arXiv:2301.01935v1 [cs.CL] 5 Jan 2023

# 聚焦局部上下文的主题分割模型

**Jeonghwan Lee、Jiyeong Han、Sunghoon Baek、Min Song***

延世大学 jeonghwan.ai@yonsei.ac.kr、jiyoung181@yonsei.ac.kr、sunghoon.baek@yonsei.ac.kr、min.song@yonsei.ac.kr

**摘要**

主题分割对于理解科学文档至关重要，因为它不仅能提升可读性，还能通过创建恰当的章节或段落，促进信息检索和问答等下游任务。在主题分割任务中，主题连贯性对于预测分割边界至关重要。现有大多数模型都试图利用尽可能多的上下文来提取有用的主题相关信息。然而，更多的上下文并不总能带来理想的结果，因为即使补充了更多句子，句子之间的局部上下文也可能变得不连贯。为缓解这一问题，我们提出了孪生句子嵌入层，分别处理两个输入句子，从而获取适量的信息，避免受到过多信息的干扰。此外，我们采用了多任务学习技术，包括同主题预测（STP）、主题分类（TC）和下一句预测（NSP）。将这三个分类层以多任务方式结合起来，可以相互弥补彼此的局限，从而提升三项任务的表现。我们试验了这三个分类层的不同组合，并报告了每一层如何影响同一组合中的其他层，以及整体分割表现。我们提出的模型在 Wiki-Section 数据集上取得了当前最先进的结果。

## 引言

如今，我们可以轻松访问大量科学文档，例如 PubMed 和维基百科。许多研究人员正在研究如何在信息检索（IR）、问答（QA）和搜索引擎等领域有效利用这些文档。然而，无法将以往的信息检索模型（或问答模型）直接应用于这些文档，因为大多数模型都假设输入最多为一个段落，而这些文档却由多个段落组成。此外，提取每份文档中的关键信息并不一定需要使用整份文档。例如，要搜索与某个感兴趣主题相似的论文，我们可以通过计算来简化问题

计算每个文档各部分之间的余弦相似度，而非对全文进行计算，以节省资源。

*通讯作者。版权所有 © 2023，人工智能促进协会 (www.aaai.org)。保留所有权利。

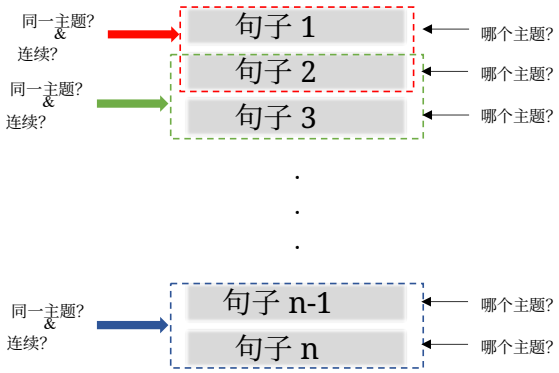


图1：一个大小为1的窗口滑过整个句子序列，为每个句子预测主题。同时，模型还会判断两个句子是否属于同一主题，以及它们是否相邻。

这些都是主题分割的应用场景。主题分割会根据各个片段的主题连贯性，将文档划分为多个片段。按照主题对文档进行合理划分，有助于提升可读性，让读者更容易在文档中找到所需信息。最重要的是，这还能促进信息检索和问答等下游任务。

尽管大多数现有的主题分割模型在划分文档时都会考虑主题连贯性，但它们并不会对每个句子进行主题标签分类，即使这些主题标签有助于推断主题连贯性。最重要的是，受神经文本分割模型（Koshorek 等人，2018年）的启发，这些模型以文本块作为输入，这可能会妨碍它们理解输入文本的局部上下文（Xing 等人，2020年）。

我们尝试采用孪生网络对两个输入句子分别进行编码，并将其输入多任务学习算法，以解决上述问题。该算法包括主题分类和其他辅助任务。首先，为了分别处理两个输入句子，我们使用 Sentence Transformer（Reimers

和 Gurevych，2019年）的句子嵌入构建孪生网络模型。这种方法使我们的模型能够保留两个输入句子之间的局部上下文，而不会受到过多信息的干扰。

我们遵循 Aumiller 等人（2021）的做法，将主题分割视为两个输入句子之间的同主题预测（STP）。然而，由于我们的模型每次只处理一个句子，以保留其独特信息，因此无法观察句子之间的上下文信息。为缓解这一问题，我们添加了两个辅助任务来捕捉局部上下文信息。其中一个是主题分类（TC），它通过主题分类层预测输入句子的确切主题，以提供详细的主题信息来辅助 STP。另一个是下一句预测（NSP）层（Devlin 等人，2019年），用于帮助模型理解连续句子之间的关系。图1简要展示了我们模型的工作方式。

总而言之，我们的模型通过孪生句子嵌入层分别处理两个输入句子，同时保留输入句子的局部上下文。此外，我们表明，以多任务方式连接利用相同输入句子提取不同句子特征的任务，能够提升主题分割性能。因此，我们的模型在使用 WikiSection 数据集进行主题分割任务时达到了最先进的水平。

## 相关工作

### 主题分割

Koshorek 等人（2018）将主题分割任务建模为一个有监督的神经网络模型。模型接收由多个句子组成的文本块，并预测每个句子是否应作为分割点。

Badjatiya 等人（2018）引入了 $k$大小的左侧和右侧辅助句子，其中相邻的 $k$ 个句子为输入句子注入上下文。然而，Xing 等人（2020）指出，在主题分割任务中，“局部上下文”比“全局上下文”更重要，这意味着过多的上下文可能会降低性能。由于模型对局部上下文的理解能力减弱，来自不同句子的信息可能会妨碍对单个句子的标签进行预测。

Arnold 等人（2019）提出了 SECTOR，其架构包含一个主题嵌入层。该主题嵌入层用于主题分类，主题分类的结果随后用于分割。

Aumiller 等人（2021）将主题分割视为两个输入段落之间的同主题预测（STP）。STP 用于判断两个输入段落是否涉及同一主题。他们还尝试了多种采样方法，我们采用其中的连续采样。连续采样的详情将在第 节中说明。

### 句子嵌入

句子嵌入是一种捕捉句子中词语之间语义关系的方法（Conneau 等人，2017年）。句子嵌入的质量至关重要，尤其是在主题分割任务中，因为该任务不可避免地必须从长句和短句中尽可能多地捕捉信息。Koshorek 等人（2018年）利用双向长短期记忆网络生成句子嵌入：从 Word2Vec 中提取词嵌入向量，将句子中的每个词依次作为输入馈入双向长短期记忆网络层，并对长短期记忆网络的输出进行最大池化，以得到最终的序列表示。

BERT 问世后，Reimers 和 Gurevych（2019）提出了专门用于生成句子嵌入的句子 BERT。句子 BERT 是一种经过微调的 BERT 版本，使用自然语言推理（NLI）和语义文本相似度（STS）任务进行训练。为有效处理输入句子，作者采用了孪生网络，分别对每个句子进行编码，再将编码后的句子拼接起来，输入分类层。因此，句子 BERT 处理长句子的能力更强，因为该模型能够理解这些句子的高层语境。

作为句子 BERT 的另一条分支，SimCSE 由 Gao、Yao 和 Chen（2021）提出。作者将对比学习应用于句子嵌入的生成。他们尝试了无监督方法，并表明随机失活可以作为数据增强，从而防止表示坍塌。他们还通过实证和理论证明，对比学习目标适合正则化语言模型嵌入的各向异性空间，使其分布更加均匀；因此，在有监督设置下，它能更好地对齐正样本对。

Lukasik 等人（2020）提出了跨片段 BERT。他们使用预训练 BERT，通过 [SEP] 标记分隔左右语境，并将 WordPiece 标记序列编码为句子表示。Aumiller 等人（2021）使用句子 BERT 作为句子编码器，该编码器以理解高层语境的强大能力而著称。

## 提出的方法

### 架构

我们的模型遵循文本分段模型的典型架构：先经过句子嵌入层，再由片段分类器进行分类；在我们的模型中，片段分类器被 SameTopic 预测层取代。

不过，我们的模型接收两个输入句子。为了分别处理这两个句子，我们将句子嵌入层构建为孪生网络形式，使模型能够获得适量的信息来预测标签。随后，编码后的句子依次输入主题分类层。每个句子经过该层后，模型便能获取与该句子相关的主题信息。此外，我们采用 NSP 层来捕捉两个句子之间的语义关系。最后，STP 层预测这两个句子是否属于同一主题。

我们有 $k$ 篇文档 $D_1$，$...D_k$ 由 $D$ $n$ num 个句子 $s_1$，..... $s_n$组成，这些句子按顺序配对；$[(s_1,\ s_2),\ (s_2,\ s_3)$ .....$(s_{n-2},\ s_{n-1}),\ (s_{n-1},\ s_n)]$。每个 $s_i(i \leq n)$ 都会被赋予一个主题标签 $t_i$，该标签描述第 $i$个句子的主题。

| 模型 | STP损失 | TC损失 | NSP损失 |
|---|---|---|---|
| STP+主题分类 | 4 | 1 | - |
| STP+NSP | 1 | - | 1 |
| STP+主题分类+NSP | 4 | 1 | 4 |

表1：多任务学习情况下各层指定的损失权重

句子嵌入层对每个输入句子 $s_i$ 和 $s_{i+1}$ 进行编码，编码后的句子分别表示为 $u$ 和 $v$。图 2 展示了我们模型的整体结构。

**来自 Sentence Transformer 的孪生句子嵌入层** 我们提出了孪生句子嵌入层。在我们的模型中，Sentence Transformer 在输入层分别独立地对两个输入句子中的每个句子进行编码。随后，将编码后的句子拼接起来，再输入 STP 层。此方法旨在保留每个句子的独特信息，同时获取两个句子之间的局部上下文。

**多任务学习** 我们的模型共有三个分类层，并以多任务方式对它们进行训练。

**主题分类层：** 主题分类层旨在捕捉句子的精确信息。主题分类是一种多类分类任务，它从英语城市数据集的30个标签和英语疾病数据集的27个标签中预测输入句子的主题（Arnold 等人，2019）。该层依次接收 $u$ 和 $v$，并分别预测各自的主题标签 $t_i$ 和 $t_{i+1}$。

**NSP 层：** NSP 层接收 $u; v; |u - v|$ (Reimers 和 Gurevych，2019年)作为输入，并预测 NSP标签。该层旨在弥补 STP层的局限性：STP层只能判断两个输入句子是否属于同一主题，却无法确定这两个句子是否实际连续。通过加入 NSP，模型可以捕捉两个句子之间的语义关系，从而判断这些句子是否连续。NSP 层必须与连续采样配合使用，具体将在下文说明。

**STP层：** STP层再次接收 $u; v; |u - v|$ 作为输入，最终预测分割标签，用于在两个句子属于不同主题的位置绘制分割点。

**多任务学习：** 以多任务方式组合这三个层，可以相互弥补各自的局限性。由于 STP 基于二元分类，其任务比基于多类分类的主题分类简单得多。然而，STP 无法捕捉输入句子的确切主题标签，而主题分类可以提供这一信息，帮助 STP层确定分割边界。NSP 的作用已在上文说明。

**损失权重：** 由于各层的损失各不相同，因此需要调整各项损失的权重，以提升模型性能。我们通过大量人工实验确定各项损失的权重，并使用加权和计算各分类层的三项损失。表 1 总结了各项损失的权重。

| | 英语城市 | 英语疾病 |
|---|---|---|
| **Docs** | 19,539 | 3,590 |
| **主题** | 30 | 27 |

表 2：**英语城市**和**英语疾病**的文档和主题数量

**连续采样**

为了提高模型的鲁棒性，我们采用连续采样（Aumiller 等人，2021），向数据集中添加负样本。在连续采样中，所有样本都来自同一篇文档。

我们有一个文档 $D_a$ 和一个句子 $s_i^{t_i} \in D_a$，其中上标 $t_i$ 表示 $s_i$的主题标签。我们选取一个正样本和两个负样本。正样本 $s_{i+1}^{t_{i+1}=t_i} \in D_a$ 与 $s_i$ 相邻，第一个负样本 $s_{k\neq i+1}^{t_k=t_i}$ 与 $s_i$属于同一主题，但与 $s_i$ 不相邻。最后，第二个负样本 $s_l^{t_l\neq t_i}$属于不同主题，因此自然可认为与 $s_i$ 不相邻。

## 实验

**数据集**

我们使用 WikiSection（Arnold 等人，2019）训练和评估模型。WikiSection 涵盖两个不同的领域：城市和疾病。每个领域分别包含 19,539 和 3,590 篇文档，每篇文档都包含不同的主题。两个领域分别共有 30 个和 27 个主题。数据集划分为 70% 的训练集、10% 的验证集和 20% 的测试集。表 2 给出了数据集的统计信息。

**实验设置**

我们使用 nltk 句子分词器[1] 将文档拆分为句子单元，并且仅在训练集上应用连续采样。表3给出了应用句子拆分和连续采样后的数据统计信息。我们使用 Sentence-Transformers[2]中的 all-MiniLM-L12-v2 实现句子编码器。我们将最大训练轮次设为 14，但仅在验证集 $P_k$ 得分最佳时保存模型。批次大小为 48，学习率为 $1e{-}6$，并使用默认参数设置应用 LinearLR 调度器。

**指标**

为全面评估，我们使用 $P_k$、$WindowDiff$ 和微平均 F1 分数来评估模型。我们使用 $P_k$ 分数比较我们的模型与所有其他基线模型，并使用 $WindowDiff$ 评估我们实现的模型和跨片段 BERT。F1 分数用于对我们的模型进行消融研究。

[1]NLTK::自然语言工具包[2]预训练模型 — Sentence-Transformers 文档 (sbert.net)

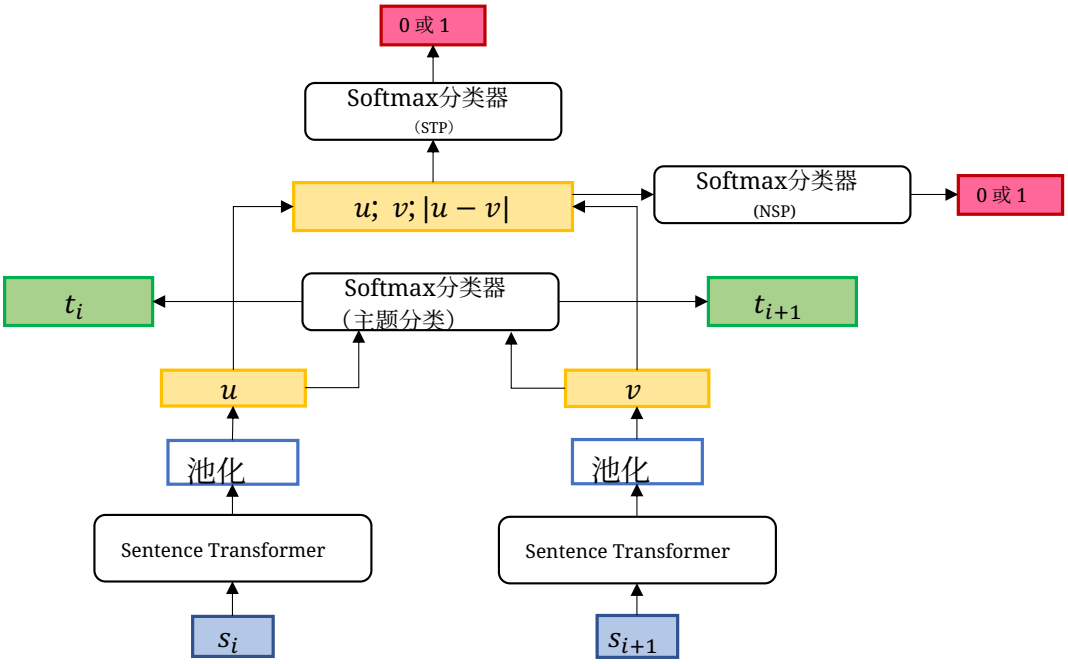


图 2：我们模型的概览。两个被视为连续的输入句子分别输入 Sentence Transformer，并对每个输入句子进行编码。每个经过最大池化编码的句子分别用 $u$ 和 $v$ 表示，并输入主题分类层。在输入 NSP 层和 STP 层之前，我们将特征拼接为 $u;v;|u-v|$。利用$u;\ v;|u-v|$，NSP 层预测这两个句子是否连续，而 STP 层最终判断它们是否属于同一主题。

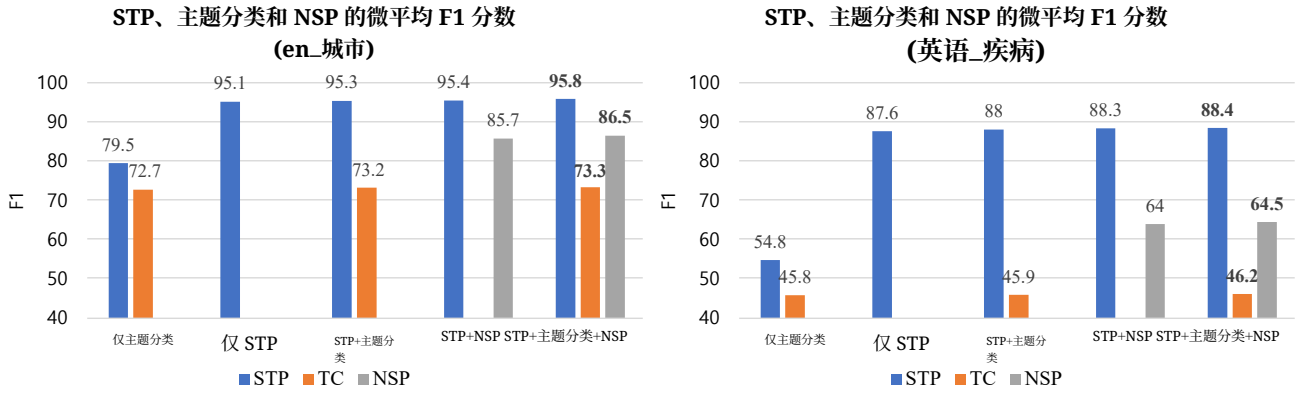


图 3：不同任务层组合下的 F1 分数图。在仅 TC 模型中，如果对每个句子的主题分类结果指向同一主题，则 STP 输出为 1，否则为 0。

| | 英语城市 | 英语疾病 |
|---|---|---|
| **训练集** | 1,690,103 | 336,459 |
| **验证集** | 85,072 | 16,285 |
| **Test** | 168,924 | 31,110 |

表3：应用 nltk 句子分词器和连续采样后的行数。连续采样仅应用于训练集。

$P_k$ $P_k$（Beeferman、Berger 和 Lafferty（1999））表示分段模型执行错误分割的概率。当大小为 k 的滑动窗口沿句子移动时，根据窗口两端是否位于同一片段，确定其状态为 0 或 1。$P_k$ 通过统计真实标注与预测值之间不匹配的情况计算得出。与许多以往研究一样，我们将窗口大小 k 设为真实标注的平均分段长度的一半。

| 数据集 | 英语城市 | | 英语疾病 | |
|---|---|---|---|---|
| 指标 | $P_k$ | $WinDiff$ | $P_k$ | $WinDiff$ |
| SEC>T+嵌入 | 15.5 | - | 26.3 | - |
| Transformer$^2_{BERT}$ | 8.2 | - | 18.8 | - |
| 双向长短期记忆网络 + BERT | 9.3 | - | 28.0 | - |
| 跨片段 BERT 和上下文 = 2 | 15.4 | 27.4 | 33.9 | 59.0 |
| 跨片段 BERT 和上下文 = 4 | 18.3 | 32.2 | 34.8 | 60.3 |
| 跨片段 BERT 和上下文 = 6 | 45.1 | 50.0 | 34.0 | 57.1 |
| 仅主题分类 | 15.0 | 17.8 | 41.5 | 45.4 |
| 仅 STP | 5.1 | 5.8 | 14.8 | 15.8 |
| STP + 主题分类 | 5.0 | 5.7 | 14.0 | 15.0 |
| STP + NSP | 4.9 | 5.6 | 14.1 | 15.1 |
| **STP + 主题分类 + NSP** | **4.6** | **5.2** | **13.7** | **14.7** |

表 4：基线模型和我们的模型在测试 $P_k$和 $WindowDiff$方面的得分。请注意， $WinDiff$指标仅用于我们的模型和跨片段 BERT 模型。我们按照跨片段 BERT 的官方代码自行重新实现了该模型。

$WindowDiff$ $WindowDiff$（Pevzner 和 Hearst，2002）是对 $P_k$ 的改进指标，因为它减轻了假阴性惩罚和片段大小分布的影响。

与 $P_k$ 类似， $WindowDiff$得分也使用滑动窗口，将真实标注与预测值进行比较。不过，该指标还考虑了边界数量。当模型在 $P_k$和 $WindowDiff$两项上的得分都较低时，其结果更接近真实标注。

### 基线模型

我们将模型与具有竞争力的神经文本分段基线进行比较：1) SEC>T+emb（Arnold 等人，2019），2) Transformer$^2_{BERT}$ （Lo 等人，2021），这是一个基于两个 Transformer 的框架，其中一个是用于编码句子的预训练 Transformer，另一个是用于分段的 Transformer；3) Bi-LSTM + BERT（Xing 等人，2020），它基于层次注意力 Bi-LSTM 网络；以及 4) 跨片段 BERT（Lukasik 等人，2020），它使用 BERT 编码器同时处理左右上下文。

我们采用 Arnold 等人（2019）的 SEC>T+emb，以及 Xing 等人（2020）的 Transformer$^2_{BERT}$ 和 Bi-LSTM + BERT 的结果。我们按照其官方代码自行实现了跨片段 BERT，并使用不同大小的上下文。

### 结果与分析

我们在图3和表 4中报告了评估结果。图3总结了各分类层组合对其 F1 分数的影响。表 4分别展示了我们的模型与其他基线模型在 $P_k$和 $WindowDiff$中的性能对比。除仅 TC 模型外，我们提出的模型均大幅优于所有基线模型。

**多任务学习的效果** 图3展示了上述任务组合所得到的 F1 分数。可以看出，多任务学习能够有效提升性能，这不仅适用于负责分割的 STP，也适用于主题分类和 NSP 的性能。这据信，这是因为，正如我们在该部分指出的，各层通过从相同的输入句子中提取不同特征来弥补彼此的局限性，从而帮助理解语义信息。

**仅 STP 与仅 TC 对比** 为了验证 STP层的有效性，我们还对仅 TC 模型进行了实验。该模型与 Sectorin 接近，分割仅使用主题标签完成。

$P_k$和 $WindowDiff$仅 TC 模型的结果远高于仅 STP 模型。主题分类的分类性能不佳直接导致了这一现象。图3表明，仅 TC 模型的 F1 分数显著低于仅 STP 模型。由于主题分类层基于多类分类，因此比仅 STP 模型的二元分类更难。

**STP 与 NSP 的比较** 尽管 STP 和 NSP 具有相同的架构，但在两个数据集中，STP 的 F1 分数始终高于 NSP。我们认为，这种差异源于 STP 和 NSP 所关注的信息不同。STP-only 只判断两个句子是否属于同一主题，因此只关注两个句子之间的主题差异。换言之，由于任务本身的性质，STP 不考虑两个输入句子之间的关系。然而，在 NSP 任务中，由于采用连续采样，即使两个句子属于同一主题，它们也可能并不连续，因此该层面临困难。因此，NSP 不仅要判断两个句子之间的语义关系，还要考虑主题连贯性，这使任务更具挑战性。

**上下文数量对性能的影响**为展示局部上下文的重要性，我们采用不同大小的上下文对模型进行处理，以实现跨片段 BERT（Lukasik 等人，2020年）。表 4 显示，增大上下文大小反而会降低性能。我们推测，这是因为句子越多，不同主题越有可能混杂在一起，从而可能妨碍模型理解局部上下文，并

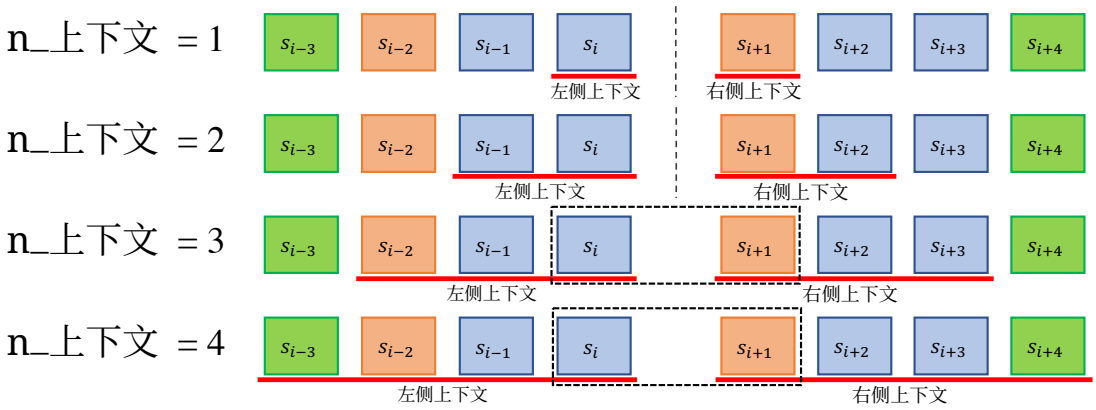


图4：上下文大小对预测的影响。方框的颜色表示句子的主题，红线表示辅助上下文。垂直虚线表示 $s_i$ 和 $s_{i+1}$ 之间的分割点，而虚线框表示这两个句子没有被分割。 $s_i$ 和 $s_{i+1}$ 应当被分开，因为它们属于不同主题，但在 $n\ context\ = 3$ 和 $n\ context\ = 4$ 的情况下却没有被分割。

引入过多噪声。由于使用左右上下文结构同时处理多个句子反而会给上下文带来噪声，因此我们选择分别编码两个输入句子。

图4说明了这种局部上下文捕获错误。图4中的模型都应在 $s_i$ 和 $s_{i+1}$ 之间创建一个分割点，但上下文大小较大的模型未能将这两个句子分开，因为分割只考虑了两侧的整体上下文。随着上下文大小增加，模型会受到泛化的影响，即使两个特定句子涉及不同主题、因而应当被分割，它仍会将左右上下文视为相似。

此外，跨片段 BERT 会同时对左侧和右侧上下文进行编码。由于交叉编码器不可避免地会使一个输入句子的上下文影响另一个句子（Humeau 等，2019年），每个上下文的独特信息可能会意外改变。因此，我们通过孪生句子嵌入对每个句子独立处理，以保留原始局部上下文。

**处理科学文档** 我们还发现，英语疾病的得分低于英语城市。我们认为，这是因为英语疾病涉及的科学领域更为专门（即生物学），而英语城市涵盖的主题相对更广泛。Arnold 等人（2019）指出，英语疾病中的文档采用精确的语言描述，而英语城市中的文档则采用通俗的语言描述。考虑到我们的主干模型 miniLM 是在维基百科等通用文档上预训练的，这一结果似乎在意料之中。

为了提升英语疾病上的表现，我们采用了 SPECTER（Cohan 等人，2020）；该模型以 Sci-BERT 为主干模型，在科学论文上进行训练。

如表5所示， $P_k$ 和 $WinDiff$ 在英语疾病任务上的表现相比基于miniLM的模型分别提升了1.6和1.8。我们将这一提升归因于SPECTER对科学文档的理解。我们预计这些分数

| | $P_k$ | $WinDiff$ |
|---|---|---|
| 英语城市 | 4.6 | 5.2 |
| **英语疾病** | 12.1 | 12.9 |

表5：基于 SPECTER 的主题分割模型在 STP+主题分类+NSP 上的测试 $P_k$ 和 $WindowDiff$ 得分

如果使用更多生物学领域专用的模型（如BioBERT）作为骨干网络，英语疾病任务的分数将会提高。

有趣的是，尽管 SPECTER 的参数数量是 miniLM 的两倍（即 768 对 384），但英语城市上的表现并未提升。从这一结果中，我们再次确认，领域知识对表现至关重要。

## 结论与未来工作

在这项工作中，我们提出了自己的主题分割模型，该模型由 Sentence Transformer 的孪生句子嵌入层和三个分类层组成。通过多项不同的实验，我们证明了所提出的模型优于所有现有模型。我们还发现，以多任务方式结合同主题预测、主题分类和下一句预测能够提升分割性能。

此外，我们通过实验证明了局部上下文在主题分割任务中的重要性。与普遍看法相反，增加上下文数量反而可能因局部上下文的泛化而降低性能。我们的实验表明，通过孪生句子嵌入层缩小上下文范围，有助于保留局部上下文。

未来的工作可以侧重于局部上下文的理论方法。尽管本文通过实验证明了上下文大小对模型性能的影响，但我们并未着重探讨如何确定哪些输入句子能够为执行分割任务提供重要信息。如果我们能够推断每个句子在预测中的重要性——

那么我们希望模型能够自主捕捉重要句子，从而使模型不受上下文大小的影响。

## 致谢

本研究得到韩国政府（MSIT）资助的韩国信息通信技术规划与评估研究院（IITP）项目支持（编号：2020- 0-01361，人工智能研究生院项目（延世大学））。

## 参考文献

Arnold, S.; Schneider, R.; Cudr´e-Mauroux, P.; Gers, F. A.; 和 L¨oser, A. 2019年。SECTOR：用于连贯主题分割与分类的神经模型。Aumiller, D.; Almasian, S. S.; Lackner, M.; 和 Gertz. 2021年。法律文档的结构化文本分割。载于第十八届人工智能与法律国际会议论文集，2–11。计算机协会。Badjatiya, P.; Kurisinkel, L. J.; Gupta, M.; 和 Varma, V. 2018年。基于注意力的神经文本分割。Beeferman, D.; Berger, A.; 和 Lafferty, J. 1999年。文本分割统计模型。机器学习，34：177–210。Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; 和 Weld, D. 2020年。SPECTER：使用引用信息引导的变换器进行文档级表示学习。载于计算语言学协会第*58*届年会论文集，2270–2282。在线出版：计算语言学协会。Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; 和 Bordes, A. 2017年。利用自然语言推理数据监督学习通用句子表示。载于*2017*年自然语言处理实证方法会议论文集，670–680。计算语言学协会。Devlin, J.; Chang, M.-W.; Lee, K.; 和 Toutanova, K. 2019年。BERT：面向语言理解的深度双向变换器预训练。载于计算语言学协会北美分会人类语言技术会议*2019*年论文集，第1卷，4171–4186。计算语言学协会。Gao, T.; Yao, X.; 和 Chen, D. 2021年。SimCSE：句子嵌入的简单对比学习。载于*2021*年自然语言处理实证方法会议论文集，6894–6910。在线出版，地点为多米尼加共和国蓬塔卡纳：计算语言学协会。Humeau, S.; Shuster, K.; Lachaux, M.-A.; 和 Weston, J. 2019年。多重编码器：用于快速、准确多句评分的变换器架构与预训练策略。Koshorek, O.; Cohen, A.; Mor, N.; Rotman, M.; 和 Berant, J. 2018年。作为监督学习任务的文本分割。载于计算语言学协会北美分会人类语言技术会议*2018*年论文集，第2卷，469–473。计算语言学协会。Lo, K.; Jin, Y.; Tan, W.; Liu, M.; Du, L.; 和 Buntine, W. 2021年。基于预训练 Transformer 的 Transformer：增强主题连贯性的神经文本分割。载于计算语言学协会研究成果：*EMNLP 2021*，3334–3340。计算语言学协会。Lukasik, M.; Dadachev, B.; Papineni, K.; 和 Sim˜oes, G. 2020年。基于跨段注意力的文本分段。*2020*年自然语言处理经验方法会议（*EMNLP*）论文集，4707–4716。Pevzner, L.; 和 Hearst, M. A. 2002年。对文本分割评估指标的批判与改进。计算语言学，28(1)：19–36。Reimers, N.; 和 Gurevych, I. 2019年。Sentence-BERT：使用孪生BERT网络的句子嵌入。载于*2019*年自然语言处理实证方法会议暨第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集，3982–3992。计算语言学协会。Xing, L.; Hackinen, B.; Carenini, G.; 和 Trebbi, F. 2020年。改进神经主题分割中的上下文建模。载于计算语言学协会亚太分会首届会议暨自然语言处理第十届国际联合会议论文集，626–636。计算语言学协会。