arXiv:2301.00867v1 [cs.CL] 2 Jan 2023

跟随时间线！按时间顺序生成抽象式与抽取式时间线摘要

XIUYING CHEN*，计算生物科学研究中心，阿卜杜拉国王科技大学
MINGZHE LI*，王选计算机研究所，北京大学SHEN GAO，王选计算机研究所，北京大学
ZHANGMING CHAN，王选计算机研究所，北京大学DONGYAN ZHAO，王选计算机研究所，
北京大学XIN GAO，计算生物科学研究中心，阿卜杜拉国王科技大学XIANGLIANG ZHANG，[1]
圣母大学；[2]阿卜杜拉国王科技大学RUI YAN†，高瓴人工智能学院，中国人民大学

如今，与一般新闻查询相关的带时间戳网络文档在互联网上大量涌现，时间线摘要旨在沿时间线简洁地概述事件的演变轨迹。与传统文档摘要不同，时间线摘要需要对输入事件的时间序列信息进行建模，并按时间顺序概述重要事件。为应对这一挑战，本文提出了一种统一时间线摘要器（UTS），可按时间顺序生成抽象式与抽取式时间线摘要。具体而言，在编码器部分，我们提出了一种基于图的事件编码器，根据多个事件之间的内容依赖关系将其关联起来，并学习每个事件的全局表示。在解码器部分，为确保抽象式摘要符合时间顺序，我们提取其生成过程中保留序列信息的事件级注意力特征，并用它来模拟真实摘要的演化注意力。事件级注意力还可用于辅助抽取摘要，使抽取出的摘要也按时间顺序排列。我们扩充了此前的中文大规模时间线摘要数据集，并收集了一个新的英文时间线数据集。在这些数据集和域外 Timeline 17 数据集上开展的大量实验表明，UTS 在自动评估和人工评估方面均达到了最先进的性能[1]。

CCS 概念：• **信息检索** → 摘要生成。

附加关键词和短语：时间线摘要、抽取式摘要、抽象式摘要

*共同贡献。作者顺序由抛硬币决定。
†通讯作者：Rui Yan (ruiyan@ruc.edu.cn)[1]
https://github.com/iriscxy/Unified-Timeline-Summarizer

作者地址：Xiuying Chen，阿卜杜拉国王科技大学计算生物科学研究中心，xiuying.chen@kaust. edu.sa；Mingzhe Li，北京大学王选计算机研究所，li_mingzhe@pku.edu.cn；ShenGao，北京大学王选计算机研究所，shengao@pku.edu.cn；Zhangming Chan，北京大学王选计算机研究所，zhangming. chan@pku.edu.cn；Dongyan Zhao，北京大学王选计算机研究所，zhaody@pku.edu.cn；Xin Gao，阿卜杜拉国王科技大学计算生物科学研究中心，xin.gao@kaust.edu.sa；Xiangliang Zhang，[1]圣母大学；[2]阿卜杜拉国王科技大学，xzhang33@nd.edu；Rui Yan，中国人民大学高瓴人工智能学院，ruiyan@ruc.edu.cn。

允许为个人或课堂使用而免费制作本作品全部或部分内容的数字或纸质副本，前提是这些副本不得用于营利或商业用途，且不得为此目的进行制作或分发；副本还须附有本声明及首页上的完整引文。本作品中由 ACM 以外的其他方拥有版权的部分，其版权必须得到尊重。允许在注明出处的情况下摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，则须事先获得明确许可和/或支付费用。请向 permissions@acm.org 申请许可。© 2022 美国计算机协会。稿件已提交至 ACM

ACM 引用格式：
Xiuying Chen、Mingzhe Li、Shen Gao、Zhangming Chan、Dongyan Zhao、Xin Gao、Xiangliang Zhang 和 Rui Yan。2022。跟随时间线！按时间顺序生成抽象式与抽取式时间线摘要。*ACM*信息系统汇刊1, 1, 文章 1（2022年1月），30 页。https://doi.org/10.1145/3517221

1 引言

万维网的迅速发展意味着带有时间戳的文档洪流遍布互联网。通用搜索引擎只会根据查询相关性对网页进行排序，但并不太擅长处理意图模糊的查询，例如有关演变中的新闻“新冠肺炎”的查询。人们可能对事件的起因、演变或最新情况有各种各样的普遍兴趣，而仅仅根据相关性对返回的网页进行排序并不足够。许多情况下，读者厌倦了在海量文档中逐一浏览：他们希望只需浏览内容，就能追踪热点话题的演变轨迹。摘要生成是理想的解决方案，它能以精简且信息丰富的方式重新组织文档，从而更快捷、更好地呈现新闻演变。时间线摘要将演变中的新闻按时间归纳为一系列彼此独立但相互关联的组成摘要，因此有助于人们了解事件发展的整体情况 [75]。

现有的时间线摘要方法，例如 [34, 53, 75]，都基于抽取方法。然而，这些方法依赖人工设计的特征，以及对高质量摘要至关重要的复杂能力，例如改写、泛化或融入现实世界知识，而这些只有在抽象式框架中才有可能实现。近年来，随着强大的文本生成式神经网络模型的出现 [4], ，生成式技术也日益受到青睐。因此，我们在早期工作中提出了抽象式时间线摘要任务 [10], ，旨在对输入文章中的事件信息进行简洁改写。表1展示了一个示例，其中文章包含一位著名艺人在不同时期经历的事件，而摘要则按照顺序正确概括了输入文章中的重要事件。

包括 [24, 28, 56, 80] 在内的生成式摘要方法已被证明对传统摘要任务有用。然而，与传统文档摘要不同，时间线摘要数据集由一系列带时间戳的事件构成，因此，时间线摘要模型必须捕捉这些时间序列信息，以更好地指导按时间顺序生成摘要的过程。此外，忠实度问题对于时间线摘要也至关重要，因为混淆不同事件的信息会导致摘要质量低下。以表1 为例，错误的摘要混淆了这位名人的出生地与居住地、首张专辑与最畅销专辑。这里的正确摘要是我们数据集中的真实摘要，而错误摘要则是在初步实验中发现的一则包含典型错误的摘要。正如我们在实验中发现的，这种不忠实现象是摘要任务中普遍面临的问题。

为应对上述挑战，在我们此前的工作中 [10], ，我们提出了一种基于记忆的时间线摘要（MTS）模型。具体而言，我们首先使用带有选择性阅读单元的事件嵌入模块对所有事件进行嵌入。随后，我们提出一种存储时间序列信息的键值记忆模块，以指导摘要生成过程。具体来说，记忆模块中的键是表示时间序列信息的时间位置嵌入，而值则是相应的事件表示。值项包含局部和全局表示，其中局部值是事件嵌入模块的输出，全局值则取自局部表示的平均值。键共同构成一条时间线，我们利用事件在时间线上的时间位置来指导生成过程。稿件已提交至 ACM

事件	迈克尔·杰克逊（被誉为“流行音乐之王”）出生于 1958年8月[illegible]等安纳州加里。他是 他是家里的第七个孩子。
	19[illegible]，Jackson 发布了他的首支个人单曲《Got to Be There》，标志着他个人事业的开端。
	职业生涯：
	198[illegible]ackson发行了他的第六张专辑《Thriller》，其中的音乐视频《Beat It》和《Billie Jean》 被认为打破了种族壁垒，并将这一媒介转变为一种艺术形式 和宣传工具。
	19[illegible]Jackson在加利福尼亚州建造了一处名为梦幻庄园的新居，那里有 100多台街机。
	20[illegible]，吉尼斯世界纪录表彰了他支持39家慈善机构，并捐赠了更多 向以他本人名义成立的慈善机构捐赠了超过3亿美元，比其他任何艺人都多。
质量不佳的摘要	迈克尔·杰克逊于1958年8月2[illegible]福尼[illegible]1年[illegible]地[illegible] 《Thriller》发行。2000[illegible]吉尼斯世界纪录认可他支持了39家 慈善机构。
优质摘要	迈克尔·杰克逊于1958年8月2[illegible]安纳[illegible]于 1982年发行[illegible]200[illegible] 吉尼斯世界纪录认可他支持了39家慈善机构。

表1. 时间线摘要示例。粉色文字表示时间戳，蓝色文字表示错误的事件描述。事件按行划分。

最后，在每个解码步骤中，我们引入事件级注意力，并利用它确定词级注意力，以避免事件之间产生混淆。

在MTS中，时间信息以隐式且间接的方式捕获。MTS将嵌入记忆中的时间位置存储下来，并期望解码器在训练过程中学会关注正确的时间位置。然而，这种策略属于较弱的监督，很难验证和确保解码器确实捕获了时间序列信息。在这项工作中，我们更进一步，通过显式的时间线引导建模改进了此前提出的MTS框架。换句话说，我们精心设计了一种策略，使时间信息成为摘要生成过程中的明确信号。

总而言之，本文提出了一种新颖的统一时间线摘要器 (UTS)，能够按时间顺序生成生成式和抽取式时间线摘要。具体而言，在生成式部分的编码器中，我们首先提出一种基于图的事件编码器，根据多个事件之间的内容依赖关系将它们关联起来，并学习每个事件的表示。其动机在于，每个事件的重要性以及是否应将其纳入摘要，不仅取决于事件本身，还与其他事件相关。以表 1 为例，Jackson 发行首张个人专辑可能是一件重要事件，但其重要性会因他打破种族壁垒的《Thriller》专辑而减弱。因此，图编码器生成的表示融入了其他事件的全局信息，可用于替换记忆中的旧全局表示。在解码器部分，为了避免表 1中的糟糕摘要所呈现的情况——由于模型对时间线不够敏感，混淆了出生地和居住地——我们提出了一种强调时间信息的摘要解码器。具体而言，为了确保生成式摘要遵循时间顺序，我们提取其生成过程中保留序列信息的事件级注意力特征，并用它来模拟真实摘要的演化注意力。

在抽取式部分，我们提出了一种句子嵌入模块，用于对每个句子进行编码。接下来，句子提取器会依次选择要纳入摘要的句子。在此过程中，还可以利用事件级注意力来辅助摘要提取；为此，我们设计了一种时间感知不一致性损失函数，用于惩罚生成式注意力与抽取式注意力之间的不一致。请注意，抽取式摘要是逐条提取的，因此提取出的摘要也按时间顺序排列。

我们在早期工作提出的公开数据集[2] 上对 MTS 和 UTS 进行了实证比较 [10]。这是一个大规模的真实世界时间线摘要数据集，由一系列带时间戳的事件及其对应摘要组成。此外，由于这一先前的数据集仅包含有关名人的时间线语料库，我们还通过加入社会事件案例扩充了该数据集。我们还收集了一个英文时间线摘要数据集。在这些数据集以及域外 Timeline17 数据集上的实验结果表明，我们新提出的 UTS 模型显著优于现有方法。具体而言，与早期工作 MTS 相比，UTS-*abs* 在名人和事件时间线数据集上，ROUGE-1 分别提高了 4.47 和 5.90 个百分点。除了全面评估之外，我们还通过细粒度分析评估了所提出的图编码器和注意力机制。分析揭示了模型如何利用显式时间线信息来指导生成式和抽取式摘要生成过程，并让我们深入了解它们为何能够大幅超越当前最先进的方法。

总而言之，我们的贡献可概括如下：

- 我们提出了一个统一的生成式与抽取式时间线摘要框架，并提出一种具备时间感知能力的不一致性损失函数，以统一这两个过程。
- 我们提出了一种基于图的编码器，根据内容依赖关系关联多个事件，并学习每个事件的全局表示。
- 我们提出使用真实摘要的演化注意力，引导生成式和抽取式摘要生成过程，确保生成的摘要严格遵循时间顺序。
- 我们还通过加入社会事件语料库和英文语料库，扩充了首个真实世界大规模时间线摘要数据集[3]。在三个数据集和域外基准数据集 Timeline 17 上开展的实验表明，我们的模型优于所有基线方法，包括最先进的模型。实验还验证了 UTS 中各模块的有效性及其可解释性。

本文其余部分安排如下：我们在 §2 中总结相关工作。随后，我们在 §3 中阐述研究问题，并在 §4 中详细介绍我们的方法。§5 介绍实验设置的细节，§6 展示实验结果。最后，§7总结全文。

2 相关工作

我们详细介绍文本生成方法、时间线摘要、抽取式摘要、抽象式摘要、统一摘要和记忆网络的相关工作。

2.1 文本生成方法

近年来，基于序列到序列（seq2seq） [61] 的神经网络已被证明能够有效生成流畅的句子。序列到序列模型最初是为机器翻译提出的，后来被应用于各种自然语言生成任务，例如文本摘要 [25, 38, 50, 65, 67] 和对话生成 [6, 63, 77, 79, 81]。

[2]https://github.co
m/yingtaomj/Learning-towards-Abstractive-Timeline-Summarization [3]数据将在最终定
稿版本中发布。提交至 ACM 的稿件

Rush 等人 [54] 将带有注意力模型的序列到序列机制应用于文本摘要领域。随后，See 等人 [56]加入复制机制和覆盖损失，以生成不含词汇表外词语和冗余词语的摘要。序列到序列架构也被广泛应用于对话系统中。Tao 等人 [62] 提出多头注意力机制，以捕捉查询的多个语义方面并生成信息更丰富的回复。Yao 等人 [77]提出使用内容引入方法，解决生成无意义回复的问题。Wang 等人[68] 使用三个通道来拓宽并深入探讨感兴趣的话题，并尝试让对话模型进行更多轮对话。

2.2 时间线摘要生成

时间线摘要任务最初由 Allan 等人提出 [2], ，他们将新闻报道的时间摘要定义为：从某一新闻主题中的每个事件抽取一个句子。随后，一系列研究 [72, 74, 75, 82]进一步探讨了时间线摘要任务。Yan 等人 [75] 通过迭代替换，正式将该任务表述为一个优化问题：从句子集合中选出一个子集，使其满足上述要求，同时兼顾连贯性、多样性度量以及局部和全局摘要质量。在后续研究中，Yan 等人 [72]提出对时间线中各组成摘要之间的跨时间相关性进行建模，采用日期间和日期内的句子依赖关系，并提出了一种新颖的组合方法。另有一些研究关注与时间线摘要相关的推文摘要。例如， [53] 研究了根据用户兴趣选择有意义推文的问题；用户兴趣的动态变化、推文数量庞大以及单条消息内容稀疏，使这一问题颇具挑战性。具体而言，他们基于用户历史记录以及来自“社交圈”的协作性社会影响，研究时间感知推文摘要。Ghalandari 和 Ifrim [26] 使用恰当的评估框架比较了不同的时间线摘要策略。为了进行更稳健的评估，他们还推出了一个新的时间线摘要数据集，其覆盖的时间跨度长于以往的数据集。然而，上述所有研究都基于抽取式方法，不如抽象式方法灵活。

与我们最相似的工作由 [58], 提出，他们构建了一个词邻接图，然后通过寻找从句子起始节点到句子结束节点的路径，从该图中生成新句子。这与我们基于神经网络的方法截然不同，我们也在实验中证明了我们模型的优越性。

2.3 抽取式摘要生成

尽管研究重点在于生成式摘要，抽取式摘要仍是一种颇具吸引力的方法。在抽取式摘要中，Kobayashi 等人 [30] 提出了一种基于词嵌入、利用文档级相似度的摘要生成方法。与此同时，Filippova 等人 [20] 使用循环神经网络删除文档中的词，以进行句子压缩。Yan 和 Wan [76]提出了名为频繁深层依存子结构的、更有意义且信息量更丰富的单元，以及一种用于多文档摘要的主题敏感多任务学习模型。Cheng 和 Lapata [13] 提出了一种通用的单文档文本摘要框架，该框架采用分层文章编码器，并结合基于注意力机制的抽取器。此后，Nallapati 等人 [47] 提出了一种简单的、基于循环神经网络的序列分类器，其表现优于或不逊于当时最先进的模型。Chen 等人 [11]提出了一种模型，通过多次遍历文档，迭代优化文档表示，从而生成更好的摘要。另一种方法是，Narayan 等人 [49] 采用强化学习方法，针对文本摘要优化 ROUGE评估指标。Ren 等人 [52] 研究了句子关系的应用，例如，上下文句子关系、标题句子关系和查询句子关系，以提升抽取式摘要的性能。

近年来，预训练语言模型也被应用于摘要生成，以获取上下文词表示 [39, 83]。图是抽取式摘要的另一种直观结构，可以更好地利用句子之间的统计或语言信息。早期工作主要关注利用句子之间的内容相似度构建的文档图，例如 LexRank [19] 和 TextRank [43]。近期的一些工作旨在通过图神经网络（GNN）将关系先验融入编码器 [78]。

2.4 抽象式摘要

近年来，随着强大的生成式文本 [3],抽象式摘要神经模型的出现，抽象式摘要也越来越受欢迎 [47, 56]。这些模型通常采用卷积神经网络（CNN）或循环神经网络（RNN）的形式。例如，Rush 等人 [54] 提出了一种编码器—解码器模型，使用局部注意力机制生成摘要。Nallapati 等人 [48] 进一步发展了这项工作，解决了基本架构未能充分解决的问题，例如关键词建模以及捕捉句子到词结构的层次关系。在后续工作中，Nallapati 等人 [46] 提出了一种新的摘要模型：每次生成一个句子时，通过采样一个主题来生成摘要，然后使用以句子主题为条件的 RNN 解码器生成词。Zhu 等人 [85] 研究了跨语言摘要任务，其目标是将一种语言的文档摘要为另一种语言。他们提出了一种受翻译模式启发的方法，用于获取跨语言摘要。一系列工作依靠原型文本来辅助摘要生成。Cao 等人 [7] 选择与输入句子相似度最高的模板作为软模板来生成摘要。在此基础上，Gao 等人 [22]提出通过原型编辑生成基于模式的摘要。摘要生成技术也被用于其他任务，例如相关工作生成 [9] 和标题生成 [36]。

2.5 Unified 摘要

这里的统一摘要是指将抽取式和生成式摘要生成任务统一起来。这是提出多任务框架的一种常见方式，利用一个任务的优势来提升另一个任务的性能。例如，Hsu 等人 [28]提出了一个统一框架，通过注意力机制结合句子级注意力，同时利用抽取式和生成式摘要生成的优势。Chen 和 Bansal [12] 引入了一种多步流程，即压缩改写，用于生成式摘要生成：该流程首先从文档中抽取重要句子，然后对其进行重写，以获得最终摘要。Li 等人[33] 引入了一种引导式生成模型，首先使用抽取式模型检索源文本中的关键词。与我们最相似的工作是 [28], , 其中使用句子级注意力来调节词级注意力，使得模型不太可能生成关注程度较低的句子中的词。他们的句子级注意力在生成过程中保持静态，而在我们的模型中，高层注意力会根据当前生成的词在每个解码步发生变化，这更加合理。

2.6 记忆网络

Sukhbaatar 等人提出的记忆网络 [59] 通常由两个部分组成。第一部分是用于保存信息（即 记忆槽）的记忆矩阵，第二部分是用于读取/写入记忆槽的神经网络。与传统的长短期记忆网络相比，记忆网络在多项任务中表现更佳，例如问答 [21, 40, 51, 59], 机器翻译 [42], 文本摘要 [10, 29],对话系统 [16, 69],职位与简历匹配 [73] 以及推荐 [18, 66, 84]。这是因为记忆网络能够在较长时间范围内存储信息，而且其记忆存储单元比只有一个隐藏状态的长短期记忆网络更多。稿件已提交至 ACM

符号	描述
X	一个文档由多个事件组成
Y	g真实时间线摘要 y
$\hat{Y}$	g生成的时间线摘要 y
x_i	输入文档中的第i个事件
w_j^i	第 i个事件中的第j个词
T_e	输入事件的数量
T_w^i	第 i个事件中的词数
T_y	真实摘要中的词数
T_{ys}	真实时间线摘要中的句子数
l_i	提取摘要中第 i句的标签

表 2. 术语表。

继记忆网络之后，人们提出了许多记忆网络的变体，即 键值记忆网络 [45] 和动态记忆网络 [31, 70]。代表性工作包括 [23], 他们采用由选择性写入单元组成的读写记忆，在这些评论之间进行推理，从而在电子商务问答中生成更有意义的答案。

在我们的工作中，我们将键值记忆网络应用于时间线摘要任务，并将其融合到生成过程中。

3 问题形式化

在详细介绍我们的答案生成模型之前，我们首先介绍表 2 中列出的符号。

UTS 将事件列表 $X = (x_1、 \ldots\ldots x_{T_e})$ 作为输入，其中 T_e是事件的数量。每个事件 x_i都是一个词列表：$x_i = (w_1^i、 w_2^i、 \ldots\ldots w_{T_w^i}^i)$，其中 w_j^i是第 i个事件中的第 j个词，而 T_w^i是事件 x_i的词数。

在生成式部分，UTS-*abs* 旨在生成摘要 $\hat{} Y = (\hat{} y_1，\ \ldots\ldots，\ \hat{} y_{T_y})$，该摘要不仅语法正确，还与发生地点和时间等事件信息保持一致。从本质上说，UTS-*abs* 试图优化参数，以最大化概率 $P(Y|X) = \prod_{t=1}^{T_y} P(y_t|X)$，其中 $Y = (y_1，\ \ldots\ldots，\ y_{T_y})$ 是真实摘要。

对于抽取式部分，UTS-*ext* 旨在为每个句子生成一个分数向量 $\hat{} L = \{\hat{} l_1，\ \ldots\ldots，\ \hat{} l_{T_{ys}}\}$，其中每个分数表示该句子的抽取概率。我们将人工撰写的摘要转换为金标准标签向量$L = \{l_1，\ \ldots\ldots，\ l_{T_{ys}}\}$，其中 $l_i \in \{0，\ 1\}$ 表示是否选取第 i 个句子（1 表示选取，0 表示不选取）。在训练过程中，计算 L 与 $\hat{} L$ 之间的交叉熵损失，并通过最小化该损失来优化 $\hat{} L$。

4 模型

4.1 概述

本节将详细介绍我们的统一时间线摘要器（UTS）。UTS 的概览如图1所示，可分为两部分：一部分旨在生成生成式摘要，另一部分旨在选择重要句子作为摘要。

生成式部分包括：(1) 事件嵌入模块（参见§ 4.2）：为了获得每个事件的向量表示，我们采用带有选择性阅读单元（SRU）的循环网络来学习局部表示。(2) 基于图的编码器（参见 § 4.3）：上一个模块学习到的表示未纳入事件之间的交互。

表 3. MTS 与 UTS 的比较。

	MTS	UTS
事件嵌入模块	SRU	SRU
基于图的编码器	-	Transformer
时间-事件记忆	键值记忆	键值记忆
摘要生成器	编辑门	编辑门
句子嵌入模块	-	SRU
句子提取器	-	RNN
统一器	-	不一致性损失

因此，我们提出一种基于图的编码器，用于学习每个事件的全局表示，并融入其他事件的信息及它们之间的关系。(3) 时间-事件记忆（参见 §4.4）：我们提出一种时间-事件记忆，用于存储局部和全局事件表示，并以时间位置键共同构成时间线。(4) 摘要生成器（参见 §4.5）：最后，我们使用基于 RNN 的解码器，在事件级注意力和词级注意力的引导下生成摘要。

抽取式部分包括：(5) 句子嵌入模块（参见 § 4.6）：该模块以类似于事件嵌入模块的方式，将句子嵌入为向量表示。(6) 句子提取器（参见 § 4.7）：句子提取器按照时间先后顺序，选取显著句子作为摘要。

此外，我们提出了 (7) 时间顺序注意力统一器（参见 § 4.8），通过统一生成式部分和抽取式部分的注意力分布，使这两部分相互补充。具体而言，我们提出一种时间感知不一致性损失，以惩罚这两个任务之间的不一致。

尽管 MTS 中的一些编码器和解码器模块与 UTS 相似，但与 MTS 相比，我们的 UTS 模型有三个显著差异：

(1) MTS 会独立编码每个事件，不考虑事件之间的信息交互。而在 UTS 中，我们提出了一种图编码器，用于学习输入事件的全局表示。(2) 我们提出了一个统一时间线框架，不仅能够生成生成式摘要，还能生成抽取式摘要。也就是说，只有 UTS 包含抽取式部分。(3) 我们提出将生成式部分与抽取式部分统一起来，使两项任务能够相互受益。

具体而言，我们在表 3 中展示了 MTS 与 UTS 的对比。

4.2 事件嵌入模块

我们首先提出一个事件嵌入模块，用于获取词级和事件级向量表示。首先，我们使用嵌入矩阵 e 将 x_i 中每个词的独热表示映射到高维向量空间。我们将 $e(w_t^i)$ 记为词 w_t^i 的嵌入表示。随后，我们使用双向循环神经网络（双向循环神经网络）对词之间的时间交互进行建模：

$$\overleftarrow{h_t^i} = \text{LSTM}^{\text{enc}}([e(w_t^i); p^i], \overleftarrow{h_{t-1}^i}), \tag{1}$$

$$\overrightarrow{h_t^i} = \text{LSTM}^{\text{enc}}([e(w_t^i); p^i], \overrightarrow{h_{t-1}^i}), \tag{2}$$

$$h_t^i = \overrightarrow{h_t^i} + \overleftarrow{h_t^i}, \tag{3}$$

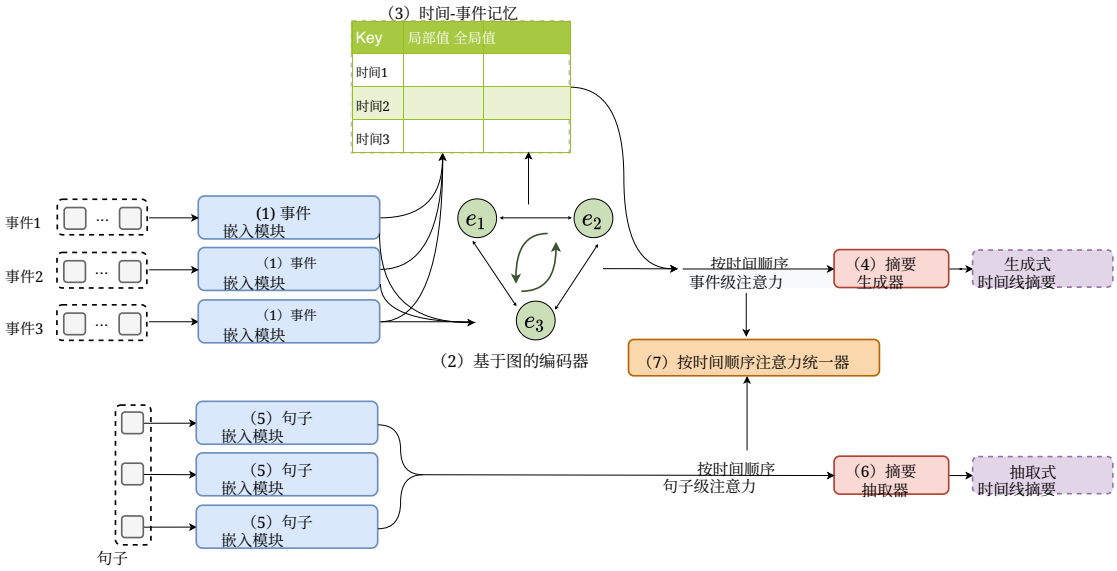

图 1. UTS 概览。我们将模型分为生成式摘要部分和抽取式摘要部分。生成式部分包括：(1) 事件嵌入模块、(2) 基于图的编码器、(3) 时间-事件记忆和 (4) 摘要生成器。抽取式部分包括：(5) 句子嵌入模块 和 (6) 句子提取器。此外，还有一个 (7) 时间顺序注意力统一器，用于统一这两项任务。

其中，“;”表示向量之间的拼接，h_t^i 表示事件x_i在双向循环神经网络中第 t个词的隐藏状态。为捕捉事件的序列信息，我们随机初始化第 i个事件的时间位置编码向量 p^i，并将其纳入双向循环神经网络的输入。

除了获取词表示 h_t^i，我们还需要获得事件表示。仅将双向循环神经网络 $h_{T_w^i}^i$ 的最终状态作为整个事件的表示，无法充分捕捉整个事件的特征。因此，我们采用由 [11] 中提出的 SRU 组成的选择性阅读模块，以获得新的事件表示 a^i:

$$s_t^i = \text{SRU}(s_{t-1}^i, [h_t^i, h_{T_w^i}^i]), \tag{4}$$

$$a^i = s_{T_w^i}^i, \tag{5}$$

其中，s_t^i是第 i个事件中第 t个 SRU 单元的隐藏状态。从较高层面来看，SRU 是 GRU 的一种改进版本，它用一个新门替代了原始 GRU [15] 中的更新门，该新门会同时考虑每个输入 h_t^i和粗粒度事件表示 $h_{T_w}^i$。由于篇幅有限，我们在此省略细节，读者可参阅 [11] 了解详情。至此，我们获得了 a^i中第 i个事件 a^i和第 t个词的表示，即，h_t^i。

4.3 基于图的编码器

上一节中的事件表示 a^i 是独立计算的，没有考虑不同事件之间的信息流动。然而，每个事件的重要性以及是否应将其纳入摘要，不仅取决于事件本身，也与其他事件有关。例如，在表1中，Jackson 发布首张个人专辑可能是一件重要事件，但其重要性会因他打破种族

存在这些障碍。因此，我们提出一种基于图的编码器，以学习事件之间的关系，并获得融合此类信息的全局事件表示。

如图1所示，为了嵌入关系信息，我们在文档建模图中设置了关系边。图中的关系边首先由事件表示初始化：

$$r^{i,j} = \mathrm{MLP}_a([a^i; a^j]), \tag{6}$$

其中，MLP 是多层感知机。

接下来，在关系感知编码过程中，我们通过自注意力操作将关系边 $r^{i,j}$ 融入最终的事件表示中：

$$b^i = \mathrm{RE}(a^i, a^*, r^{i,*}), \tag{7}$$

其中 $*$ 表示 包括 1 到 T_e之间的所有索引。该模块基于 Transformer。因此，我们首先介绍 Transformer ：

$$b^{i'} = \mathrm{Transformer}(a^i, a^*). \tag{8}$$

具体而言，第一个输入用于查询，第二个输入用于键和值。每个输出元素 $b^{i'}$均通过对线性变换后的输入值进行加权和计算得出：

$$b^{i'} = \sum_{j=1}^{T_e} \alpha_g^{i,j} \left(a^j W^V\right). \tag{9}$$

每个权重系数 $\alpha^{i,j}_g$均使用 softmax函数计算：

$$\alpha_g^{i,j} = \frac{\exp\left(\beta_g^{i,j}\right)}{\sum_{k=1}^{T_e} \exp\left(\beta_g^{i,k}\right)}. \tag{10}$$

$\beta^{i,j}_g$通过兼容性函数计算，该函数用于比较两个输入元素：

$$\beta_g^{i,j} = \frac{\left(a^i W^Q\right)\left(a^j W^K\right)^T}{\sqrt{d}}, \tag{11}$$

其中， d是隐藏维度， W^Q， W^K， $W^V \in \mathbb{R}^{d\times d}$是参数矩阵。

RE 与 Transformer 类似，但在公式9 和11中有两处改动。具体而言，我们修改公式9 ，将边信息传播到子层输出：

$$b^i = \sum_{j=1}^{T_e} \alpha_g^{i,j} \left(a^j W_r^V + r^{i,j}\right). \tag{12}$$

这样一来，每个事件的表示都更加全面，因为它包含了与其他事件之间的关系依赖信息。同时，在确定每条边的权重时，即 $\beta^{i,j}_g$我们也纳入了关系边信息，因为密切的关系会对边权重产生很大影响。具体而言，公式11 改为：

$$\beta_g^{i,j} = \frac{\left(a^i W_r^Q\right)\left(a^j W_r^K + r^{i,j}\right)^T}{\sqrt{d}}. \tag{13}$$

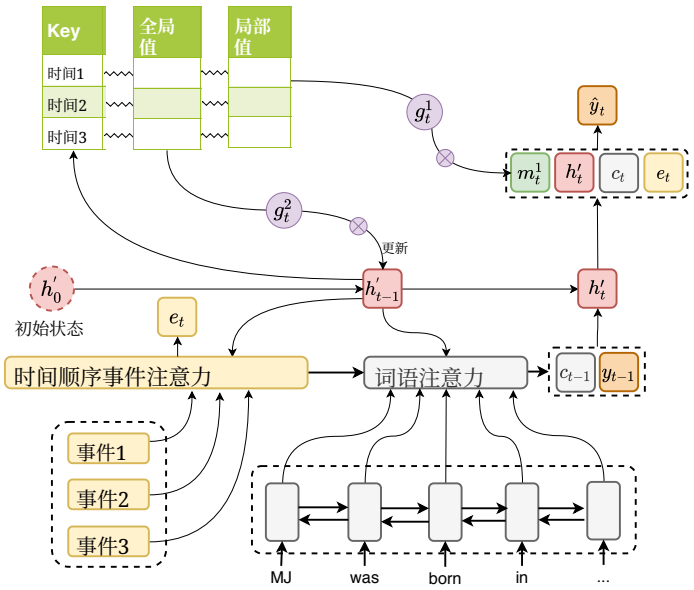

图 2. 抽象式部分中摘要生成器的概览。摘要生成器基于词级注意力、事件级注意力以及键值记忆生成下一个词。

Transformer 架构背后的直觉是，每个输入都不是孤立的，其表示也取决于其他输入。在我们增强后的 Transformer 中，即基于图的编码器，优化后的事件表示 b^i 遵循同样的理念，并扩展了输入文档之间的依赖关系。 b^i 不仅取决于对应的内容，还取决于其他输入以及与其他输入之间的关系。

4.4 时间-事件记忆

如引言中所述，在时间线数据集中，生成的摘要应捕捉时间序列信息，以指导按时间顺序生成过程。因此，我们提出了一种键值记忆模块，其中键共同构成一条时间线，并利用这些时间序列信息指导生成过程，如图2所示。

该记忆中的键是时间位置编码 p^i，在 § 4.2中介绍。我们将使用此键作为时间指导，从记忆的值部分提取信息，具体内容将在 § 4.5中详细介绍。值部分将局部方面的事件信息存储在局部值中，并将全局方面的事件信息存储在全局值中。局部值仅存储事件表示 a^i，这意味着它只捕捉当前事件的信息。另一方面，全局值负责从全局视角学习事件特征，不仅依据事件自身，还考虑其与其他事件的关系。因此，它存储基于图的编码器的输出， b^i。

4.5 摘要生成器

为了生成连贯且信息丰富的摘要，避免因不了解正确的时间线而混淆不同时间戳的信息，我们提出了一种基于 RNN 的解码器，它融合了时间-事件记忆模块的输出和事件表示，如图2所示。

遵循 [35], 我们随机初始化一个 LSTM 单元，以所有事件表示的拼接作为输入，并将其输出用作解码器初始状态：

$$h'_0 = \text{LSTM}^{\text{ini}}\left(h_c, [a^1; ...; a^{T_e}]\right), \tag{14}$$

其中 h_c 是一个随机变量。

4.5.1 词级注意力。 接下来，遵循 [4], 中的传统注意力机制，我们动态地将输入文档概括为上下文向量 c_t，第 t个解码步骤的计算方式如下：

$$h'_t = \text{LSTM}^{\text{dec}}\left(h'_{t-1}, [c'_{t-1}; e(y_{t-1})]\right), \tag{15}$$

其中 h_t 是第 t个解码步骤的隐藏状态。上下文向量 c_t 的计算方式如下：

$$\alpha^{t'}_{i,j} = W_a^\top \tanh\left(W_b h'_{t-1} + W_h h^i_j\right), \tag{16}$$

$$\alpha^t_{i,j} = \exp\left(\alpha^{t'}_{i,j}\right) / \sum_{k=1}^{T_e}\left(\sum_{j=1}^{T^i_w} \exp\left(\alpha^{t'}_{k,j}\right)\right), \tag{17}$$

$$c'_{t-1} = \sum_{i=1}^{T_e}\left(\sum_{j=1}^{T^i_w} \alpha^t_{i,j} h^i_j\right), \tag{18}$$

其中，我们首先使用解码器状态 h_{t-1} 关注各个状态 h^i_j，从而得到注意力分布$\alpha^t_{i,\ j}$，如公式17所示。 h^i_j表示事件 x_i中第 j个词的表示。然后，我们使用注意力分布$\alpha^t_{i,\ j}$对文档状态进行加权求和，得到上下文向量 c_{t-1}。

上下文向量 c_{t-1} 在此仅考虑词级注意力，而未考虑事件级信息。然而，在时间线摘要中，模型必须知道当前描述的是哪个事件，否则可能会混淆不同事件的信息，导致摘要失实。因此，我们引入一种事件级注意力 β，其计算方式与词级注意力类似，并用它来调整词级注意力：

$$\beta^{t'}_i = W_c^\top \tanh\left(W_d h'_{t-1} + W_e a^i\right), \tag{19}$$

$$\beta^t_i = \exp\left(\beta^{t'}_i\right) / \sum_{j=1}^{T_e} \exp\left(\beta^{t'}_j\right), \tag{20}$$

$$\gamma^t_{i,j} = \alpha^t_{i,j} \beta^t_i. \tag{21}$$

新的上下文向量 c_t （替换 c_t 在公式 15中的表达式）现在计算如下：

$$c_t = \sum_{i=1}^{T_e}\left(\sum_{j=1}^{T^i_w} \gamma^t_{i,j} h^i_j\right). \tag{22}$$

4.5.2 事件级注意力。除了使用事件级注意力直接引导词级注意力之外，我们还使用它来获取事件表示的加权和，并将其拼接到公式 31中的投影层：

$$e_t = \sum_{i=1}^{T_e} \beta^t_i a^i. \tag{23}$$

4.5.3 记忆引导。 至此，我们已完成上下文向量的计算。接下来，我们将介绍如何融入记忆的引导信息。首先，我们使用隐藏状态 $h_{\prime t}$ 关注记忆中的每个键。如 § 4.4 所述，键（即时间位置嵌入）构成了表示时间序列信息的时间线。因此，我们让模型利用这些序列信息，并计算位置编码与当前状态之间的相关性，得到时间注意力 $\pi(p^i,\ h_{\prime t})$:

$$\pi(p^i, h'_t) = \exp(h'_t W_e p^i) / \sum_{j=1}^{T_e} \exp(h'_t W_e p^j). \tag{24}$$

随后，通过时间注意力获得记忆中局部值 v_1 和全局值 v_2 的加权和:

$$m_t^{1'} = \sum_{i=1}^{T_e} \pi(p^i, h'_t) v_1^i, \tag{25}$$

$$m_t^{2'} = \sum_{i=1}^{T_e} \pi(p^i, h'_t) v_2^i. \tag{26}$$

$m_t^{1'}$ 和 $m_t^{2'}$ 存储了不同层次的信息，因此在生成器中应发挥不同的作用。

通过融合门，局部值 $m_t^{1'}$ 被更改为 m_t^1，并将被纳入投影层，见公式 31。

$$g_t^1 = W_o([h'_t; c_t; m_t^{1'}]), \tag{27}$$

$$m_t^1 = g_t^1 \cdot m_t^{1'}. \tag{28}$$

我们将局部值置于投影层中，因为 m_t^1 存储的是输入中的详细信息，而非全局特征，因此在生成每个词时应发挥重要作用。

至于全局值 $m_t^{2'}$，它存储了事件在不同位置的全局特征，因此应影响整个生成过程。具体而言，来自 $m_t^{2'}$ 的信息通过一个门融合到解码状态 $h_{\prime t}$ 中:

$$g_t^2 = W_n([h'_t; c_t; m_t^{2'}]), \tag{29}$$

$$h'_t = g_t^2 \cdot h'_t + (1 - g_t^2) \cdot m_t^{2'}. \tag{30}$$

F最后，应用输出投影层，得到词汇表上的最终生成分布 P_v:

$$P_v = \text{softmax}\left(W_v[m_t^1; h'_t; c_t; e_t] + b_v\right). \tag{31}$$

我们将解码器 LSTM $h_{\prime t}$的输出、词上下文向量 c_t、事件上下文向量 e_t和记忆向量 m_t^1拼接起来，作为输出投影层的输入。

为了处理词汇表外（OOV）问题，我们为解码器配备了指针网络 [27, 56]，使解码器能够从源文本中复制词。指针网络的设计与 [56], 中使用的模型相同，因此受篇幅所限，我们省略这部分内容。

我们的目标函 数在生成式部分是目标词的负对数似然 y_t，如公式 32所示 :

$$\mathcal{L}^{\text{abs}} = -\sum_{t=1}^{T_y} \log P_v(y_t). \tag{32}$$

采用梯度下降法更新抽象式部分的参数，以最小化该损失函数。

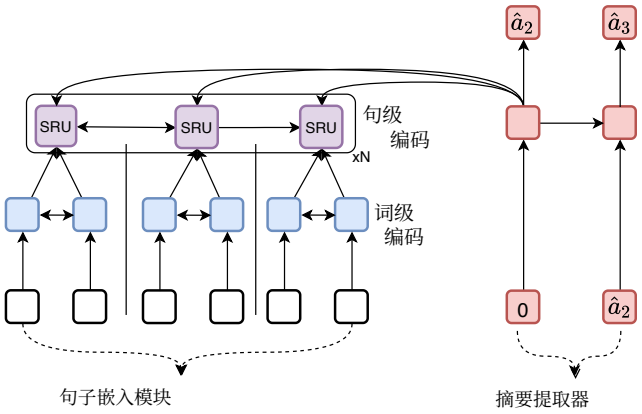

图 3. 抽取式部分中句子提取器的概览。在每个解码步数中，都会依次将一个句子纳入摘要。

4.6 句子嵌入模块

至此，我们介绍了抽象式时间线摘要部分。接下来，我们将介绍 UTS 中的抽取式摘要部分，以及如何统一这两项任务。

我们的句子嵌入模块受到 [11], 的启发，其中嵌入模块同样采用层次结构，并包含一个迭代优化过程，以便更好地对输入文档进行编码。具体而言，我们采用一种新的双向循环神经网络处理每个句子，并获得其表示 $\hat{h}_t^i$，表示 t个词，位于 i个句子中。我们使用最后一个隐藏状态表示整体句子表示，记为 $\hat{h}_{T_w}^i$。文档表示初始化为所有句子表示的平均值：

$$D_1 = \tanh\left(W\frac{1}{T_{ys}}\sum_{i=1}^{T_{ys}}\left[\hat{h}_{T_w}^i\right] + b\right). \tag{33}$$

接下来，为了建模句子之间的序列关系并获得更全面的句子表示，我们迭代优化句子表示和文档表示。为简洁起见，我们以第一次迭代为例说明这一过程。具体而言，在迭代中使用了一个基于 SRU 的循环神经网络（见公式4）：

$$\hat{a}_1^i = \mathrm{SRU}(\hat{a}_0^{i-1}, [\hat{h}_{T_w}^{i-1}, D_1]), \tag{34}$$

$$D_2 = \mathrm{GRU}_{\mathrm{iter}}(\hat{a}_1^{T_{ys}}, D_1), \tag{35}$$

其中，$\hat{a}_1^i$ 是第一次迭代中第 i个 SRU 单元的隐藏状态。通过这种方式，我们迭代地优化句子和文档表示。我们用 I表示迭代次数，因此，第 i个句子的最终表示为 $\hat{a}_I^i$。

4.7 句子提取器

不同于以往为每个句子构建分类器并分配重要性分数的工作，我们使用由 LSTM 单元组成的循环神经网络来选择句子，其中每一步都会选择一个句子，如图 3 所示。遵循 [4], 中的传统注意力机制，我们动态地将输入文档中的句子汇总为上下文向量 c。稿件已提交至 ACM

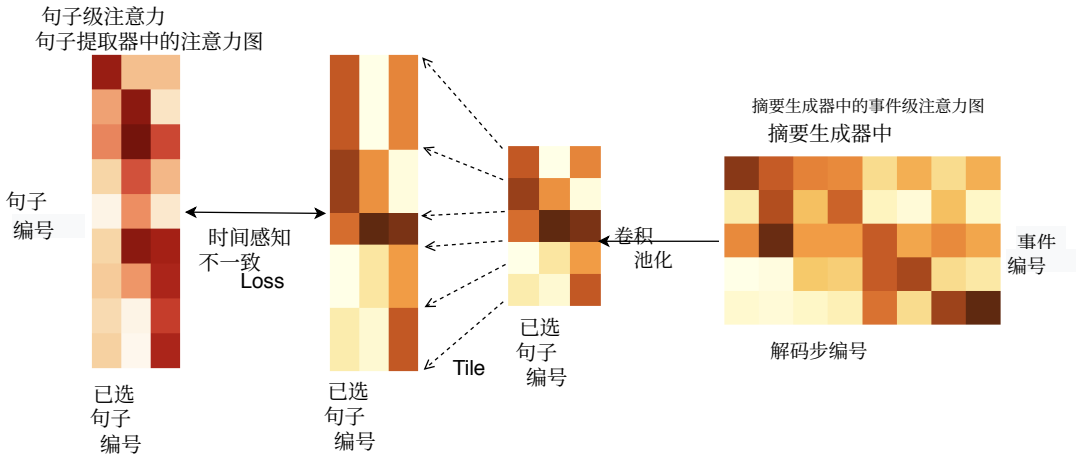

图 4. 时间顺序注意力统一器示意图。经过卷积和平铺操作后，将摘要生成中的事件级注意力与句子提取器中的句子级注意力进行比较，并引入一种新颖的不一致性损失函数，以惩罚这两个层级注意力之间的不一致。

第 t 个解码步骤的计算方式如下：

$$\tilde{h}_{t+1} = \text{LSTM}^{\text{ext}}(\tilde{h}_t, [c_t^{\text{ext}}; \hat{a}_I^{ot}]), \tag{36}$$

$$c_t^{\text{ext}} = \sum_{i=1}^{T_{ys}} \hat{\beta}_t^i \hat{a}_I^i, \tag{37}$$

$$ot = \text{argmax}(\text{MLP}(\tilde{h}_t)), \tag{38}$$

其中 ot 是所选句子在 t 步中的索引，而$\hat{}a_I^{ot}$是先前所选句子的隐藏状态。$\hat{}\beta_t^i$是 i步中第 t个句子的注意力权重，其计算方式与 §4.5.1类似。因此，受篇幅所限，此处省略具体细节。

通过这种方式，我们按顺序生成抽取式摘要，从而更好地捕捉输入中的序列信息：

$$\mathcal{L}^{\text{ext}} = -\sum_{i=1}^{T_{ys}} \log P_s\left(l_i\right). \tag{39}$$

4.8 时间顺序注意力统一器

在生成式和抽取式时间线摘要任务中，对输入文档的注意力都应遵循时间顺序。因此，在训练过程中，促使这两个层级的注意力彼此大致一致，作为一种内在学习目标，而无需额外的人工标注，是很自然的做法（即不需要额外的人工标注）。

在 §4.5.1中，我们提出了生成式部分中的事件级注意力 β；而在 §4.7中，抽取器对输入$\hat{}\beta$采用句子级注意力。每当摘要生成器预测一个新词时，事件级注意力就会更新；而每当选中一个新句子时，句子级注意力就会更新。因此，我们首先使用卷积神经网络（CNN）从生成器中提取不断变化的事件注意力特征。具体而言，如图4所示，一个

沿解码步数轴对事件级注意力图进行卷积，得到新的、以句子编号轴为轴的注意力矩阵。然后，对于每个句子选择步骤，我们将事件级注意力 β_t^i 复制多次，复制次数等于第 i 个事件中的句子编号。

最后，我们希望句子级注意力较高时，事件级注意力也较高。因此，我们设计了以下时间感知不一致性损失：

$$\mathcal{L}^{\text{inc}} = -\frac{1}{T_y}\sum_{t=1}^{T_{ys}} \log\left(\frac{1}{|\mathcal{K}|}\sum_{t\in\mathcal{K}} \hat{\beta}_t \times \beta_t\right), \tag{40}$$

其中 $\mathcal{K}$ 是注意力最高的 K 个句子的集合。这会隐式地促使句子级注意力的分布更加尖锐，并使事件级注意力较高。为避免句子注意力分布退化为独热分布且事件注意力较高的解，我们将用于训练抽取器的原始损失函数（$\mathcal{L}_{\text{ext}}$ 见 §4.7）和抽象器的原始损失函数（$\mathcal{L}_{\text{abs}}$ 见 §4.5）纳入其中。请注意，该模块是抽取器与抽象器交互的唯一部分。我们的时间感知不一致性损失有助于端到端训练的统一模型，使抽取器和抽象器双方均受益。

5 实验设置

5.1 研究问题

我们列出七个指导实验的研究问题：**RQ1** (See § 6.1)：UTS 的总体性能如何？它在多语言数据集上的表现是否优于其他基线方法？**研究问题2** (See §6.2)：我们的模型在域外经典时间线摘要数据集上的表现如何？**研究问题3** (See § 6.3)：UTS 中各个模块的作用是什么？我们的多任务框架是否有助于提升摘要生成性能？**研究问题4** (See § 6.4)：时间位置嵌入是否有用，能否使摘要生成器关注时间-事件记忆中的正确信息？**RQ5** (See § 6.5)：在生成式部分中，事件级注意力能否在解码过程中正确引导词级注意力？**RQ6** (See§ 6.6)：在生成式和抽取式摘要任务中，时间顺序注意力是否成功统一？**研究问题7**(See § 6.7)：参数设置会产生什么影响？

5.2 数据集

据我们所知，目前尚无大规模的时间线摘要生成数据集。因此，在我们先前的工作中，[10],我们从世界上最大的中文百科全书收集了一个大规模时间线数据集[4]。该网站的人物分栏涵盖了古今中外的名人。每个网页都为相应人物提供了时间线摘要；在该网页的人物经历部分，每个事件都以一个段落的形式呈现，并附有说明和详细信息，我们将其选作输入文章。在先前的时间线研究中，[72], 研究者没有从事件新闻文章中预先筛选重要句子，以此检验所提出模型的摘要生成能力。因此，我们的工作也没有对事件段落进行预处理，因为这些事件段落在内容和风格上都与新闻文章相似。我们过滤掉了引用来源和图片等无关内容。我们没有专门从输入中提取时间信息，因为模型以隐式方式学习时间信息，而不是对其进行专门编码。训练数据集共包含169,423个样本，另有5,000个验证样本和5,000个测试样本。文章和摘要的平均长度分别为353.79词和61.19词。

[4]https://baike.baidu.com/提交

数据集	文档数（训练集/验证集/测试集）	平均文档长度		平均摘要长度		词汇量	
		词	句子	词	句子	文档	摘要
TL17	4,650	1,252.33	63.76	43.66	2.73	102,099	6,725
名人 TS	169,423/5,000/5,000	353.79	12.76	61.19	3.97	444,725	191,334
事件时间线摘要	83,188/3,000/3,000	495.19	18.73	141.62	6.05	1,083,249	368,619
维基时间线摘要	140,000/5,000/5,000	606.65	27.79	79.19	7.89	1,029,617	438,011

表4. 摘要数据集在语料库总规模、训练集、验证集和测试集的规模、文档（源端）和摘要（目标端）的平均长度（按词数和句子数计算），以及源端和目标端词汇量方面的比较。TS 表示时间线摘要。

此外，在这项工作中，我们首次在先前的数据集基础上加入了事件时间线摘要案例。中文百科全书中还有一个社会事件子栏目，收录了相关事件随时间推移的发展情况。具体而言，每个页面的发展历史章节都包含描述相应事件发展过程的事件段落，也为这些事件提供了相应的时间线摘要。经过相同的清理操作后，我们得到83,188个训练样本、3,000个验证样本和3,000个测试样本。文章和摘要的平均长度分别为495.19词和141.62词。

请注意，上述两个数据集均为中文。为了测试我们的模型在多语言数据集上的性能，我们从维基百科网站收集了一个英文时间线摘要数据集。由于维基百科中没有人物或事件子栏目，我们通过检查每个网站上摘要和文档中是否包含多个时间戳来筛选时间线页面。其他预处理步骤与中文百科全书类似。对从收集语料中抽取的200个样本进行人工评估后发现，其中196个是时间线文档—摘要对，其中145个与人物有关，51个与事件有关。

由于我们拥有大规模英文摘要数据集，因此可以在规模较小的经典时间线摘要数据集上测试模型的泛化能力。具体而言，我们报告了 UTS 在域外数据集 Timeline 17（TL17）上的性能 [64]。TL17 包含由人工撰写、主题包括内战或英国石油漏油事故等的时间线，这些时间线收集自主要新闻媒体。每个主题还包含一组从网络上抓取的相关新闻文章。

四个数据集的统计信息列于表4中。我们还在表5中给出了时间线统计信息。可以看出，与 TL17 数据集相比，我们的三个数据集规模显著更大。这再次证明了构建这些数据集的必要性：它们规模足够大，可用于训练基于神经网络的模型。在时间线相关属性方面，我们的数据集中的摘要平均每个句子包含更多日期标记，因此要求摘要模型具备更强的时间感知能力。我们的数据集中，文档输入的平均日期数较少，这是因为我们的输入文档比 TL17 短。不过，我们数据集的平均句子/日期比与 TL17 相当，这证明了我们数据集的时间属性。

5.3 对比方法

我们首先进行消融研究，以验证 UTS 中各个模块的有效性。随后，为评估我们提出的数据集和模型的性能，我们将其与以下基线方法进行比较：

生成式基线：

(1) **Pointer-Ge n** [56] 是一种基于循环神经网络的模型，采用注意力机制，并允许系统从源文本中复制词
源文本中的 m
通过指针机制进行生成式摘要。

数据集	文档		摘要长度		压缩率	
	平均日期数	平均每个日期的句子数	平均日期数	平均每个日期的句子数	Sent	Date
TL17	77.55	1.22	1.61	1.14	34.70	32.46
名人 TS	11.03	1.18	3.15	1.10	3.76	3.50
事件时间线摘要	13.09	1.54	5.55	1.19	3.03	2.36
维基时间线摘要	6.81	3.99	2.49	3.17	3.44	2.74

表5. 我们的数据集和 TL17 数据集的时间线特定统计属性。

(2) **FTSum** 利用开放信息抽取和依存句法分析技术，从源文本中提取实际事实描述 [8]。由于中文中没有开放信息抽取工具，我们使用词性标注来提取实体和动词，以替代这些工具。

(3) **Unified**是一种统一模型，结合了抽取式摘要生成和生成式摘要生成的优势，由 [28],提出。该模型使用句子级注意力调节词级注意力，从而降低较少受关注句子中的词被生成的可能性。

(4) **GPG** 是Shen等人 [57] 提出的一种模型，它通过“编辑”指向的词元而非直接复制来生成摘要。编辑过程是通过学习到的关系嵌入，将指向的词向量转换到目标空间中。

(5) **SAGCopy** 是一种增强型Transformer，采用自注意力引导的复制机制，由Xu等人 [71]提出。具体而言，他们首先根据Transformer中自注意力层构建的有向图，通过度中心性确定每个源端词的重要性。随后，他们利用每个源端词的中心性显式引导复制过程。

(6) **MTS** 是我们先前工作 [10]提出的首个抽象式时间线摘要框架。该方法在名人时间线摘要数据集上取得了最先进的性能。

抽取式基线方法：

(1) **Lead3** 是一种抽取式基线方法，将每篇源端文档的前 3 个句子拼接成摘要。(2) **TextRank** [44] 是一种无监督算法，在抽取式句子摘要生成中，根据加权图中的特征向量中心性计算句子重要性分数。

(3) **ITS** 是 [11] 提出的最先进抽取式摘要模型之一。ITS 通过多次遍历文档，迭代优化文档表示，从而抽取更好的摘要。

为了在域外数据集上测试我们的模型，我们将其与一些传统的时间线摘要基线方法进行比较：

(1) **Chieu** [14] 是一种基于直接摘要的无监督基线方法。

(2) **Martschat** [41] 从整个集合中贪心地选择句子组合，使内容覆盖、文本和时间多样性以及日期引用数量最大化，这些目标通过次模函数实现。

(3) **Tran**[5] 是一种原创的逐日期时间线摘要方法，在日期选择和摘要生成中都使用回归，并将某日期的所有句子作为候选句子。

(4) **Pubcount** [26] 是一种简单的逐日期基线方法，使用发表数量对日期进行排序，并将某日期发表的所有句子用于候选选择。

(5) **Datawise** [26] 采用监督式日期选择，使用 PM-MEAN 进行候选句子选择，并使用 CENTROID-OPT 进行摘要生成。

(6) **Clust** [26] 使用日期提及次数对簇进行排名，并使用 CENTROID-OPT 进行摘要生成。这些基线方法的表现与以下结果一致： [26]。

5.4 评估指标

在评估指标方面，我们采用 ROUGE F1 分数 [37]，该指标广泛应用于摘要生成评估 [11, 60]。ROUGE 指标通过计算重叠的词汇单元，将生成的摘要与参考摘要进行比较，包括 ROUGE-1（一元组）、ROUGE-2（二元组）和 ROUGE-L（最长公共子序列）。

对于域外测试数据集，我们遵循 [26], , 并使用特定的时间线评估指标，即，基于对齐的 ROUGE F1 分数和日期 F1 分数。基于对齐的 ROUGE F1 分数比较系统时间线与真实时间线之间的文本重叠，同时也考虑日期与文本的对应关系。日期 F1 分数仅比较系统时间线与真实时间线中的日期。

[55] 指出，仅使用 ROUGE 指标评估摘要质量可能会产生误导。因此，我们还通过人工评估来评估模型。我们请三位受过高等教育的参与者，对 GPG、SAGCopy、MTS 和 UTS-摘要生成的 100 篇随机抽取的摘要进行评分。我们使用双尾配对 t 检验来检验两次运行性能差异的统计显著性，并用 ▲（或 ▼）表示显著性较强，适用于 $\alpha = 0.01$。

5.5 实现细节

我们使用 TensorFlow [1]在 NVIDIA GTX 1080 Ti GPU 上进行实验。所有实验中，模型的隐藏状态维度为 256，词嵌入维度为 128。按照 See 等人 [56], 的做法，我们不对词嵌入进行预训练，而是在训练过程中从头学习。源端和目标端的词汇表均包含 50k 个词。对于时间-事件记忆，键、全局值和局部值的维度分别为 128、512 和 256。我们使用 [-0.02 到 0.02]的均匀分布随机初始化所有参数。批量大小设为 16，事件数量设为 8。对于生成式部分，在训练和测试时，我们将文章截断至 400 个词元，并将摘要长度限制为 70 个词元。对于抽取式部分，我们使用与 [47] 类似的贪心算法，为每篇文档生成预言摘要，以训练抽取式模型。该算法生成由多个句子组成的预言摘要，使其相对于金标准摘要的 ROUGE-2 分数最大化。我们将输入句子数量限制为 24，每个输入句子的长度限制为 20，选取的句子数量限制为 4。对于时间顺序注意力统一器，计算 Linc 时，我们将 K 设为 3。我们采用 Adagrad 优化器 [17]作为优化算法，学习率为 0.15。（在随机梯度下降、Adadelta、动量、Adam 和 RMSProp 中，该设置的效果最佳。）我们使用梯度裁剪，将最大梯度范数设为 2，但不使用任何形式的正则化。我们使用验证集上的损失来实施提前停止。解码时，我们采用束宽为 4 的束搜索，以生成更流畅的摘要句子。在 Timeline 17 数据集这一域外数据集上测试模型时，对于每个包含 S 篇源端输入文档的样例，我们从每篇源文档中取前 400/S 个词元。

在训练效率方面，训练一个轮次约需 9.7 小时，而我们的模型仅经过 3 个轮次便达到最佳性能。相比之下，基线方法 Pointer-Gen 训练一个轮次需要 7 小时，但经过 7 个轮次才达到最佳性能。尤其是，我们的模型在训练初期进展快得多。这证明了我们的统一模型是有效的。在测试方面，为测试数据集中的所有样例生成摘要需要 1.06 小时。我们根据验证集上的评估损失选取排名前三的检查点，并报告测试集上的平均结果。

模型	名人时间线数据集			事件时间线数据集			维基时间线数据集		
	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
句子抽取方法									
Lead3	32.36	17.96	30.99	21.47	9.26	15.73	25.35	5.94	20.56
TextRank	32.27	15.34	30.86	23.89	10.43	16.66	24.98	5.47	22.40
ITS	34.03	18.20	31.24	27.94	14.28	20.39	27.82	5.91	25.37
Unified-文	34.18	18.29	31.16	28.06	14.39	20.47	26.48	5.82	24.28
UTS-文	**34.81**	**22.26**	**32.03**	**29.12**	**16.01**	**23.06**	**29.00**	**6.64**	**25.81**
抽象式方法									
Pointer-Gen	36.61	21.35	34.51	22.56	7.84	21.00	23.12	5.07	19.65
FTSum	37.84	21.47	35.37	23.41	6.95	21.66	24.08	5.80	20.05
Unified-生成式	38.24	21.95	36.42	23.58	7.93	21.95	24.34	5.84	20.37
GPG	38.43	21.59	36.38	22.38	7.77	20.81	24.71	5.80	20.85
SAGCopy	38.64	20.84	36.41	23.40	7.95	21.72	26.00	5.84	22.01
MTS	39.78	22.24	37.69	23.89	8.38	21.97	26.68	5.88	23.18
UTS-生成式	**41.56**	**23.95**	**39.18**	**25.30**	**9.63**	**23.28**	**27.71**	**5.92**	**24.62**

表6。RQ1：基线方法之间的 ROUGE分数比较。上半部分的模型和基线方法为抽取式，下半部分的则为生成式。根据官方ROUGE脚本报告，我们所有 ROUGE分数的 95% 置信区间均不超过 ±0.24。

6 实验结果

6.1 总体性能

针对研究问题 **RQ1**，我们根据表 6所示的 ROUGE 分数，考察了我们的模型和基线方法的表现。首先，在名人时间线数据集上，生成式模型在我们的数据集上的表现显著优于抽取式模型。我们认为，这是因为该数据集的金标准摘要往往会使用新的表达方式来概括原始输入文档。这表明采用抽象式时间线摘要方法十分必要。其次，我们将先前的模型MTS 与近期提出的基线方法进行比较，其中包括SAGCopy 和 GPG。这两种基线方法在我们的数据集上取得的 ROUGE 分数低于 MTS，这证明了我们先前模型的有效性。最后，在 MTS的基础上，我们改进后的模型 UTS-*ext* 和 UTS-*abs* 取得了更好的表现。

具体而言，在生成式部分，在名人时间线数据集上，UTS-*abs* 在 ROUGE-1、ROUGE-2 和 ROUGE-L 指标上分别比 SAGCopy 高出 7.56%、14.92% 和 7.61%，比 MTS 高出 4.47%、7.68% 和 3.95%。在抽取式部分，与 ITS 相比，我们的抽取方法在名人时间线数据集上的 ROUGE-2 得分提高了约 2.23 个百分点。我们将这一提升归因于两个方面：首先，生成式目标能够促进基于时间顺序注意力统一网络的抽取式模型识别重要句子。此外，尽管抽取式金标准标签序列是通过在金标准摘要上贪心优化 ROUGE-2 得到的，但这些标签可能并不准确。联合学习两个目标或许能够纠正不准确标签给抽取式模型带来的一些偏差。上述结果证明了我们模型的优越性。需要注意的是，我们主要将模型与 ITS 进行比较，因为我们的抽取式部分主要基于 ITS。我们的框架也可以应用于其他抽取式模型，理论上能够使两项任务均受益。我们将此留作未来工作。

我们的人工评估研究通过邀请三位受教育程度较高的参与者，根据以下标准对名人时间线数据集上的摘要进行排名，从而评估其整体质量：流畅度（稿件提交给 ACM

	流畅度	信息量	忠实度
GPG	2.59	2.53	2.39
SAGCopy	2.64	2.57	2.43
MTS	2.71	2.58	2.61
UTS-生成式	**2.77**▲	**2.62**▲	**2.65**▲

表 7. RQ1：在 celebritytimeline 数据集上与主要基线方法进行的人工评估比较。

摘要是否流畅且符合语法规范？）、信息量（摘要是否传达了有关所讨论主题的重要事实？）以及忠实度（摘要是否忠实于输入内容？）。我们选择SAGCopy 和GPG 作为基线方法，因为与其他基线方法相比，它们的表现相对较好。评分范围为 1 到 3，3 分为最佳。结果见表7。可以看出，我们的模型表现远优于所有基线方法。在流畅度指标上，我们的模型获得了 2.77 的高分，高于GPG 的 2.59 和SAGCopy的 2.64，这表明我们的模型能够减少语法错误并提高摘要的可读性。在信息量指标上，我们的模型比SAGCopy高出 0.05，说明我们的模型能够有效捕捉显著信息。在忠实度指标上，UTS-*abs* 大幅优于所有基线方法，这表明多粒度语义信息以及与抽取式摘要的联合学习确实有助于避免生成摘要中出现不忠实的信息。值得注意的是，不忠实性是时间线摘要中的一个严重问题，而MTS 和 UTS-*abs* 大大缓解了这一问题。我们还对模型与SAGCopy （带阴影背景的行）进行了配对学生 t 检验，结果证明了上述结果具有显著性。κ统计量分别为 0.46 和 0.49，表明标注者之间的一致性适中[5]。为了证明这些结果具有显著性，我们还对模型与SAGCopy进行了配对学生 t 检验。流畅度、信息量和忠实度的 p 值分别为 3×10^{-8}， 8×10^{-12}，以及 9×10^{-11} 。

我们还在表 9 中展示了一项案例研究，表 8 为从 celebritytimeline 数据集中选取的译文。该案例讲述詹姆斯·卡梅隆的导演生涯。由于篇幅有限，我们省略了输入文档中的不重要信息。输入文档涵盖了他的大部分作品以及各事件的详细信息，而摘要参考文本仅介绍了其经历中的主要事件，省略了这些细节和不重要的事件。可以看到，UTS-*abs* 生成的摘要成功捕捉了重要事件，并按正确顺序进行介绍。我们的 UTS-*ext* 输出与真实标注的重合度很高。至于基线方法 GPG，它未能捕捉最重要的事件，却包含了无关信息，例如《食人鱼2》的拍摄细节。基线方法 SAGCopy也生成了不重要的描述，其中包括“《末世黑天使》”的信息。此外，我们的抽取式摘要和生成式摘要表现一致，重合度较高，这进一步表明这两种方法能够协同促进对重要信息的识别。与抽取式摘要相比，生成式摘要更加简洁、连贯。

6.2 域外测试

接下来，我们回答研究问题**研究问题2**。表10展示了 UTS 在经典时间线摘要 TL17 数据集上的表现，并将其作为域外测试。可以看出，我们的两种模型都优于现有基线方法。具体而言，UTS-*ext* 在 AR1-F 分数上比最佳基线方法 Datawise 高出 19.4%，这表明

[5][32] 表征 k appa 值 < 0 表示无一致性，0-0.20 表示轻微一致，0.21-0.40 表示一般一致，0.41-0.60 表示中等一致，0.61-0.80 表示高度一致，0.81-1 表示几乎完美的一致性美的一致性。

	在1981年，詹姆斯·卡梅隆执导了他的首部电影《食人鱼2》，全片都在意大利拍摄。卡梅隆与讲意大利语的工作人员相处得并不融洽，制片人也不让他参与影片的最终剪辑……在1984年，卡梅隆推出了他的首部自编自导电影《终结者》，制作成本仅为650万美元……在1986年，詹姆斯·卡梅隆的第二部自编作品《异形2》上映……在1987年，《异形2》获得了七项奥斯卡金像奖提名……詹姆斯·卡梅隆凭借这部电影获得第14届土星奖最佳导演奖。在1989年，卡梅隆编剧并执导了他的第三部电影《深渊》……在1991年，他的电影《终结者2》在美国票房达到2亿美元，他还凭借该片获得第18届土星奖最佳导演和最佳编剧奖。在1997年，卡梅隆执导了电影《泰坦尼克号》，该片票房达18.4亿美元，由莱昂纳多·迪卡普里奥和凯特·温斯莱特主演。在1998年，该片在第70届奥斯卡金像奖上获得14项提名和11项奖项。在2000年，他执导并监制了以基因治疗为主题的电视剧《末世黑天使》。
参考文献	在1981年，他执导了首部电影《食人鱼2》 。在1984年，他凭借科幻电影声名鹊起《终结者》。在1986年，他编剧并执导了电影《异形2》。在1991年 ，他获得了最佳导演奖他凭借电影《终结者2》获得第18届土星奖最佳导演奖和最佳编剧奖。在1997年 ，他的电影《泰坦尼克号》票房达18.4亿美元，打破了全球票房纪录；它在第70届奥斯卡金像奖上获得包括最佳影片奖在内的11项大奖，詹姆斯·卡梅隆则获得了奥斯卡奖最佳导演奖。
GPG	在1981年，詹姆斯·卡梅隆执导了电影《食人鱼2》。卡梅隆卡梅隆与一群说意大利语的工作人员相处得并不好，制片人也不让他参与影片的最终剪辑。在1984年，他凭借科幻电影《终结者》声名鹊起。在1986年 ，他编剧并执导了电影《异形2》。在1986年，他编剧并执导了电影《异形2》。在1991年 ，他获得了最佳导演奖凭借电影《终结者2》获得第18届土星奖最佳编剧奖和最佳编剧奖。在1997年 ，他的电影《泰坦尼克号》票房收入达18.4亿美元。
SAGCopy	在1981年，詹姆斯·卡梅隆执导了他的第一部作品《食人鱼2》。在1984年，卡梅隆推出了他的第一部电影，《终结者》。在1986年，詹姆斯·卡梅隆编剧并执导了他的第二部作品《异形2》。在1997年，詹姆斯卡梅隆执导了电影《泰坦尼克号》。在2000年，他执导并监制了以基因治疗为主题的电视剧《黑暗天使降临末世》。
UTS-扩展	在1981年，詹姆斯·卡梅隆执导了他的第一部电影《食人鱼2》，该片完全在意大利拍摄。在1984年，卡梅隆推出了他首部自编自导的电影《终结者》，制作成本仅为650万美元美元。 1986年，詹姆斯·卡梅隆的第二部自编作品《异形2》上映。到了 1997年，卡梅隆执导了电影《泰坦尼克号》，该片票房达18.4亿美元，主演是莱昂纳多·迪卡普里奥和凯特·温斯莱特。
UTS-抽象式	在 1981年，卡梅隆执导了他的第一部作品 ，《食人鱼2》。在 1984年，他发行了自己的首部电影《终结者》。在 1986年，他的第二部电影《异形2》上映。在 1987年，《异形2》获得了第14届土星奖的最佳导演。在 1991年 ，他的电影《终结者2》在美国票房达2亿美元。在 1997年 ，他执导了电影《泰坦尼克号》。在1998年，该片获得了14项奥斯卡金像奖提名，其中11项在第70届奥斯卡金像奖颁奖典礼上。

表8。RQ1：UTS-生成式、UTS-抽取式和基线方法生成的答案示例（翻译版本）。

神经网络在传统抽取式风格下的有效性。UTS-生成式的表现与UTS-抽取式相近，使Pubcount的AR1-F 分数提高了3.79。这表明，抽象式方法可以适用于域外小规模数据集。具体而言，我们原始的WikiTS数据集采用百科全书风格，而Timeline 17是一个新闻数据集，这表明我们的模型可以应用于采用不同语言风格的数据集。

6.3 消融研究

接下来，我们转向研究问题**研究问题3**，在测试集上进行消融研究，以考察我们提出的 UTS 模型中不同模块的影响。我们通过四种方式测试这些模块：(1) 移除句子提取器 稿件已提交至 ACM

1981年，詹姆斯·卡梅隆执导了他的第一部作品《食人鱼2》，影片完全在意大利拍摄。卡梅隆与一口意大利语的工作人员相处得并不愉快，拍摄完成后，制片方也不让他参与影片的最终剪辑。1984年，卡梅隆推出了他第一部自编自导的影片《终结者》，这部影片的拍摄只花了650万美元.....1986年，詹姆斯·卡梅隆自编自导的第二部作品《异形2》问世.....1987年，《异形2》获得了七项奥斯卡奖提名.....詹姆斯·卡梅隆凭借此片获得了第14届土星奖最佳导演奖.....1989年，卡梅隆自编自导了第三部电影《深渊》.....1991年，他执导的电影《终结者2》在美国上映后取得了2亿美元的票房，他也凭借该片获得了第18届土星奖最佳导演奖以及最佳编剧奖.....1997年，詹姆斯·卡梅隆执导了电影《泰坦尼克号》，该片获得了18.4亿美元的票房，由莱昂纳多·迪卡普里奥、凯特·温斯莱特等主演。1998年，在第70届奥斯卡金像奖上，这部影片获得了14项奥斯卡奖提名，并获得了其中的11个奖项.....2000年，他执导并监制了以基因治疗为题材的电视剧《末世黑天使》.....	
参考	1981年年，詹姆斯·卡梅隆执导首部电影《食人鱼2》。1984年，因自编自导科幻电影《终结者》成名。1986年年,自编自导电影《异形2》。1991年年,凭借电影《终结者2》获得第18届土星奖最最佳导演奖以及最佳编剧奖。1997年,他执导的电影《泰坦尼克号》取得了18.4亿美元的票房,打破全球影史票房纪录;该片在第70届奥斯卡金像奖上获得了包括最佳影片在内的11个奖项,詹姆斯·卡梅隆凭借该片获得了奥斯卡奖最佳导演奖。
GPG	1981年年，詹姆斯·卡梅隆执导了部作品《食人鱼2》，卡梅隆卡梅隆和一口意大利语的工作人员相处得并不愉快，而拍摄完毕后制片方不让他参与影片的最终剪辑。1984年年，卡梅隆凭借科幻电影《终结者》出名。1986年年,他自编自导了电影《异形2》。1986年年,他自编自导了电影《异形2》。1991年年,他凭借《终结者2》获得了第18届土星奖最佳导演奖和最佳编剧奖。1997年年,他的电影《泰坦尼克号》在美国获得了18.4亿票房。
SAGCopy	1981年年，詹姆斯·卡梅隆执导了第一部作品《食人鱼2》。1984年年，卡梅隆推出了他第一部自编自导的影片《终结者》。1986年年,詹姆斯·卡梅隆自编自导的第二部作品《异形2》。1997年年，詹姆斯·卡梅隆执导了电影《泰坦尼克号》。2000年年，他执导并监制了以以基因治疗为题材的电视剧《末世黑天使》。
UTS-*ext*	1981年年，詹姆斯·卡梅隆执导了第一部作品《食人鱼2》，影片完全在意大利拍摄。1984年年，卡梅隆推出了他自编自导的首部电影《终结者》。这部影片的拍摄只花了650万美元元。1986年年,詹姆斯·卡梅隆自编自导的第二部作品《异形2》问世。1997年年,詹姆斯·卡梅隆执导了电影《泰坦尼克号》,该片获得18.4亿美元的票房,由莱昂纳多·迪卡普里奥、凯特·温斯莱特等主演。
UTS-摘要	1981年年，卡梅隆执导了第一部作品《食人鱼2》。1984年年，卡梅隆推出了他第一部自编自导的影片《终结者》。1986年年,自编自导的第二部作品《异形2》问世。1987年年,《异形2》获得了第14届土星奖最佳导演奖。1991年年,他执导的电影《终结者2》在美国上映后取得了2亿美元的票房。1997年年,执导了电影《泰坦尼克号》。1998年年,在第70届奥斯卡金像奖上这部影片获得了14个奥斯卡奖提名，并获得了其中的11个奖项。

表9。RQ1：UTS-摘要式、UTS-抽取式 和基线方法生成答案的示例。

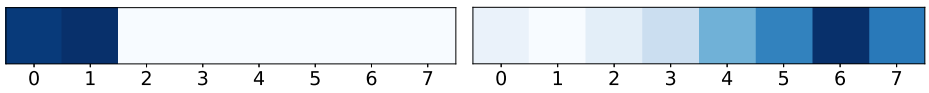

图5. 研究问题3：时间注意力的可视化。左侧图像展示了第一个解码步骤中的注意力图，右侧图像展示了最终解码步骤中的注意力图。

并且只训练生成器，以验证联合学习对抽象式摘要生成的有效性；（2）我们移除摘要生成器部分，只训练句子提取器，以验证联合学习对抽取式摘要生成的有效性；（3）我们移除基于图的编码器，只将局部表示存储在

	AR1-F	AR2-F	Date-F1
Chieu	6.66	1.9	25.1
Martschat	10.5	3.0	54.4
Tran	9.4	2.2	51.7
Pubcount	10.5	2.7	48.1
Datewise	12.0	3.5	54.4
Clust	8.2	2.0	40.7
UTS-提取式	**16.73**	**4.08**	**54.9**
UTS-摘要	14.29	3.51	54.6

表 10。研究问题2：在域外 TL17 摘要数据集上的 ROUGE分数。

	ROUGE-1	ROUGE-2	ROUGE-L
UTS-摘要	41.56	23.95	39.18
不使用多任务	39.58	22.54	37.55
不使用全局	38.66	22.87	36.76
不使用局部	39.14	23.15	36.00
UTS-抽取式	34.81	22.26	32.03
无多任务	33.78	21.09	29.03
无全局表示	33.00	18.89	27.07
无局部表示	33.28	20.98	29.69

表11. 研究问题3：不同消融模型的ROUGE分数。

记忆中，以验证全局表示的有效性；（4）我们完全移除时间-事件记忆，以进一步验证全局和局部表示的重要性。

表11展示了结果。移除摘要生成器后，抽取式摘要的ROUGE-2分数下降了5.26%。这表明，联合学习方法有助于抽取式摘要从生成式摘要中受益。移除句子提取器后，生成式摘要的ROUGE-2分数下降了5.54%。这表明，在交互式解码阶段，抽取式摘要确实有助于生成式摘要识别重要句子。移除全局表示后，抽取式摘要的ROUGE-2分数下降了4.72%，而生成式摘要的ROUGE-2分数下降了6.25%。这表明，建立基于图的编码器来模拟事件之间的关系，对于提升抽取式摘要和生成式摘要的性能是必要的。移除全局表示和局部表示后，ROUGE-2分数相较于UTS-生成式摘要分别下降了4.72%和3.45%。这表明，时间-事件记忆的语义信息对于编码多个事件至关重要。

6.4 时间位置嵌入分析

接下来，我们探讨**研究问题4**。记忆中的时间注意力体现了时间位置嵌入的作用，记为 π，如公式24所示。如果时间位置嵌入成功编码了时间信息，那么时间注意力就应遵循输入文档的发展顺序。我们对从名人时间线数据集中随机抽取的两个示例绘制了注意力图，如图5 所示。左图是第一个解码步骤中的注意力图，右图则是最后一个解码步骤中的注意力图。颜色越深，注意力越高。提交至 ACM 的稿件

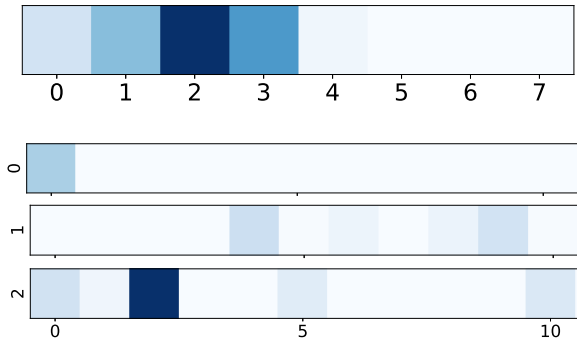

图 6. RQ5：可视 化的两级注意力 . 上图展示的是事件 -级注意力，下方三幅图展示的是词级注意力首个 lead 三个事件的.....

由于篇幅有限，我们省略相应的事件描述。解码开始时，UTS-*abs*学会关注前两个事件，它们通常包含并列信息，例如人物的出生地和出生日期。由于无需提前获取这些信息，因此对最后几个事件的注意力较低。解码结束时，UTS-*abs* 会更多地关注最后几个事件。不过，它也会关注前几个事件，因为时间线摘要是一个信息累积的过程，后续句子需要考虑先前的信息。以上示例展示了时间位置嵌入的有效性。

6.5 事件级注意力分析

现在我们转向**RQ5**：事件级注意力能否在生成式部分引导词级注意力。我们首先进行案例研究，以可视化两级注意力，如图6所示。上方的图展示事件级注意力，下方的三幅图展示对应于前三个事件的词级注意力。我们只展示一个事件中的前 11 个词。结果表明，在这一步解码中，第三个事件是最重要的事件，而且该事件中各词的权重平均而言也高于其他词。以上观察表明，事件级注意力为词级注意力提供了正确的引导。

除了可视化之外，我们还进行了定量分析，以衡量事件级信息对词级注意力的影响程度，这体现在不一致性损失上。我们调整了 §4.8 中提出的不一致性损失，以评估事件注意力与词注意力之间的不一致性。在第 t 步解码时，新的 一致性损失是对注意力值最高的词的注意力值及其对应的事件级注意力的乘积取负对数似然。其直观目的是验证词级注意力较高时，事件级注意力是否也较高。训练开始时，不一致性损失约为 5.3；训练结束时，该损失降至 2.1。这意味着事件级信息会显著影响词级注意力，并且模型学会了统一这两种注意力。我们没有直接将不一致性损失加入训练，因为我们发现这样会导致 UTS 的表现变差。相反，我们让模型自行学习统一这两种注意力。

6.6 Unified 时间顺序注意力分析

接下来，我们探讨**RQ6**，考察生成式部分和抽取式部分中的时间顺序注意力是否确实统一。请记住，我们提出了时间感知不一致性损失来统一这两种注意力，因此，通过观察损失曲线，我们可以评估这部分的效果。

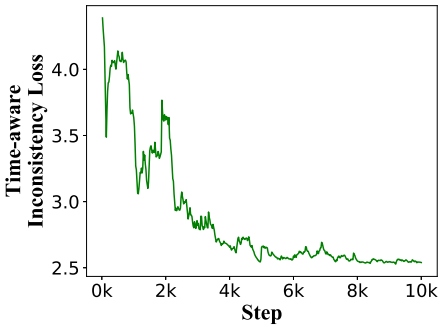

图 7. RQ6：时间感知不一致性损失曲线。

不一致性损失的曲线如图7所示。我们可以看到，在训练开始时，不一致性损失时有波动，这可能是因为模型在训练过程初期尝试分别训练抽取器和生成器。不过，不一致性损失的平均值呈下降趋势，这意味着在整个训练过程中，抽取器和生成器逐渐趋于统一。最终，时间感知不一致性损失从 4.0 降至 2.5。

6.7 参数设置的鲁棒性

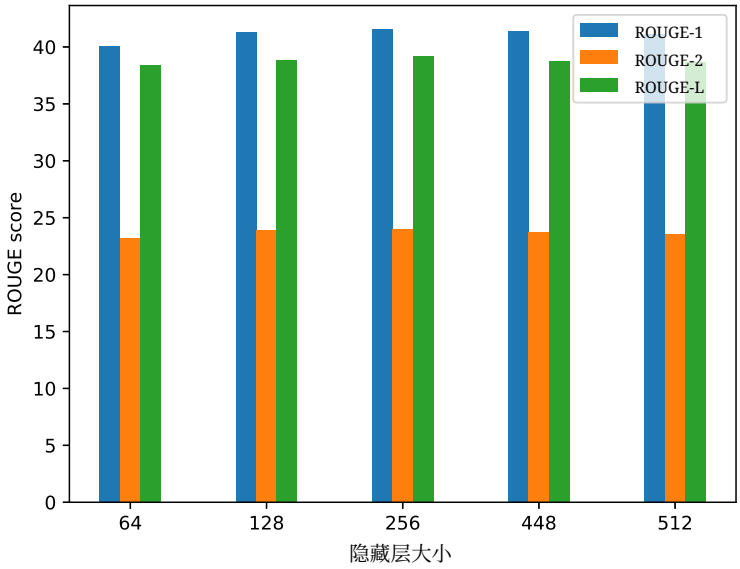

图 8. UTS-摘要 在不同参数设置下的性能。

最后，我们来回答**研究问题7**，以探究参数设置的鲁棒性。我们在不同的参数设置下训练模型，如图所示8。循环神经网络的隐藏层大小在 64 到 512 之间进行调整，并使用 ROUGE F_1分数来评估各个模型。随着隐藏层大小从 64 增至 256，性能也随之提升。隐藏层大小的增加使 ROUGE-1 和 ROUGE-L 分数分别提高了 0.54 和 0.77 分。当隐藏层大小继续从 256 增至 512 时，性能略有下降。隐藏层大小的增加使 ROUGE-1 和 ROUGE-L 分数分别下降了 1.15% 和 1.25%。尽管如此，我们可以发现，各项指标都保持在稳定范围内，这表明我们的 UTS 对不同的参数规模具有鲁棒性。

7 结论与未来工作

在我们之前的工作中，我们提出了一种名为 MTS 的框架，旨在生成能够简洁概括时间线中演变轨迹的摘要。然而，该方法以隐式且间接的方式捕捉时间信息，因此难以验证并确保解码器确实捕捉到了时间序列信息。因此，在本工作中，我们提出了一种新颖的统一时间线摘要器 (UTS)，能够按时间顺序生成生成式和抽取式时间线摘要。具体而言，在编码器部分，我们提出了一种基于图的事件编码器，根据多个事件之间的内容依赖关系将其关联起来，并学习每个事件的表示。在解码器部分，为确保生成式摘要遵循时间顺序，我们在其生成过程中提取保留了序列信息的事件级注意力特征，并用其模拟真实摘要的演化注意力。事件级注意力还可用于辅助摘要抽取；为此，我们设计了一种时间感知不一致性损失函数，用于惩罚抽象式注意力与抽取式注意力之间的不一致。需要注意的是，抽取式摘要是逐条生成的，因此抽取出的摘要也遵循时间顺序。我们将之前工作提出的角色时间线摘要数据集与事件时间线摘要语料库和英文语料库结合，对其进行了扩充。在这些数据集以及域外 Timeline 17 数据集上的实验结果表明，我们的 UTS 模型显著优于现有方法。近期，我们计划提出一种多模态时间感知时间线摘要框架。

致谢

我们谨此感谢匿名审稿人提出的建设性意见。本研究得到国家重点研发计划（编号：2020YFB1406702）和国家自然科学基金（国家自然科学基金项目编号：62122089和61876196）的资助。

参考文献

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard 等。2016。TensorFlow：一个面向大规模机器学习的系统。发表于 OSDI，第 16 卷。265–283。[2] James Allan、Rahul Gupta 和 Vikas Khandelwal。2001。新主题的时间摘要。发表于 SIGIR。ACM，10–18。[3] Dzmitry Bahdanau、Kyunghyun Cho 和 Yoshua Bengio。2014。通过联合学习对齐与翻译实现神经机器翻译。arXiv 预印本 arXiv:1409.0473（2014）。[4] Dzmitry Bahdanau、Kyunghyun Cho 和 Yoshua Bengio。2015。通过联合学习对齐与翻译实现神经机器翻译。发表于 ICLR。[5] Giang Binh Tran、Mohammad Alrifai 和 Dat Quoc Nguyen。2013。预测时间线摘要中的相关新闻事件。发表于第 22 届国际万维网大会论文集。91–92。[6] Deng Cai、Yan Wang、Wei Bi、Zhaopeng Tu、Xiaojiang Liu 和 Shuming Shi。2019。通过匹配到生成框架实现检索引导的对话回复生成。发表于 EMNLP。[7] Ziqiang Cao、Wenjie Li、Sujian Li 和 Furu Wei。2018。检索、重排与改写：基于软模板的神经摘要生成。发表于计算语言学协会第 56 届年会论文集（第 1 卷：长篇论文）。152–161。[8] Ziqiang Cao、Furu Wei、Wenjie Li 和 Sujian Li。2018。忠于原文：具有事实感知能力的神经抽象式摘要。发表于 AAAI。[9] Xiuying Chen、Hind Alamro、Mingzhe Li、Shen Gao、Xiangliang Zhang、Dongyan Zhao 和 Rui Yan。2021。捕捉科学论文之间的关系：用于生成相关工作章节的抽象式模型。发表于 ACL。[10] Xiuying Chen、Zhangming Chan、Shen Gao、Meng-Hsuan Yu、Dongyan Zhao 和 Rui Yan。2019。面向抽象式时间线摘要的学习。发表于 IJCAI。[11] Xiuying Chen、Shen Gao、Chongyang Tao、Yan Song、Dongyan Zhao 和 Rui Yan。2018。通过润色迭代式学习文档表示以生成摘要。EMNLP（2018）。[12] Yen-Chun Chen 和 Mohit Bansal。2018。通过强化学习选取句子改写，实现快速抽象式摘要生成。ACL（2018）。[13] Jianpeng Cheng 和 Mirella Lapata。2016。通过抽取句子和词语实现神经摘要生成。arXiv 预印本 arXiv:1603.07252（2016）。[14] Hai Leong Chieu 和 Yoong Keok Lee。2004。沿时间线进行基于查询的事件抽取。发表于第 27 届 ACM SIGIR信息检索研究与发展年度国际会议论文集。425–432。[15] Kyunghyun Cho、Bart Van Merriënboer、Caglar Gulcehre、Dzmitry Bahdanau、Fethi Bougares、Holger Schwenk 和 Yoshua Bengio。2014。使用用于统计机器翻译的 RNN编码器-解码器学习短语表示。EMNLP（2014）。[16] Eric Chu、Prashanth Vijayaraghavan 和 Deb Roy。2018。通过注意力记忆网络从对话中学习人物角色。发表于 EMNLP。[17] John C. Duchi、Elad Hazan 和 Yoram Singer。2010。用于在线学习和随机优化的自适应次梯度方法。JMLR 12（2010），2121–2159。[18] Travis Ebesu、Bin Shen 和 Yi Fang。2018。用于推荐系统的协同记忆网络。发表于 SIGIR。[19] Günes Erkan 和 Dragomir R Radev。2004。LexRank：基于图的词汇中心性作为文本摘要中的显著性指标。人工智能研究期刊 22（2004），457–479。[20] Katja Filippova、Enrique Alfonseca、Carlos A Colmenares、Łukasz Kaiser 和 Oriol Vinyals。2015。使用长短期记忆网络通过删除进行句子压缩。发表于 2015 年自然语言处理实证方法会议论文集。360–368。[21] Jiyang Gao、Runzhou Ge、Kan Chen 和 Ram Nevatia。2018。用于视频问答的运动-外观协同记忆网络。发表于 CVPR。[22] Shen Gao、Xiuying Chen、Piji Li、Zhangming Chan、Dongyan Zhao 和 Rui Yan。2019。如何利用模式撰写摘要？通过原型编辑学习生成抽象式摘要。arXiv 预印本 arXiv:1909.08837（2019）。[23] Shen Gao、Xiuying Chen、Zhaochun Ren、Dongyan Zhao 和 Rui Yan。2020。电子商务问答中的有意义答案生成。arXiv 预印本 arXiv:2011.07307（2020）。[24] Daniil Gavrilov、Pavel Kalaidin 和 Valentin Malykh。2019。用于标题生成的自注意力模型。发表于欧洲信息检索会议。Springer，87–93。[25] Sebastian Gehrmann、Yuntian Deng 和 Alexander Rush。2018。自底向上的抽象式摘要生成。发表于 EMNLP。[26] Demian Gholipour Ghalandari 和 Georgiana Ifrim。2020。考察新闻时间线摘要的最新进展。发表于 ACL。[27] Jiatao Gu、Zhengdong Lu、Hang Li 和 Victor O. K. Li。2016。在序列到序列学习中融入复制机制。CoRR abs/1603.06393（2016）。[28] Wan-Ting Hsu、Chieh-Kai Lin、Ming-Ying Lee、Kerui Min、Jing Tang 和 Min Sun。2018。使用不一致性损失的抽取式与抽象式摘要统一模型。ACL，132–141。[29] Byeongchang Kim、Hyunwoo Kim 和 Gunhee Kim。2019。使用多层记忆网络对 Reddit 帖子进行抽象式摘要。发表于 NAACL。[30] Hayato Kobayashi、Masaki Noguchi 和 Taichi Yatsuka。2015。基于嵌入分布的摘要生成。发表于 2015 年自然语言处理实证方法会议论文集。1984–1989。[31] Ankit Kumar、Ozan Irsoy、Peter Ondruska、Mohit Iyyer、James Bradbury、Ishaan Gulrajani、Victor Zhong、Romain Paulus 和 Richard Socher。2016。有问必答：用于自然语言处理的动态记忆网络。ArXiv abs/1506.07285（2016）。[32] J Richard Landis 和 Gary G Koch。1977。分类数据中观察者一致性的测量。Biometrics（1977），159–174。

[33] Chenliang Li、W. Xu、S. Li 和 Sheng Gao。2018。基于关键信息引导网络的抽象式文本摘要引导生成。发表于 NAACL-HLT。[34] Jiwei Li 和 Sujian Li。2013。用于时间线摘要的进化层次狄利克雷过程。发表于 ACL，第2卷。556–560。[35] Jing Li、Aixin Sun、Jianglei Han 和 Chenliang Li。2018。深度学习在命名实体识别中的应用综述。arXiv 预印本 arXiv:1812.09449（2018）。[36] Mingzhe Li、Xiuying Chen、Min Yang、Shen Gao、Dongyan Zhao 和 Rui Yan。2021。吸引力的风格—内容二元性：通过解耦学习撰写吸睛标题。发表于 AAAI。[37] Chin-Yew Lin。2004。ROUGE：用于摘要自动评估的软件包。文本摘要的拓展（2004）。[38] Junyang Lin、Xu Sun、Shuming Ma 和 Qi Su。2018。抽象式摘要的全局编码。发表于 ACL。[39] Yang Liu 和 Mirella Lapata。2019。使用预训练编码器进行文本摘要。arXiv 预印本 arXiv:1908.08345（2019）。[40] Chao Ma、Chunhua Shen、Anthony Dick、Qi Wu、Peng Wang、Anton van den Hengel 和 Ian Reid。2018。基于记忆增强网络的视觉问答。发表于 CVPR。[41] Sebastian Martschat 和 Katja Markert。2018。用于时间线摘要的时间敏感子模框架。发表于第22届计算自然语言学习会议论文集。230–240。[42] Sameen Maruf 和 Gholamreza Haffari。2018。基于记忆网络的文档上下文神经机器翻译。发表于 ACL。[43] Rada Mihalcea 和 Paul Tarau。2004。TextRank：为文本构建秩序。发表于2004年自然语言处理实证方法会议论文集。404–411。[44] Rada Mihalcea 和 Paul Tarau。2004。TextRank：为文本构建秩序。发表于 EMNLP。[45] Alexander H. Miller、Adam Fisch、Jesse Dodge、Amir-Hossein Karimi、Antoine Bordes 和 Jason Weston。2016。用于直接读取文档的键值记忆网络。ArXiv abs/1606.03126（2016）。[46] Ramesh Nallapati、Igor Melnyk、Abhishek Kumar 和 Bowen Zhou。2017。Sengen：句子生成神经变分主题模型。arXiv 预印本 arXiv:1708.00308（2017）。[47] Ramesh Nallapati、Feifei Zhai 和 Bowen Zhou。2017。Summarunner：一种基于循环神经网络的序列模型，用于文档抽取式摘要。发表于 AAAI。[48] Ramesh Nallapati、Bowen Zhou、Caglar Gulcehre、Bing Xiang 等。2016。使用序列到序列循环神经网络及其他方法进行抽象式文本摘要。arXiv 预印本 arXiv:1602.06023（2016）。[49] Shashi Narayan、Shay B Cohen 和 Mirella Lapata。2018。利用强化学习对抽取式摘要中的句子进行排序。发表于 NAACL。1747–1759。[50] Romain Paulus、Caiming Xiong 和 Richard Socher。2018。用于抽象式摘要的深度强化模型。发表于 ICLR。[51] Juan Pavez、Hector Allende 和 Hector Allende-Cid。2018。工作记忆网络：通过关系推理模块增强记忆网络。发表于 ACL。[52] Pengjie Ren、Zhumin Chen、Z. Ren、Furu Wei、L. Nie、J. Ma 和 M. Rijke。2018。基于深度神经网络的抽取式摘要句子关系。TOIS 36（2018），1–32。[53] Zhaochun Ren、Shangsong Liang、Edgar Meij 和 Maarten de Rijke。2013。个性化时间感知推文摘要。发表于 SIGIR。ACM，513–522。[54] Alexander M Rush、Sumit Chopra 和 Jason Weston。2015。用于抽象式句子摘要的神经注意力模型。arXiv 预印本 arXiv:1509.00685（2015）。[55] Natalie Schluter。2017。根据 ROUGE 衡量自动摘要的局限性。发表于计算语言学协会欧洲分会第15届会议论文集：第2卷，短论文。ACL，41–45。[56] Abigail See、Peter J. Liu 和 Christopher D. Manning。2017。切中要点：使用指针生成网络进行摘要。ACL，1073–1083。[57] Xiaoyu Shen、Yang Zhao、Hui Su 和 Dietrich Klakow。2019。通过推广指针生成器改进文本摘要中的潜在对齐。发表于2019年自然语言处理实证方法会议暨第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集。3753–3764。[58] Julius Steen 和 Katja Markert。2019。抽象式时间线摘要。发表于第2届摘要新前沿研讨会论文集。21–31。[59] Sainbayar Sukhbaatar、Arthur Szlam、Jason Weston 和 Rob Fergus。2015。端到端记忆网络。发表于 NIPS。[60] Min Sun、Wan Ting Hsu、Chieh-Kai Lin、Ming-Ying Lee、Kerui Min 和 Jing Tang。2018。利用不一致性损失进行抽取式与抽象式摘要的统一模型。发表于 ACL。[61] Ilya Sutskever、Oriol Vinyals 和 Quoc V. Le。2014。神经网络的序列到序列学习。发表于 NIPS。[62] Chongyang Tao、Shen Gao、Mingyue Shang、Wei Wu、Dongyan Zhao 和 Rui Yan。2018。理解我的话语要点！利用多头注意力机制学习生成有效回复。发表于 IJCAI。4418–4424。[63] Chongyang Tao、Lili Mou、Dongyan Zhao 和 Rui Yan。2018。RUBER：一种用于开放领域对话系统自动评估的无监督方法。发表于 AAAI。[64] G. Tran、Tuan Tran、N. Tran、M. Alrifai 和 Nattiya Kanhabua。2013。在时间线摘要的优化框架中运用学习排序。[65] Kai Wang、Xiaojun Quan 和 Rui Wang。2019。BiSET：结合模板的双向选择性编码，用于抽象式摘要。发表于 ACL。

[66] Qinyong Wang、Hongzhi Yin、Zhiting Hu、Defu Lian、Hao Wang 和 Zi Huang。2018。采用对抗训练的神经记忆流式推荐网络。发表于 *KDD*。[67] Wenbo Wang、Yang Gao、Heyan Huang 和 Yuxiang Zhou。2019。用于抽象式摘要的概念指针网络。发表于*EMNLP*。[68] Wenjie Wang、Minlie Huang、Xin-Shun Xu、Fumin Shen 和 Liqiang Nie。2018。多聊一点：通过深度模型拓展聊天话题的深度与广度。发表于 *SIGIR*。ACM。[69] Chien-Sheng Wu、Richard Socher 和 Caiming Xiong。2019。用于面向任务对话的全局到局部记忆指针网络。发表于*ICLR*。[70] Caiming Xiong、Stephen Merity 和 Richard Socher。2016。用于视觉与文本问答的动态记忆网络。*ArXiv*abs/1603.01417（2016）。[71] Song Xu、Haoran Li、Peng Yuan、Youzheng Wu、Xiaodong He 和 Bowen Zhou。2020。自注意力引导的抽象式摘要复制机制。发表于 第*58*届计算语言学协会年会论文集。1355–1362。[72] Rui Yan、Liang Kong、Congrui Huang、Xiaojun Wan、Xiaoming Li 和 Yan Zhang。2011。通过进化式跨时段摘要生成时间线。发表于*EMNLP*。ACL，433–443。[73] Rui Yan、Ran Le、Yang Song、Tao Zhang、Xiangliang Zhang 和 Dongyan Zhao。2019。面试选择揭示你在市场中的偏好：通过用户画像记忆改进职位与简历匹配。发表于第*25*届*ACM SIGKDD*国际知识发现与数据挖掘会议论文集。914–922。[74] Rui Yan、Xiaojun Wan、Mirella Lapata、Wayne Xin Zhao、Pu-Jen Cheng 和 Xiaoming Li。2012。时间线可视化：通过文本与图像流之间的迭代强化实现进化式摘要。发表于*CIKM*。ACM，275–284。[75] Rui Yan、Xiaojun Wan、Jahna Otterbacher、Liang Kong、Xiaoming Li 和 Yan Zhang。2011。进化式时间线摘要：通过迭代替换实现的平衡优化框架。发表于*SIGIR*。ACM，745–754。[76] S. Yan 和 Xiaojun Wan。2015。基于深层依存子结构的多文档摘要学习。*TOIS* 34（2015），3:1–3:24。[77] Lili Yao、Yaoyuan Zhang、Yansong Feng、Dongyan Zhao 和 Rui Yan。2017。面向生成式短文本对话系统的隐式内容引入。发表于*EMNLP*。[78] Michihiro Yasunaga、Rui Zhang、Kshitijh Meelu、Ayush Pareek、Krishnan Srinivasan 和 Dragomir Radev。2017。基于图的神经多文档摘要。*arXiv* 预印本 *arXiv:1706.06681*（2017）。[79] Hainan Zhang、Yanyan Lan、Liang Pang、Hongshen Chen、Zhuoye Ding 和 Dawei Yin。2020。多轮对话生成的主题相关性建模。发表于*IJCAI*。[80] Ruqing Zhang、Jiafeng Guo、Yixing Fan、Yanyan Lan 和 Xueqi Cheng。2020。标题生成的结构学习。。发表于*AAAI*。9555–9562。[81] Xueliang Zhao、Wei Wu、Chongyang Tao、Can Xu、Dongyan Zhao 和 Rui Yan。2020。低资源知识驱动对话生成。发表于*ICLR*。[82] Xin Wayne Zhao、Yanwei Guo、Rui Yan、Yulan He 和 Xiaoming Li。2013。结合社交注意力的时间线生成。发表于*SIGIR*。ACM，1061–1064。[83] Ming Zhong、Pengfei Liu、Yiran Chen、Danqing Wang、Xipeng Qiu 和 Xuanjing Huang。2020。将抽取式摘要视为文本匹配。*arXiv* 预印本 *arXiv:2004.08795*（2020）。[84] Xiao Zhou、Cecilia Mascolo 和 Zhongxiang Zhao。2019。主题增强记忆网络：个性化兴趣点推荐。发表于 *KDD*。[85] Junnan Zhu、Y. Zhou、Jiajun Zhang 和 Chengqing Zong。2020。注意、翻译与摘要：一种高效的神经跨语言摘要方法。发表于 *ACL*。