句向量并保留了 Bert 的准确性。各标题下的正文经过主题分割后形成一个个主题，每个主题所在章节的标题作为幻灯片的标题，主题所包含的关键词作为第一层文本。如果该关键词在之前的主题中已使用，则选择下一个关键词。对于上一级标题和下一级标题之间的正文，如本文第 2 章与 2.1 节，采用和上述相同的做法。

BertSum 的摘要比例为 20%，batch size 为 10，训练 1500000 步，每两步进行 1 次梯度累加，每 30000 步进行一次保存，其他参数与 3.2 小节中相同。当相邻两次检查点之间的损失值变化小于 0.1 时，认为模型已经收敛。模型收敛后，选择在验证集上表现最好的 3 个检查点进行测试，在测试集上表现最好的模型将被选中。对一篇新论文进行摘要时，模型先预测每个主题中每个句子的分数，再贪心地选择每个主题中的摘要句作为主题的第二层文本。特别的，如果主题只包含一句摘要句，则舍弃关键字并将摘要句视为第一层文本。

由于幻灯片篇幅有限和突出重点的要求，本文设置每张幻灯片上最多包含 4 句摘要，超过则分解成两张幻灯片。

3 实验结果与分析

目前，公开的论文-演示文稿数据集仅有两个，分别是 PS5K 数据集和 SciDuet 数据集[4]。本文使用 Sefid 等人构建的 PS5K 数据集，该数据集包含 5000 对发表于 2013-2019 年的论文-演示文稿数据，主要在计算机语言，系统和系统安全领域[8]。在该数据集中，平均每篇演示文稿对应 35 张幻灯片，每张幻灯片包含 8 行文本。

由于主题并没有一个确切的划分标准，分割的结果难以直接量化评价，因此，本文通过评价摘要模型的表现来间接判断主题分割的性能。

本文使用 ROUGE 作为摘要模型的评价指标。ROUGE 通过比对自动摘要和原始摘要的 n-gram 重叠情况来量化生成的摘要质量，是目前应用最为广泛的评测方法。本文分别计算 ROUGE-1，ROUGE-2，ROUGE-L 的值来评价结合主题分割的摘要模型性能，实验结果如表 1 和表 2 所示。

通过表 1 的数据可以看出，在主题分割的阈值 $\theta=1$ 时，模型取得了最佳结果。此外，本文测试了在 $\theta=1$ 时原主题分割模型的表现（表 1 第 5 行），结果表明，改进后的主题分割算法在文摘任务上表现更好。表 2 则展示了不同模型的摘要结果。

表 1 不同阈值下的实验结果

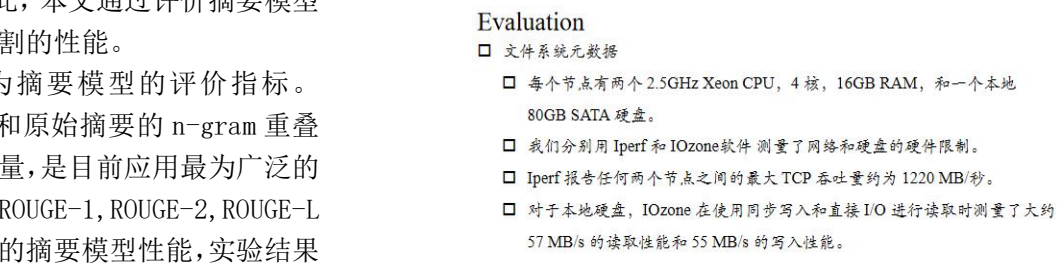
阈值	ROUGE-1	ROUGE-2	ROUGE-L
0.5	55.338	15.70	48.69
0.7	55.88	15.92	49.25
1.0	**58.49**	**17.39**	**51.39**
1.0（不合并）	55.41	15.60	48.82
1.3	55.45	15.67	48.86

表 2 本文方法与其他摘要方法的结果对比

模型	ROUGE-1	ROUGE-2	ROUGE-L
Lead20%	37.68	6.62	15.90
TextRank[19]	38.87	9.28	19.75
SummaRuNNer[20]	45.04	11.67	23.03
SciBertSum[9]	52.34	15.06	36.67
本文方法	**58.49**	**17.39**	**51.39**

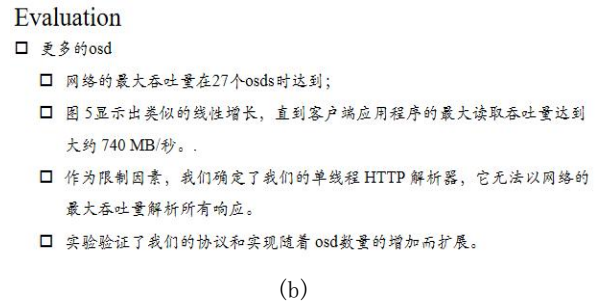
Lead20%选择原文前 20%文本作为摘要，由于论文包含摘要和引言章节，该结果具有一定的可比性。TextRank 是一种基于图模型的无监督摘要方法，由 Google 的 PageRank 算法衍生而来[18]。SummaRuNNer 是一种基于循环神经网络（RNN）的序列分类模型，相较于常规方法，在自动文摘领域的表现更好[19]。SciBertSum 是 Sefid 等人提出的 BertSum 的变种模型，主要采用分块编码解决学术论文过长时 BertSum 占用显存过大的问题[9]。从表 4 可以看出，本文提出的主题分割+BertSum 模型相比于 SciBertSum 模型在 ROUGE-1，ROUGE-2 和 ROUGE-L 指标上均有所提高。我们认为 SciBertSum 表现弱于本文模型的原因主要有以下两点：1)根据 GPU 内存大小对论文进行分块编码，可能将一个完整的章节或主题划分到不同的块中；2)划分后的块仍属于长文本，模型存在信息损失较多的问题。

一个本文模型生成的演示文稿示例如下(仍使用图 1 中的论文)：


Evaluation

- 文件系统元数据
  - 每个节点有两个 2.5GHz Xeon CPU，4 核，16GB RAM，和一个本地 80GB SATA 硬盘。
  - 我们分别用 Iperf 和 IOzone软件 测量了网络和硬盘的硬件限制。
  - Iperf 报告任何两个节点之间的最大 TCP 吞吐量约为 1220 MB/秒。
  - 对于本地硬盘，IOzone 在使用同步写入和直接 I/O 进行读取时测量了大约 57 MB/s 的读取性能和 55 MB/s 的写入性能。

(a)



(b)

图4 本文模型生成的演示文稿示例(翻译为中文)

经过主题分割后，原文的Evaluation章节被划分为两个主题。第一个主题描述了实验目的和实验设置，第二个主题则主要描述实验结果与分析。由上图可以看出，本文的方法保留原文章节结构的基础上，对各层标题的文本内容也进行了条理化，使产生的演示文稿既能遵循论文作者的表达逻辑，又能体现演示文稿提纲挈领的特点。

4 结束语

本文根据学术论文和演示文稿的特点，提出了一种主题分割和自动文摘结合的演示文稿生成方法。该方法在论文章节结构的基础上对各标题下的正文进行主题分割，构建演示文稿的层次结构。实验结果表明，本文方法既能体现出论文的行文逻辑，又能提高摘要句抽取的准确性。

然而，本文方法仍存在不足之处，后续工作会从以下两个方面进行：首先，一个整句所表达的内容并非都可以作为摘要内容，这会导致生成的演示文稿冗长，未来我们将考虑使用汉语的基本篇章单元作为更细粒度的抽取对象[20]；其次，根据演讲的不同时长，演示文稿的篇幅并不固定，在内容的取舍上需要综合考虑不同章节和主题的权重，这将成为我们下一步的研究重点。

参 考 文 献

[1] Masao U, Kôiti H. Automatic slide presentation from semantically annotated documents[C]//Proceedings of the Workshop on Coreference and its Applications. 1999: 25-30.

[2] Shibata T, Kurohashi S. Automatic Slide Generation Based on Discourse Structure Analysis[J]. Journal of Natural Language Processing, 2005, 13(3):754-766.

[3] Hu Y, Wan X. PPSGen: Learning-based presentation slides generation for academic papers[J]. IEEE transactions on knowledge and data engineering, 2014, 27(4): 1085-1097.

[4] Sun E, Hou Y, Wang D, et al. D2S: Document-to-Slide Generation Via Query-Based Text Summarization[C]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021: 1405-1418.

[5] Teufel S, Moens M. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status [J]. Computational Linguistics, 2002, 28(4): 409-445.

[6] 李金鹏,张闯,陈小军,等.自动文本摘要研究综述[J].计算机研究与发展,2021,58(01):1-21.

[7] Wang S, Wan X, Du S. Phrase-Based Presentation Slides Generation for Academic Papers[C]// Proc of the AAAI Conf on Artificial Intelligence.2017,196-202.

[8] Sefid A, Mitra P, Wu J, et al. Extractive Research Slide Generation Using Windowed Labeling Ranking[C]// Proceedings of the Second Workshop on Scholarly Document Processing.2021,91-96.

[9] Sefid A, Mitra P, Giles L. SlideGen: an abstractive section-based slide generator for scholarly documents[C]//Proceedings of the 21st ACM Symposium on Document Engineering. 2021: 1-4.

[10] Levow G A. Prosody-based topic segmentation for mandarin broadcast news[C]//Proceedings of HLT-NAACL 2004: Short Papers. 2004: 137-140.

[11] Kehagias A, Nicolaou A, Petridis V, et al. Text segmentation by product partition models and dynamic programming[J]. Mathematical and Computer Modelling, 2004, 39(2-3): 209-217.

[12] 李天彩,王波,席耀一,等.基于分层狄利克雷过程模型的文本分割[J].数据采集与处理,2017,32(02):408-416.

[13] 肖梦雅.基于改进的主题分割模型在教师华语文本分析中的应用研究[D].华中师范大学,2020.

[14] 李宇雯.基于中文长文本的自动文本摘要系统研究[D].上海交通大学,2020.

[15] Kention J DMWC, Toutanova L K.BERT: pre-training of deep bidirectional transformers for language

understanding[C]//Proceedings of NAACL-HLT,NAACL, 2019, 4171-4186.

[16] Liu Y, L apata M. Text Summarization with Pretrained Encoders[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). 2019: 3730-3740.

[17] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019.

[18] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the 2004 conference on empirical methods in natural language processing. 2004: 404-411.

[19] Nallapati R, Zhai F, Zhou B. Summarunner: A recurrent neural network-based sequence model for extractive summarization of documents[C]//Thirty-first AAAI conference on artificial intelligence. 2017.

[20] 葛海柱,孔芳,周国栋.基于主述位理论的汉语基本篇章单元识别[J].中文信息学报,2019,33(8):20-27.

5 联系方式

负责人：王鑫，15601319577，15601319577，15601319577@163.com.

作者：王鑫，硕士生，文档信息处理，320829199610212230，15601319577，北京信息科技大学 计算机学院，北京北四环中路 35 号，100101，15601319577@163.com.

（通信）作者：李宁，教授/博士，文档信息处理，110105196405185438，13501218765，北京信息科技大学 计算机学院，北京北四环中路 35 号，100101，ningli.ok@163.com.

作者：田英爱，副教授/硕士，文档格式与标准&大数据分析及处理，220222197510060241，13681560012，北京信息科技大学 计算机学院，北京北四环中路 35 号，100101，tianyingai@bistu.edu.cn.