arXiv:2301.08277v2 [cs.DL] 9 Apr 2023

LaTeX、元数据与出版工作流程（更新于2023年4月3日）

Joppe W. Bos[1] 和 Kevin S. McCurley[2]

[1] 恩智浦半导体[2] 自身

摘要。LaTeX 所服务的科学出版领域，越来越依赖出版物相关元数据的可用性。我们探讨如何使用 LaTeX 类和 BibTeX 样式，在已发表文章的整个生命周期中整理元数据。我们的重点是简化并自动化出版工作流程中的诸多环节。我们综述现有方法的各种选择及其缺点，并概述我们在一种新 LaTeX 样式文件中的实践方法，其主要目标是让作者只需指定一次元数据，便可在整个出版流程中使用。我们认为，这有助于降低出版成本，因为它减少了编辑和提供出版物元数据所需的人工工作量。

关键词： 出版 · LaTeX · 元数据

1 引言

TEX 最初的目标主要侧重于排版，以及输出内容在纸面上的呈现效果。后来，LATEX 的发明旨在“让用户专注于文本的结构，而非格式设置命令” [Lam86]。我们鼓励用户使用诸如 `\section` 这样的高级宏来撰写论文，并将章节前后留多少空白等决定交由所使用的样式处理。因此，作者不必过多担心论文的外观，只需关注论文的逻辑结构。这种“关注点分离”是计算机科学中的一个更普遍概念：程序会被拆分为不同的单元，每个单元专注于程序任务中受限的一部分。

将外观与结构分开关注已被证明非常有效；如今，大多数（如果不是全部）科学出版商都拥有自己的 LATEX 样式（Overleaf 列出了约 400 种学术期刊样式）。[1] 这些样式让作者能够轻松遵循期刊统一的外观风格。如果作者遵守样式要求，就能简化期刊的制作流程。

LATEX 社群在某个方面迟迟未能适应现代出版工作流程的需求，那就是对出版物元数据的整理。这是本文的重点。其中一些元数据相当直观，例如标题、作者名单、所属机构、资助来源和参考文献。大部分元数据应由作者提供，因为他们最了解不同实体之间的关系。我们认为，LATEX 文档类应提供便捷机制，让作者只需输入一次元数据，并以能够体现实体间关系的方式进行编码，以便该 LATEX 文档类生成适当的机器可解析格式，供出版流程的各个阶段使用。

电子邮箱： joppe.bos@nxp.com (Joppe W. Bos)， iacrcc@digicrime.com (KevinS. McCurley)[1]参见 https://www.overleaf.com/latex/templates/tagged/academic-journal

本文综述了元数据的各种要求，以及可用于方便地收集这些信息、减轻作者负担的方案。我们概述了一些 LaTeX 引擎的可能性与缺点，并介绍我们在一个新的 LaTeX 文档类文件中解决这一问题的方法。

2 元数据的重要性

在竞争日益激烈的学术出版领域，元数据的重要性日益凸显，原因有以下几点：

- 元数据为我们浏览学术文献提供了重要的分类体系。
- 作者、机构和资助机构都需要获得恰当的归属判定。
- 出版物数量激增，进一步凸显了对出版物、作者、大学和院系进行排名的必要性。衡量科学影响力并对出版物进行排名的主要方法是引文分析，用以确定哪些出版物对某一科学领域影响最大。研究人员自身也需要将注意力集中在某一领域最重要的成果上。质量指标虽不尽完善，但仍能发挥作用。
- 论文往往会以不同形式发布在网络上的多个地方，因此重复项消除是个棘手的问题。出版商往往会让问题变得更严重：他们试图将“正式出版”的版本置于付费墙之后，同时允许作者在个人主页和电子预印本服务器上自由发布非正式版本。

如今，众多索引服务都在追踪学术出版物，包括 Google 学术、科睿唯安（科学引文索引）、Scopus、PubMed 等。由于出版物中的元数据状况不佳，这项工作变得复杂。从出版物中推断元数据本身就是一个复杂的主题 [TXM+20]；如果出版物附带以机器可解析格式提供的元数据，这个问题就能得到简化。这正是我们开展这项工作的目标之一。

3 出版工作流程

我们的动机部分源于参与创办一个面向专业非营利组织国际密码学研究协会（IACR）的新开放获取期刊。[2] 该协会已经运营着两本钻石开放获取期刊，但运营经验表明，平均而言，每篇已发表论文都需要投入约一小时的人工工作来完成制作和元数据处理。

另一项研究 [GB21] 估计，编辑和制作每篇已出版论文所需的人工劳动为 7.5 个工时。我们认为，其中大部分工作本应且可以实现自动化，这有助于降低出版成本。这对于开放获取出版尤其重要，因为开放获取出版严重依赖志愿劳动 [BFK+21] 来控制成本。它还可以用于提高商业出版商的盈利能力。

出版工作流程因组织而异，但至少包括以下步骤：创作、投稿、评审、修订、提交最终版本、文字编辑、制作可读的输出格式（如 PDF 和 HTML）、DOI 注册、编目和托管。即使在评审阶段，元数据也可能十分重要，因为它可用于自动识别潜在的利益冲突

[2] 参见 https://iacr.org.

。对于 2022年NeurIPS会议这类大型活动，其收到 9,634 篇完整论文投稿，并有 10,406 名审稿人，[3] 元数据有助于自动分配审稿人，并避免利益冲突。

在一些系统（如开放期刊系统）中，[OJS], 元数据的投稿和整理被视为与 Word、LaTeX 或 PDF 文档的投稿相分离的任务。这给作者增加了额外负担，也使工作流程容易因元数据存在于两个位置而出现不一致。据我们的经验，文章经过修订并获接受后，标题、摘要、所属机构、电子邮件地址等往往会发生变化。检查并修正这些不一致，最终会耗费作者和编辑的人工时间。

当作者提交最终版本时，元数据在出版工作流程中的重要性日益凸显。最终版本准备就绪后，通常需要为其注册 DOI，而这要求提供大量元数据。此外，论文的网络“落地页”通常也必须根据元数据创建。随后，索引机构会介入，要么抓取数据，要么通过出版商提供的元数据馈送获取数据。

4 我们的整体方法

我们通过使用 LaTeX 类和 BibTeX 样式 `iacrcc.bst`.[4]，在出版工作流程中自动采集元数据。

- 作者按照惯常方式，使用提供的 `cls` 文件编写自己的 LaTeX。几乎所有元数据都由作者通过类文件中的结构化宏提供。`cls` 和 `bst` 文件的作用，既是在输出格式中显示元数据，也是在编译过程中提取元数据，并将其作为副产品生成易于解析的外部格式。

- 作者提交最终版本时，会将 LaTeX 源文件上传到云服务器，由服务器编译源文件，并从中提取所有元数据，保存为结构化格式文本文件。投稿过程无需作者输入任何额外的元数据，因为所有信息都已编码在 LaTeX 源文件中。DOI 后缀由服务器分配，投稿时会将 DOI 直接编译到 PDF 中（DOI 稍后会在 DOI 注册机构注册）。

- 我们使用后编译步骤来解析结构化元数据，并将其转换为其他格式，包括 JSON 和 XML。
- 论文经编辑批准后，提取出的元数据将进一步用于注册 DOI。提取出的元数据还用于为期刊网站生成各种网页、RSS馈送和 OAI-PMH馈送，并向各种索引服务注册。

我们要求输出的元数据必须是文本，而文本编码的通用标准是 UTF-8。除了标题或摘要中的行内公式等数学结构之外，文本中不应包含 TEX 宏。这在 LaTeX 环境中会带来一些问题。

在我们的讨论中，区分 TEX 世界中的文本输入编码与文本输出编码非常重要。传统上，TEX 的设计只接受 7位ASCII 输入，并使用 |verb|ü| 之类的方式来编码带重音符号的字符。如今，所有 TEX 引擎都能够处理 UTF-8输入编码，但对输出编码的支持则复杂得多。这里的问题在于字体，而字体当然是必不可少的，因为

[3]参见 https://media.neurips.cc/Conferences/NeurIPS2022/NeurIPS_2022_Fact_Sheet.pdf。 [4] 下载这些内容的权威来源是 https://publish.iacr.org/iacrcc。

排版。TEX 引擎通常专注于使用字体和断词生成排版内容，这使得所有关于 TEX 输出编码的讨论都变得复杂。LATEX 还有一种称为“LICR”的字符内部表示。对此的完整讨论超出了本文的范围，但更多详情请参见 [MFL16]和 [MGB+04, § 7] 。

像 `# $ % & _ ^ ~ \ { } %` 这样的字符和空格需要作者特别处理才能将其编码为文本，因为它们对 TEX 解析器有特殊含义。当 LATEX 读取输入文件时，文件内容会转换为一系列词元，其中一些词元表示文本字符。这种内部表示方式使得访问通常在其他任何计算机语言中都会被视为“文本”的内容变得不便。此外，LATEX 的核心功能是用户定义的宏，因此作者可能会定义`\pe`来表示文本字符串“Paul Erdős”。只有在 LATEX 输出过程中，宏展开为词元时，我们才能发现这一点。TEX 展开过程非常复杂，并且有许多特殊情况。

5 元数据采集

我们考虑了几种从作者的 LATEX 源文件中提取元数据的方法。一种方法是使用高级编程语言编写的 LATEX 解析器，直接从 LATEX 中提取元数据。解析 LATEX 的难点在于需要展开宏，而迄今为止，LATEX 引擎本身是唯一可靠的解决方案。我们考虑的另一种方法是在 `lualatex` 中使用 `lua` 。与使用 LATEX 本身相比，这种方法更适合文本处理，但我们最初的目标是尝试让各种 LATEX 引擎都能正常工作。我们尝试的第三种方法是使用 `\write` 宏来采集元数据并将其写入文件。

5.1 我们首次尝试使用 `\write`

`\write` 宏的作用是展开一组词元，并将这些词元可解析的表示形式写入文件。它最初是为写入辅助文件而设计的，例如 `.bbl`、 `.aux` 等；这些文件会在后续编译过程中读入 LATEX 源文件。 `\write` 会执行展开，这一点对我们非常有用，因为它会展开用户定义的宏。遗憾的是，我们发现使用 `\write` 生成元数据时会遇到一些问题。

1. ~ 字符会被 `\write` 展开为以下不易阅读的词元表示形式：

   ```
   \protect \unhbox \voidb@x \protect \penalty \@M \ {}
   ```

   从元数据的角度来看，它应该被替换为不换行空格；这种空格几乎存在于所有字符编码中。

2. 展开过程有时会产生多余的空格，例如， `$\alpha$` 会变成`$\alpha $`。更严重的是，如果我们将 `$` 数学定界符改为 LATEX 版本`\(` 和 `\)`，那么 `\write` 就会彻底失败，而且部分错误信息会写入输出文件。还有其他一些内容无法作为参数传递给 `\write`，包括带有可选参数的宏。

3. 表达式 `$\frac{1+x}{2-x}$` 会展开成可读性较差的版本

   ```
   $\protect {\begingroup 1+x\endgroup \over 2-x}$
   ```

4. LaTeX 中的文本处理与排版混为一谈，因此输出中可能会出现混合编码。在排版过程中，这种情况作为中间步骤是可以接受的，因为 TEX 会使用采用不同编码的字体，将其转换为页面上的字形。但遗憾的是，对于处理来自 `\write` 的元数据而言，这种情况不可接受。如果我们使用 `\write\f{Ð and \DJ\ and ü and \"u}` ，那么 `lualatex` 的输出由以下内容组成：

   `Ð and \TU\DJ \ and ü and ü`

   因为它使用 Unicode 编码，但 pdflatex 的输出由以下内容组成：

   `Ð and \T1\DJ \ and ü and`

   （ `\DJ` 宏要求我们使用 `\usepackage[T1]{fontenc}`）。遗憾的是，这个来自 `\write` 的输出包含采用混合编码的字符：Ð 采用 UTF-8 `C390` 编码，ü 的编码也正确，但 `\"u` 采用的是 `FC` 的 T1（Cork）编码，而不是 `C3BC`的 UTF-8 编码。

其中许多问题都可以在用高级语言编写的流水线后处理步骤中解决，但我们无法接受输出文件中存在多种字符编码。此外，这些仅仅是一些示例，说明我们可能会在作者提供且能够正确编译的 LaTeX 中遇到哪些情况。我们的目标是提供一个系统，支持作者提供的任何内容，并向他们清楚说明如何准备这些内容，从而避免中断（错误）或给作者的排版体验带来其他不便。最终，作者以文本形式编码的元数据，无论是作为 TEX 宏还是 UTF-8 输入，都应能以 UTF-8 输出的形式保留下来。

我们收到的一条建议是尝试使用 `expl3` 宏 `\text_purify:n`，我们发现它有助于清理一些内容，例如移除 `\texttt`。遗憾的是，我们还发现它会执行一些破坏性操作，例如彻底删除某些宏，如 `\dag`、 `\copyright`和 `\pounds` ；这些宏会生成 †、© 和 £，其 UTF-8 编码显然分别是 U+2020、U+00A9 和 U+00A3。我们也无法从文档或源代码中弄清楚这个函数的行为，因此决定不再使用它。也许其他人会觉得它有用。

还有其他需要对文本进行编码的 LaTeX 宏包。一个很好的例子是 `hyperref` 宏包，它需要生成采用 PDF 支持的文本编码格式的 PDF 书签。PDF 最初只支持 PDFDocEncoding 文本编码。自 1996年 起，PDF 开始支持 UTF-16BE 编码，也就是在字符串开头带有字节顺序标记的 UTF-16 编码。UTF-16 编码与 UTF-8 大不相同，因为它为每个字符使用一个或两个 16 位块，而 UTF-8 则使用 1 到 4 个字节来表示字符。从 2017年 开始，PDF 2.0 允许文本以 UTF-8 编码；但截至 2022年， `hyperref`宏包仍只支持 UTF-16BE（PU）和 PDFDocEncoding 文本格式。其 Unicode 实现依赖于一个有 2000 行的文件，其中的行类似于

```
\DeclareTextCompositeCommand{\‘}{PU}{u}{\80\371}% U+00F9
```

用于定义带重音符的拉丁小写字母 U ù 的编码。构建这样的东西是个繁琐的过程。

另一个需要进行文本处理的 LaTeX 宏包示例是 `pdfx`该宏包会在由 LaTeX 生成的 PDF 输出中产生 UTF-8 编码的 XML。它们定义了一种名为 `l8u` [RTMS（第 4 节）的伪编码。2]。他们提到，这种编码可以重新打包成一个完整的宏包，用来生成 UTF-8 编码的字符串，这对我们或许会有帮助。该 `pdfx` 宏包的用途与我们的类似，因为他们的目标是将部分元数据注入 PDF。我们的目标则是从 L 提取更多元数据ATEX 源文件（例如 ORCID 标识符）。

5.2 切换到 `\protected@write`

在本论文的早期版本发布后，我们收到了审稿人的意见，建议使用 LaTeX 宏 `\protected@write`，而不是 TEX 宏 `\write`。这为我们解决了不少问题。例如，`\(` 可以在 `\protected@write` 中使用，也解决了使用 `pdflatex` 时字符编码混杂的问题。

遗憾的是，使用 `\protected@write` 会引入一些其他问题，例如将 `\frac{x}{2}` 展开为 `\protect␣\frac␣␣{x}{2}`。幸运的是，我们并不需要输出文件只包含 UTF-8 字符，因为我们可以轻松地使用更高级语言（如 Python）编写的后处理步骤，去除多余的 `\protect` 宏，或将类似 `\"u` 的内容转换为 ü；我们现在采用的就是这种方法。其他作者提供的 LaTeX 还会带来哪些问题，仍有待观察。

6 需要哪些元数据?

与出版物关联的元数据可能很多。有些字段相当明显，但即使是显而易见的字段，在编码方式上也有细微差别。例如：

- 作品标题。在某些领域，标题中使用数学内容很常见，但元数据记录中的 TEX 格式通常会转换为其他格式，例如 MathML。标题也可能包含字体样式标记（如粗体）或多种字符集。特别长的标题有时会分层拆分，并包含副标题或用于页眉标题的简短版本。一篇文章可能有不同语言的标题。

- 作品作者。仅仅询问作者姓名也可能很棘手，建议读者阅读 [McK10]。通常最好的做法是提供一个用于填写姓名的字符串字段，但某些书目格式可能要求填写姓氏和/或名字。

- 作者的贡献程度可能各不相同。有时会通过将作者姓名按非字母顺序排列来表示这一点，但在其他一些领域，通常会为作者贡献指定 角色 。人们常使用 CRediT 分类体系来反映这种 [NIS]。作者也可能被归类为“通讯作者”。

- 作者与所属机构之间和/或作者与资助机构之间的关系。如今，作者拥有多个所属机构 [HRL21]，且多位作者共享部分所属机构或资助机构的情况十分常见。对于这些多对多关系，最好将其编码为关系，而不是为每位作者重复填写信息。

- 书目信息（例如期刊或会议名称、卷、年份等）。
- 参考文献列表（由 BibTEX 提供）。
- 许可信息。

还有许多其他字段可以编码到 LaTeX 文档中，或编码到由 LaTeX 生成的输出格式中。例如：摘要、页数、作者的地址信息、作者的电子邮件地址，以及指向代码和数据等辅助作品的链接等。本文档不打算罗列所有这些字段，但已有许多尝试旨在为出版元数据定义模式。我们在此提及这些模式，是因为读者可能需要遵循其中一种或多种模式。

元数据当然可以编码为多种格式，例如 LaTeX 本身、XML、JSON 或 YAML。大多数出版物元数据描述都采用各种

XML 方言。我们认为，作者提供的元数据应编码在原始 LaTeX 中，并且对这些元数据的处理应能够生成出版流水线所需的其他格式。

下面我们介绍几种最重要的元数据模式。

6.1 Crossref

crossref.org 是一家非营利组织，其主要使命是收集元数据并分配 DOI 标识符。因此，它是最重要的兼容格式之一，并且拥有自己的 XML 模式 [Croa]。其模式支持多个所属机构、作者角色和资助机构。

6.2 JATS

期刊文章标签套件（JATS）是一种 XML 格式，包含三种变体，分别用于归档与元数据、出版和创作 [Usd16，Nat21a]。JATS 格式可视为出版物完整的结构表示；在许多方面，它类似于 LaTeX，但更注重结构而非版面布局。JATS 文档由若干部分组成，包括：

前置部分 这是一个必需部分，文章的大多数重要元数据都会在此出现。其中包含文章的元数据，也包含其所属连续出版物或期刊的相关元数据。

正文 这是一个可选部分，用于包含文章本身的主要内容，由各个章节、段落等组成。

后置部分 这是一个可选部分，包含书目参考文献、术语表、附录或文章的其他辅助信息。JATS 的一大优势在于能够封装书目参考文献。

前置部分的模式表达能力很强，并且提供了我们所见过的最广泛的出版元数据覆盖范围。JATS 可以同时容纳用于数学的 MathML 3 和行内 TEX，因此也可以被视为 LaTeX 的竞争者，或被视为第 8 节所定义的输出格式。不过，要将 LaTeX 的全部功能转换为 JATS 相当困难，而且由于 JATS 只包含语义结构而不包含布局信息，转换结果往往无法忠实体现作者的原意。要将 JATS 转换为便于使用的格式，需要有与 LaTeX 样式相对应的机制，即 XSLT 或其他 XML 格式化手段。

6.3 Elsevier 的 Scopus

Scopus 是规模最大的索引机构之一。作为一项商业服务，他们与出版商合作以获取其内容，但他们更倾向于接收 JATS 格式的 PDF 和 XML [Bre22]。

6.4 科睿唯安 Web of Science

科睿唯安的科学引文索引是另一项商业服务，提供索引服务。该服务接受采用其自有模式的 XML 数据源 [Cla21]。

6.5 其他格式

还有许多描述性较弱的元数据格式，但它们不够全面。我们提及这些格式，部分原因是它们可能定义可用于其他 XML 格式的命名空间。

开放获取期刊目录 开放获取期刊目录是开放获取期刊的简称。这是一个由社区维护的在线目录，用于索引开放获取的同行评审期刊。它们接受符合自身模式或 Crossref 模式的 XML [DOA22]。

可扩展元数据平台（XMP）这是用于表示与数字文档相关联的元数据的 ISO 标准。它最初由 Adobe Systems 创建，常用于 PDF。该标准定义了若干“标准”模式，但也可通过纳入其他 XML 模式进行扩展。这些标准模式缺少许多描述学术文献所需的基本功能。

都柏林核心都柏林核心数据集 1.1 版于 1999 年推出，是一种仅包含十五个基本元素的 XML 模式。事实证明，它不足以描述学术文章，但其中一些元素仍用于其他 XML 模式，而且它也是 XMP 的标准模式之一。2008 年推出的“限定都柏林核心”改进了 1.1 版，但仍缺少 ORCID 标识符或所属机构等基本信息。书目参考文献以非结构化字符串的形式编码，在都柏林核心 1.1 中则完全省略。

PRISM PRISM [Ken] 是另一种简单的 XML 格式，用于描述作者、标题、URL、DOI、日期、权利等。它的使用似乎不如 JATS 广泛。该 `hyperxmp` 软件包会在 PDF 中生成 XMP，并包含此模式中的一些标签。

TEI 文本编码倡议（TEI）是一种较早的基于 XML 的格式，始于 1987 年。它主要在人文学科中用作 JATS 的替代格式，但描述能力弱得多。

6.6 关于元数据模式的结论

为期刊和／或会议系列开发 LaTeX 类时，了解与该领域最相关的元数据元素非常重要。由于我们的期刊属于计算机科学、工程学和数学领域，因此不需要临床试验信息或化学反应编码之类的内容。开发一个支持出版物所有可能元数据的完整 LaTeX 软件包将是一项艰巨的工程，也会给作者带来不必要的负担，让他们必须逐一了解所有可能的元数据元素。

我们的工作流程支持向 Crossref 注册 DOI，并由多项服务进行编目。遵循这些模式会对文章、作者、所属机构和资助机构之间的关系施加一些限制。这些限制如图所示 1。我们认为，这些限制对所有学科都相当普遍，但仍有许多潜在的扩展方式。作者或所属机构等实体可以通过实体本身的属性，或实体间关系的属性进行扩展。例如，作者可以列出所属机构，同时说明完成这项工作时自己正在休假。这可以作为作者与所属机构之间关系的属性。作者也可以被标记为已故，并将此作为作者的属性。JATS 模式中描述了许多此类属性，有些属性通过特定脚注类型的脚注关联。我们将这些内容留待未来工作处理。

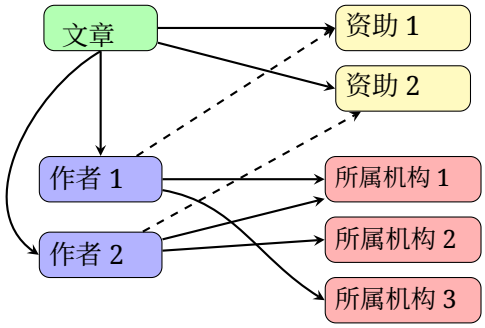

图 1：主要实体之间的关系。每个实体在文章中只列出一次。一篇文章可以有多位作者，他们与所属机构之间存在关联。在 Crossref 模式中，资助机构与文章相关联，因此我们选择以这种方式将它们关联起来。另一种方式是，使用虚线箭头表示的关系可以将作者与其资助来源关联起来，与我们将作者与所属机构关联起来的方式大致相同。我们选择使用脚注来说明资助机构与作者或所属机构之间的复杂关系。一些资助机构（例如 [Nat21b]）对论文中这些注释的呈现方式有严格规定。

7 使用唯一标识符

遗憾的是，标题和人名之类的信息并非唯一标识符。DBLP 列出了 14 位在计算机科学领域使用完全相同姓名“Thomas Müller”的作者，以及几十位姓名与之相似的其他作者，例如 Thomas F. Müller。为了开展大规模文献计量分析，以进行归属判定或重复项检测，与出版物相关联的所有实体都需要分配唯一标识符。

许多 XML 模式（如 JATS）都采用了唯一标识符。分配唯一标识符的主要举措包括：

- 出版物的 DOI 标识符 [Pas10]，
- 作者的 ORCID 标识符 [HFP+12]，
- 科研机构的 ROR 标识符 [Lam20]，
- 资助机构的 Crossref 资助机构注册库 [Crob]。

请注意，在每种情况下，当某个组织为某个实体分配了唯一标识符时，通常还会有其他组织拥有各自的标识符空间。例如，DOI 需要付费，因此有些组织会在自己的命名空间内为文章分配标识符。例如，`arxiv.org` 长期以来一直在其命名空间内为论文分配唯一标识符。最近，他们也开始分配 DOI，以编码其沿用已久的标识符系统。

有时，同一篇文章的两个几乎完全相同的版本会在同一命名空间中被分配两个 DOI（例如，出版商版本和 arXiv 版本）。这并非唯一一种会为论文的细微差异版本分配多个 DOI 的情况。`arxiv.org` 会为每次修订分配不同的 DOI，无论改动多么微小。

ORCID 标识符实际上是 ISNI 标识符中预留的一段，但 ISNI 也向一些作者颁发了标识符[5]，这些作者没有 ORCID 标识符。ISNI 还在其他领域维护标识符，包括媒体创作者、出版商和一些学术机构[6]

[5]例如，请参见 https://isni.org/isni/0000000088556337。 [6]例如，请参见 https://isni.org/isni/0000000121924307。

作为 ROR 标识符的替代方案。追踪书目信息的商业服务通常会为作者和/或机构隶属标识符维护自己的命名空间。例如，科睿唯安 Web of Science、Scopus、SciENcv、《数学评论》和 DBLP 都曾为作者颁发其他标识符。

ROR 标识符的粒度较粗，因此虽然麻省理工学院有对应的标识符，但这些标识符无法区分该大学的院系、学院或项目。相比之下，《数学评论》会为院系分配机构代码（例如，南加州大学计算机科学系的代码为 1-SCA-C）。

与学术出版物相关的标识符完整列表超出了本文档的讨论范围，而且我们或许应该预期未来会出现新的标识符系统。由于一个实体可能拥有来自不同组织的多个 ID，我们强烈建议采用一种模式，为 ID 分配命名空间，并在该命名空间中指定标识符。例如，关于 `arxiv.org` 的论文可能在 DOI 命名空间中有一个标识符，同时也在 arxiv.org 命名空间中有一个标识符。支持纳入来自多个命名空间的标识符，有助于消歧。

7.1 关于标识符的主要要点

学术文章可能关联的元数据元素有很多，而且这个列表还在不断增长。只要以尽可能细的粒度收集数据，就应该能够将其重新格式化为所需的任何 XML 模式。我们认为，JATS 格式（参见第6.2节）最适合描述出版物的元数据，因此我们设计的元数据宏与 JATS 保持一致。

出版界中的不同参与者会有不同的必需元素和可选元素。为了最大限度地满足元数据下游使用方的要求，尽可能收集更多信息非常重要。LaTeX 样式往往是为特定期刊、学会和会议编写的，因此只需涵盖其学科中最常见的元数据元素（例如，临床试验数据与数学无关）。最后，您可能需要预料到，未来可能需要扩展该模式。

8 输出格式

出版工作流程中的一个重要步骤，是创建供人工阅读的各种格式（例如 PDF 和 HTML）。在可能的情况下，最好以机器可读的方式将元数据嵌入这些输出格式中，使元数据随可阅读的文档一同提供。遗憾的是，用于此目的的标准通常缺乏全面性。

8.1 在 HTML 中嵌入元数据

HTML 对嵌入元数据的规范非常不完整。最初的 `<meta>`tag 被设计用于元数据，但这是一种扁平结构，而最初的 `name` attribute 仅涵盖学术出版物中相对较少的元数据元素。人们提出了 HTML 的各种扩展，包括用于`<meta>` 标签名称的 whatwg list[7] 。遗憾的是，这仍然缺少我们所需的许多元素。

另一种方法是在 `<script type="application/xml">` 中编码 XML。如果你的 XML 不包含 `</script>`，这种方法就行得通。另一种选择是使用

[7]请参阅 https://wiki.whatwg.org/wiki/MetaExtensions。

```
<link itemprop="meta" src="xmlurl">
```

or

```
<link rel="alternate" src="xmlurl">
```

在文档的 `<head>` 中。这两种方法都不是标准做法。

网页中另一种用于书目信息的模式由学术文章格式提供[8]，来自 `schema.org`。由于多种原因，这种格式并不完善，其中包括它不包含 ORCID 标识符这一事实。理论上，可以扩展学术文章格式来弥补模式的不足，但他们并未这样做，而且这种格式似乎很少被使用。

`schema.org` 的元数据通常会作为 `JSON-LD` 嵌入网页中，位于 HTML 页面的 `<head>`中

```
<script type="application/ld+json">
```

一种颇有前景的方法是以这种方式嵌入 JATS，即将 XML 转换为 JSON。遗憾的是，目前没有标准方法可以将 XML 转换为 JSON。显然，要使这种方法实用化，就需要实现标准化。

8.2 在 PDF 中嵌入

虽然我们可以预期，未来其他便于使用的文档格式也会越来越流行，但如今 LaTeX 主要用于生成 PDF。至少有两种方法可以将元数据嵌入 PDF，即文档信息字典和 XMP。文档信息字典自 PDF 1.0 版起便已存在，但只能编码一些非常基本的元素，例如标题、作者、主题、关键词、创建日期等。这是 `hyperref` 软件包的输出，但这种模式无法满足出版的需求。尤其是，它只有一个作者字段，也无法嵌入 DOI 或 ORCID 标识符等唯一标识符。鉴于这些不足，大多数元数据都应使用 XMP 编码，下一节将对此进行介绍。

8.2.1 XMP

从 PDF 1.4 版开始，将元数据嵌入 PDF 的首选方式是使用“元数据流”（Metadata Streams），这些元数据流本身是称为数据包的 XML 文档。这种方式称为可扩展元数据平台（XMP） [XMP]。XMP 标准定义了若干标准模式 [pdfb], ，但其中似乎只有最精简的都柏林核心模式 1.1 与学术出版相关，而我们此前已认为它并不够用（参见第 6.5 节）。幸运的是，顾名思义，这种格式具有可扩展性，XML 字典可以使用来自多个命名空间的模式 [pdfc]。该 `hyperxmp` 软件包已经为少数元数据元素采用了这种做法。

我们认为，XMP 元数据的使用仍有很大的改进空间，包括书目参考文献、ORCID 标识符、所属机构、资助信息等内容的编码。遗憾的是，XMP 扩展机制要求，只要使用了任何扩展模式，就必须将整个扩展模式嵌入 PDF 文档中。因此，PDF 文件中的模式信息可能会多于实际元数据。

8.2.2 PDF/A 与归档

PDF/A 是一组 ISO 标准 [pdfa]，用于定义 PDF 的一个子集，旨在对电子文档进行长期归档。这些标准定义了不同的版本和不同的

[8]请参阅 https://schema.org/ScholarlyArticle。

合规级别。PDF/A-1 基于 PDF 1.4 版；PDF/A-4 基于 PDF 2.0 版。PDF/A-3 增加了文件附件功能。用于元数据的 XMP 在 PDF/A-1 中引入，并且是 PDF/A-2 的一项要求。遗憾的是，由于多种原因，生成符合某一 PDF/A 标准的 PDF 文件可能很困难：

1. 可用于验证 PDF/A 合规性的开源工具很少，这或许部分是因为相关标准属于 ISO 专有文档，并且无法免费获取。可能最好的开源工具是 veraPDF，但它仍处于测试版。2. 字体必须嵌入 PDF 中。LaTeX 引擎通常会自动完成此操作，但作者应注意，如果文档中包含 PDF 图形，其中可能含有未嵌入的字体。3. PDF 中必须包含色彩空间。如果 PDF 包含使用不同色彩空间的彩色图形，这可能会造成问题，因为 PDF/A-1 只允许使用一种色彩空间。4. 某些 PDF 功能是禁止的，例如透明度、加密、嵌入式音频和 JavaScript。

XMP 元数据的一项要求是，其中包含的信息必须与文档信息字典中的数据一致。有人指出，这可能会造成问题，因为文档信息字典没有明确说明如何编码多个作者姓名 [Pak, 第 13]。

美国国家科学基金会、欧盟委员会以及 cOAlition S 成员等资助机构如今要求，将由其资助的研究出版物存入经批准的知识库，以便进行长期数字归档。具体而言，美国国家科学基金会现要求其所有资助研究的成果均以 PDF/A 文件形式存档 [Nat]。不同知识库对存档的要求各不相同，但通常都支持 PDF/A 和 JATS。

值得注意的是，有人质疑 PDF/A 的档案保存价值 [Kli17]。

8.2.3 LaTeX 的 XMP 和 PDF/A 宏包

请注意， `hyperref` 宏包包含一个用于生成 PDF/A 的选项，但该选项只会禁用一些违反 PDF/A 规范的功能，并不能处理 XMP 元数据。有几个 LaTeX 宏包专门用于将 XMP 数据注入 PDF 文件和/或生成符合 PDF/A 规范的输出，其中包括 `hyperxmp`、 `xmpincl` 和 `pdfx`。这些宏包的目标、方法和结果各不相同。 `hyperxmp` 与 `pdfx` 不兼容。另请注意， `pdfx` 需要借助 `luatex85` 宏包才能与 `lualatex` 配合使用。

对于作者而言，最简单的软件包是 `hyperxmp`。它会修改`hyperref` 软件包的行为，以便在 PDF 中生成 XMP 元数据。 `acmart` 使用它来提供 XMP，而 `hyperxmp` 软件包则与 `acmart` 配合使用，以智能推断部分元数据。遗憾的是， `hyperxmp` 存在一些问题：

1. 我们发现，在使用 `verapdf` 软件包时，某些输出无法通过 PDF/A 验证，即使采用相对宽松的 PDF/A 1-b 标准也是如此。2. 它仅限于一组预先确定的模式，其中包括基本模式，同时还添加了 PRISM 和其他模式。它遗漏了 ORCID 标识符、所属机构、ROR 标识符、参考文献以及其他我们认为理想的元素。

该 `pdfx` LaTeX 包 [RTMS] 旨在为生成符合 PDF/A 标准的文档提供支持，因此也支持将 XMP 元数据注入 PDF。它支持通过将 L 映射到 XML 元素来生成嵌入式 XMP-TEX 宏

，例如 \Title 生成 <dc:title> 和 \Volume 生成<prism:volume>。与 hyperxmp不同， pdfx 包要求作者在外部文件 \jobname.xmpdata中指定元数据元素。使用这些宏，可以确保文档信息字典中的数据与 XMP 数据一致。

虽然 hyperxmp 和 pdfx 都通过定义用于指定元数据元素的宏来支持 XMP，但 xmpincl 软件包提供了一种更灵活的方案。 xmpincl要求你创建原始 XML/RDF 文件，然后将其包含进来。这给用户带来了相当大的负担，但比 hyperxmp更加灵活。pdfx 软件包使用 xmpincl 来包含 XMP。xmpincl目前已没有活跃的维护者，但该软件包本身相当简单。

值得一提的是，LaTeX 项目正在开展一项为期多年的带标签 PDF 输出项目，该项目将定义一个新的元数据接口 [MF]。这将允许作者指定一个 pdfstandard 标志，以纳入用于 PDF/A 合规性的色彩配置文件。该功能正在被纳入 LaTeX [l3p], 但这仍无法解决如何确保所包含的图形符合所声明色彩标准的问题。LaTeX 团队正在开展一项长期的“带标签 PDF 项目”，其目标之一是更好地支持 XMP [FM22]。

9 \author有害论

现在我们来讨论如何将元数据嵌入原始 LaTeX 源文件。原始的 LaTeX 对 \author 的定义，几乎无助于记录作者元数据，而且在显示大量作者时也存在问题。在标准的 article 类中，作者将 \author 定义为包含若干格式化文本块，并以 \and 分隔。因此，例如，没有标准方法可以将 ORCID 标识符与作者姓名关联起来，也无法将所属机构或资助机构与作者关联起来。如果任由作者自行处理，他们可能会使用各种嵌入式宏或脚注，将作者与其元数据关联起来，这使得从 LaTeX 中提取元数据变得非常困难。

这里的问题部分在于， \author 宏与页面上作者信息的显示紧密交织在一起。这是一个忽视了关注点分离、将结构与显示混为一谈的例子。鉴于 \author 宏的历史遗留问题，我们有意舍弃 \author ，改用 \addauthor 。这意味着作者需要花些功夫，将其他标准 LaTeX 类转换为我们的类，但我们认为这是必要的，因为 LaTeX 助长了不良习惯。

文章中作者信息的呈现方式可能非常复杂，尤其是当每位作者有多个所属机构、资助机构和脚注，或者具名作者数量庞大时。[9] 如何呈现作者信息，应与如何采集作者信息分开决定。作者的职责是准确指定有关作者的元数据，而样式的职责是根据期刊的风格决定如何在页面上排版。请注意，期刊也可能对作者元数据的编码方式提出特殊要求（例如，是否接受笔名或幽灵署名，是否按角色对作者分组，需要哪些类型的联系信息等）。

我们并非最早认识到这个问题的人；有些 LaTeX 样式改进了对 \author 的基本用法，并将元数据采集纳入其创作流程。

authblk 此宏包提供了多种显示作者信息的方式，可以将其显示为脚注，也可以显示为文本块。

[9]例如， https://arxiv.org/abs/2210.03375

ltugboat TUGBoat 使用的样式会重新定义 `\author` 宏，以捕获单个作者的姓名。依次调用其他宏（例如 `\ORCID`）即可将元数据关联到各位作者。

acmart 的 `acmart` [Vey] 样式还重新定义了 `\author` 宏，用于捕获各位作者的信息；通过对宏进行排序，还可以捕获各位作者的所属机构和 ORCID 标识符。

amsart 美国数学会的期刊系列 [AMS19] 使用邮寄地址而非所属机构，但允许每位作者拥有多个地址，并可指定不同于开展研究所在地的“现地址”。它们不提供 ORCID 标识符或 ROR 标识符字段。它们还允许设置第二类作者，称为“贡献者”；此类作者仅因论文的一部分内容而获得署名。

elsarticle Elsevier 的期刊类也会以结构化方式捕获部分元数据。每位作者使用一个 `\author` 命令，并且允许为每位作者关联多个脚注标记。它的构造几乎理想，因为每位作者可以拥有多个所属机构，而且所属机构无需重复列出，因为每个机构都有自己的编号。此外，所属机构还具有不同的属性，因此将来可以扩展以指定 ROR 标识符等信息。

这些方案都在捕获文章元数据方面有所进展，但没有一种能达到 JATS 这类格式所具备的表现力。此外，据我们所知，还没有任何方案尝试通过从 LaTeX 中提取元数据，为出版工作流程提供功能。

10 `iacrcc` LaTeX 和 BibTeX 样式

基于我们从以往工作中汲取的经验，我们设计了一种名为 `iacrcc`[10] 的新文档类，可帮助我们从文档中尽可能多地提取元数据。它既可与 BibTeX 配合使用，并搭配我们自有的 `iacrcc.bst` 样式，也可与 `biblatex` 宏包配合使用。这些文件专为出版工作流程而设计，可生成多种不同格式的元数据。它们不仅会生成可写回 PDF 的元数据，还会生成纯文本格式的元数据，便于为 DOI 注册等其他用途进行处理。我们提取了广泛的元数据，包括替代标题、作者姓名、姓氏、ORCID 标识符、带有 ROR 标识符的所属机构及地址，以及摘要。图 `iacrcc` 展示了 2的作者元数据示例。其中一部分元数据可用 XMP 的基础模式表示，但其他部分则需要 JATS 的扩展模式。

在实现 `\addauthor`、`\title` 和 `\affiliation` 宏时，还包含了一些其他步骤，以确保它们能够捕获文本元数据。除了重音符号等少数例外，`\title` 正文中禁止使用大多数宏。标题中允许在行内数学中使用宏，但如果作者希望在标题中使用其他宏，则还必须提供仅包含文本的 `plaintext` 参数。这类似于标题过长时必须提供眉题的要求。作者姓名中的脚注必须通过 `footnote` 选项提供给 `\addauthor`，并且禁止使用 `\thanks`。

10.1 工作原理

作者的工作流程通常包括多轮运行 `latex`、`bibtex` 或 `biber`，然后再运行两次 `latex`。由此生成的输出是一个 PDF 文件

[10] 可从 https://publish.iacr.org/iacrcc下载

```
\title[running={Emojex documentation},
onclick={https://example.com/emo}、
subtitle={Emojis in LaTeX}、
plaintext={Emojex: use of emojis in LaTeX}、
]{Emojex: use of emojis in \LaTeX}
\addauthor[orcid={0000-0002-0599-0192},inst={1,2}、
onclick={https://www.madmagazine.com/}
email={fester@example.com}、surname={Bestertester}
]{Fester Bestertester}
\addauthor[orcid={0000-0001-7890-5430},inst={2}、
footnote={Thanks mom!}、surname={McCurley}
]{Kevin S. McCurley}\affiliation[ror=044t1p926,
city={New York}、country={United States}]{MAD}
\affiliation[country={United States}]{Self}
\addfunding[crossref=100011047,grantid={A-1234}、
country={Canada}]{AGE-WELL}
```

图 2： iacrcc.cls 中的元数据条目示例

包含 XMP 元数据，也包含一个 \jobname.meta file 文件，其中以结构化格式保存了所有元数据。 .meta file 文件使用宏编写，采用 \write macros。

\jobname.meta 的结构与 YAML 类似。我们考虑过尝试使用 YAML、JSON 或 XML 格式，但每种输出格式都有各自的一套特殊字符和编码要求，而在 LaTeX 中实现这些要求很复杂。编写 Python 代码来解析我们的输出格式，比编写 LaTeX 代码来生成一种更常见的格式容易得多。这段 Python 代码包含在 iacrcc 的代码仓库中。[11]

论文的基本元数据通过使用 iacrcc.cls 文件中的宏写入 .meta 文件。引文信息以两种不同方式之一写入 .meta 文件，具体取决于作者选择使用 BibTEX 还是 biblatex。这两种方法都会生成一个 .bbl 文件，可在期刊工作流程中作为原始 BibTEX 文件的替代文件提供（原始文件可能很大，并且包含未使用的参考文献）。使用 BibTEX 或 biblatex时，处理 .bbl 文件会通过使用 \write宏，在 .meta 文件中生成引文记录。对于 biblatex， \write 宏在 iacrcc.cls 文件中实现；对于 BibTEX，则在 iacrcc.bst 文件中实现。在这两种情况下，最终的 .bbl 文件都包含结构化的引文形式。这使我们能够遵循出版商的标准做法，只要求作者提交其 .bbl 文件，而非整个 BibTEX 文件。

[11]请参阅 GitHub 代码仓库： https://github.com/IACR/latex。

10.2 投稿流水线

作者提交论文时，只需提交其 LaTeX 源文件（可为多个），包括所用的 BibTeX 文件。投稿表单很简洁，因为所有元数据都包含在 LaTeX 和 BibTeX 文件本身中。我们只需采集经过身份验证的`paperid`，并要求投稿作者提供通讯作者的电子邮件地址。我们从`paperid`中提取 DOI，并在编译时将其连同录用日期和收稿日期一起注入 PDF。

作者上传 LaTeX 源文件后，我们的服务器会在包含 TeX Live 实例的 Docker 容器中运行 `latexmk`。我们选择使用 `latex`，因为我们发现无法让 `pdflatex` 从 `\write`生成 UTF-8输出文件。服务器会验证源文件是否已成功编译，并在出现错误时向作者提供报告。

文档成功编译后，我们会运行 Python脚本处理`.meta` 文件，生成 XMP、JATS、JSON 和 Crossref格式的元数据。JSON 格式便于立即在网络上发布文章。Crossref格式可用于为论文注册 DOI。

如果作者对编译生成的元数据和 PDF 感到满意，论文就会进入文字编辑的下一步。在我们的流水线中，这一步的范围相当有限。我们知道，编辑可能会执行语法检查、标点检查或拼写校正等其他任务，但我们认为，处理元数据不应占用人工的时间。尽管我们已尝试解决元数据处理的问题，但我们认为，文字编辑仍是降低开放获取出版成本的最大障碍。

论文经文字编辑最终批准后，无需人工处理任何元数据即可发布。论文发布时，将注册 DOI。

我们使用 `latex` 描述了该工作流程，但也可以使用 TEX4ht 或 LaTEXML，或其他转换器生成 HTML输出。待我们积累足够的 PDF流程经验后，计划对此进行研究。

11 总结

我们认为，LaTeX 可用于简化出版流程中的元数据处理，并开发了一个文档类，希望能大幅提升元数据的质量。通过采用这种方法，我们相信，可以简化低预算运营的开放获取期刊的出版工作流程。

致谢

作者谨向 Gaëtan Leurent 以及为 `iacrtrans`文档类做出贡献的其他贡献者表示感谢，该文档类是本项目的起点。作者还要感谢 Enrico Gregorio 和 David Carlisle 解答有关 LaTeX 内部工作原理的问题。我们也谨向 LaTeX 团队表示诚挚感谢，感谢他们对使用 `\protected@write` 提出的宝贵建议。

参考文献

[AMS19] 美国数学会。*AMS*作者手册期刊类，2019。网址：
https://www.ams.org/arc/journals/packages/jams/jams_amslatex/Author_Handbook_Journals.pdf

[BFK+21] Jeroen Bosman、Jan Erik Frantsvåg、Bianca Kramer、Pierre-Carl Langlais 和 Vanessa Proudman。《开放获取钻石期刊研究》。第一部分：研究发现，2021年3月。本报告由 Science Europe 和 cOAlition S 资助。doi：10.5281/zenodo.4558704 网址：https://doi.org/10.5281/zenodo.4558704[Bre22] Rachel Brennesholtz。《第三方内容合作伙伴内容交付说明》，2022年3月。网址：https://elsevier.widen.net/s/hhldnnsq7r/cp-content-delivery-instructions-extrnl[Cla21] 科睿唯安 Web of Science。《Web of Science 核心馆藏 Current Contents Connect®》，2021年2月。网址：https://clarivate.libguides.com/ld.php?content_id=27109663[Croa] Crossref 存储架构。访问于2022-06-30。网址：https://data.crossref.org/reports/help/schema_doc/5.3.1/index.html[Crob]Crossref 资助机构注册库。https://crossref.org/services/funder-registry/。网址：https://www.crossref.org/services/funder-registry/[DOA22] 开放获取期刊目录。DOAJ XML，2022年。网址：https://doaj.org/docs/xml/[FM22] Ulrike Fischer 和 Frank Mittelbach。在 LaTeX 中添加 XMP 元数据。TUGboat，135(3):263–267，2022年。doi：https://doi.org/10.47397/tb/43-3/tb135fischer-xmp 网址：https://tug.org/TUGboat/tb43-3/tb135fischer-xmp.pdf[GB21] Alexander Grossmann 和 Björn Brembs。《当前学术出版服务市场价格》。F1000Research，2021年。doi：10.12688/f1000research.27468.2 网址：https://f1000research.com/articles/10-20/v2[HFP+12] Laurel L Haak、Martin Fenner、Laura Paglione、Ed Pentz 和 Howard Ratner。ORCID：唯一标识研究人员的系统。《Learned Publishing》，25(4):259–264，2012年。[HRL21] Hanna Hottenrott、Michael E Rose 和 Cornelia Lawson。《学术界多重机构隶属关系的兴起》。《信息科学与技术协会期刊》，72(8):1039–1058，2021年。doi：10.1002/asi.24472 网址：https://arxiv.org/pdf/1912.05576.pdf[Ken] Dianne Kennedy。PRISM 基础元数据规范。网址：https://www.w3.org/Submission/prism/prism-basic.html[Kli17] Marco Klindt。《PDF/A 不利于数字保存》。iPRES，2017年。网址：http://www-archive.cseas.kyoto-u.ac.jp/ipres2017.jp/wp-content/uploads/15.pdf[l3p] l3pdfmeta 模块。网址：https://ctan.math.washington.edu/tex-archive/macros/latex/contrib/pdfmanagement-testphase/l3pdfmeta.pdf[Lam86] Leslie Lamport。LaTeX：文档准备系统。Addison-Wesley 出版公司，第一版，1986年。ISBN 020115790X。[Lam20] Rachael Lammey。《解决识别问题：研究组织注册库一瞥》。《Science Editing》，7(1):65–69，2020年。[McK10]Patrick McKenzie。《程序员对姓名的错误认知》，2010年。网址：https://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/

[MF] Frank Mittelbach 和 Ulrike Fischer。网址：https://ctan.math.washington.edu/tex-archive/macros/latex-dev/required/latex-lab/documentmetadata-support-code.pdf[MFL16] Frank Mittelbach、Robin Fairbairns 和 Werner Lemberg。《LaTeX 字体编码》，2016年。网址：https://www.latex-project.org/help/documentation/encguide.pdf[MGB+04] Frank Mittelbach、Michel Goossens、Johannes Braams、David Carlisle 和 Chris Rowley。《LaTeX Companion》。Addison-Wesley，第二版，2004年4月。ISBN-10：0201362996 ISBN-13：978-0201362992。[Nat] 公开获取美国国家科学基金会资助研究成果。网址：https://www.nsf.gov/news/special_reports/public_access/about_repository.jsp[Nat21a] 美国国家生物技术信息中心（NCBI）。《期刊出版标签库：NISO JATS 1.3版》。https://jats.nlm.nih.gov/publishing/tag-library/1.3/，2021年6月。[Nat21b] 美国国立卫生研究院。联邦资助的沟通与致谢，2021年。网址：https://grants.nih.gov/policy/federal-funding.htm[NIS] NISO。CRediT：贡献者角色分类法。网址：https://credit.niso.org[OJS] 开放期刊系统。网址：https://pkp.sfu.ca/ojs/[Pak] Scott Pakin。hyperxmp——在 LaTeX 文档中嵌入 XMP 元数据。网址：https://ctan.org/pkg/hyperxmp?lang=en[Pas10] Norman Paskin。数字对象标识符（DOI®）系统。《图书馆与信息科学百科全书》，第3卷，1586–1592页，2010年。[pdfa] 技术说明 0008：PDF/A-1 中预定义的 XMP 属性。网址：https://www.pdfa.org/wp-content/uploads/2011/08/tn0008_predefined_xmp_properties_in_pdfa-1_2008-03-20.pdf[pdfb] 技术说明 0008——PDF/A-1 中预定义的 XMP 属性。网址：https://www.pdfa.org/wp-content/uploads/2011/08/tn0008_predefined_xmp_properties_in_pdfa-1_2008-03-20.pdf[pdfc] 技术说明 0009：PDF/A-1 中的 XMP 扩展模式。网址：https://www.pdfa.org/resource/technical-note-tn-0009-xmp-extension-schemas-in-pdfa-1/[RTMS] C. V. Radhakrishnan、Hàn Th´ê Thành、Ross Moore 和 Peter Selinger。使用 pdfTEX 和 pdfx.sty 生成符合 PDF/X 和 PDF/A 标准的 PDF。网址：https://ctan.org/tex-archive/macros/latex/contrib/pdfx?lang=en[TXM+20] Dung Thai、Zhiyang Xu、Nicholas Monath、Boris Veytsman 和 Andrew McCallum。使用 BibTeX 自动生成用于引文字段抽取的标注数据。arXiv，2020年。DOI：10.48550/ARXIV.2006.05563 网址：https://arxiv.org/abs/2006.05563[Usd16] B. Tommie Usdin。想要采用 JATS？需要做出哪些决定？载于《期刊文章标签套件会议（JATS-Con）会议论文集 2016》，2016年。网址：https://www.ncbi.nlm.nih.gov/books/NBK350380/[Vey] Boris Veytsman。acmart——用于排版 ACM 出版物的类。网址：https://ctan.org/pkg/acmart?lang=en

[XMP] *ISO 16684-1:2019*：图形技术——可扩展元数据平台（*XMP*）——第*1*部分：数据模型、序列化和核心属性。网址： https://www.iso.org/standard/57421.html