计算拓扑在文本多标签分类中的应用：拓扑结构特征与模型融合的研究复现

姓名：蔡哲熙　学号：12311415

2026 年 1 月 8 日

摘要

本文研究计算拓扑（Topological Data Analysis, TDA）在真实工程任务——文本多标签分类——中的应用效果。我们以电影剧情简介数据集为例，首先构建传统基线模型（TF–IDF + One-vs-Rest 逻辑回归），并实现基于分块表示的拓扑结构特征方法（TP2），随后训练深度序列模型 BiLSTM 作为额外基线。在此基础上，我们使用 stacking 融合策略，将多个模型输出的类别概率拼接并在验证集上训练二层融合器，从而评估拓扑结构信号对总体性能的边际贡献。实验表明：TP2 特征单独用于分类时表现有限，但在融合框架下能提供互补信息，提升宏平均 F1 并改善部分类别的识别效果。最后，我们通过消融实验量化了各模块的贡献，并讨论了方法局限。

1　引言

传统的文本分类方法主要通过 Term Frequency（或者 TF-IDF）和 Word embedding 两种表示方法将文本转换成数据结构。参考论文尝试寻找除了这两种表示方法外，不同特征的文本是否有其他潜在的拓扑结构，以及这些拓扑结构能否帮助文本分类。

TF–IDF 表示在主题区分任务中通常表现稳健，其优势在于能够以较低的计算成本捕捉“类别关键词”所带来的强信号，并在中等规模数据上配合线性分类器取得可靠结果。然而，TF–IDF 的核心假设接近词袋模型：它主要编码词项出现的统计信息，对词序、句法关系以及跨句的语义演化缺乏显式建模能力。因此，当判别线索更多来自叙事结构、主题切换或语义推进过程时，纯词频统计往往难以充分表达这些结构性信息。相比之下，深度序列模型（如 BiLSTM）能够利用 token 序列信息建模局部上下文与部分长程依赖，但其参数规模更大、训练与调参成本更高，同时对数据规模、类别不均衡与超参数设置更为敏感，导致在实际工程数据上未必始终优于一般的 TF-IDF 线性分类。

参考论文作者提出了一种基于持续同调的拓扑结构特征，在此基础之上将文本诱导出的拓扑结构压缩为固定维度的向量并直接运用于分类。具体而言，对于一篇文档，按照顺序将其等分为 10 个 blocks（每个 block 大概含 $\frac{1}{10}$ total tokens），然后对这 10 个 blocks 分别计算它们的 TF-IDF 向量，并以余弦距离定义 blocks 之间的相似性度量，最终得到一个包含 10 个 vertices 的图。接着基于 Vietoris-Rip 过滤，计算其 0 维与 1 维持续同调，分别刻画 blocks 之间的连通性演化与环结构。最后，将条形码/PH 图汇总为固定维度的拓扑向量（例如由 H_0 的若干 death 值与 H_1 的统计量组成），作为结构特征输入到下游分类模型中。

2 数据集

2.1 数据来源

数据来源与复现的论文保持一致，来自Wikipedia Movie Plots from Kaggle，我们只选取其中包含 action/comedy/drama/romance 四类标签的文档。

2.2 数据清洗与划分

我们去除 tokens 数小于 200 的文档，对剩余的文档做数据清洗：把所有字母小写化，删去多余的空格。并把文档的标签做成 4 维的 0/1 向量。最后将整个数据集划分成训练集、验证集和测试集。

Split	Total	action	comedy	drama	romance
Train	7989	1164	3455	5140	1635
Val	1714	248	670	829	181
Test	1715	806	682	799	184

表 1: 数据划分与标签分布

3 模型方法

3.1 基线模型：TF–IDF + One-vs-Rest 逻辑回归

我们首先构建一个强基线（strong baseline），采用经典的 “TF–IDF + 线性分类器” 框架处理多标签文本分类任务。设训练语料的词表大小为 $|V|$，对每篇文档 d，TF–IDF 将其表示为稀疏向量

$$\mathbf{x}_d \in \mathbb{R}^{|V|}, \qquad x_{d,i} = \mathrm{tfidf}(t_i, d),$$

其中 t_i 为词表中的第 i 个词项（或 n-gram），tfidf 权重综合了词频（TF）与逆文档频率（IDF），从而强调对类别更具区分性的词项。

特征提取（TF–IDF） 我们使用 `scikit-learn` 的 `TfidfVectorizer` 提取 TF–IDF 特征，并固定如下参数：

- `ngram_range` $= (1, 2)$，同时使用 unigram 与 bigram；
- `min_df`$= 2$，过滤极低频词项；
- `max_df`$= 0.9$，过滤过于常见的词项；
- `max_features`$= 50000$，限制特征维度以控制计算与内存开销。

分类器（One-vs-Rest Logistic Regression） 由于任务为四类多标签分类，我们采用 One-vs-Rest（OvR）策略：对每个标签 k 训练一个二分类逻辑回归器，输出该标签为正例的概率 $\hat{p}_k(d)$，从而得到文档级的四维概率向量 $\hat{\mathbf{p}}(d) \in [0, 1]^4$。逻辑回归使用 `liblinear` 求解器（支持概率输出），最大迭代步数为 2000，并设置 `class_weight=balanced` 以缓解标签不均衡问题。

阈值与评价指标 对于每个标签 k，当 $\hat{p}_k(d) \geq \tau$ 时预测为正例；本文默认阈值 $\tau = 0.5$。评价指标采用 micro/macro 平均的 Precision、Recall 与 F1，并同时报告每个标签的 F1 分数，以便分析弱标签（如样本较少的类别）的表现差异。

超参数选择（验证集选 C） 逻辑回归的正则化强度由参数 C 控制。我们在候选集合

$$C \in \{0.25, 0.5, 1.0, 2.0, 4.0\}$$

上进行网格搜索：对每个 C，在训练集上拟合模型，并在验证集上计算 macro-F1；选择使验证集 macro-F1 最大的 $C^\star$ 作为最终基线超参数。

训练策略与可复现输出 为便于后续 stacking 融合实验，我们保存两类概率输出文件：

1. **用于二层融合训练的验证集概率：** 使用 “train-only” 训练得到的最优模型（参数为 $C^\star$）对验证集输出四维概率，并保存为 CSV（包含 `doc_id` 与四个概率列）。
2. **最终测试集概率：** 用 “train+val” 在 $C^\star$ 下重训模型后，对测试集输出四维概率并保存为 CSV，同时保存最终测试集指标结果。

此外，我们将 TF–IDF 参数、逻辑回归参数、数据过滤规则、数据划分统计与网格搜索结果一并记录为 CSV/JSON 文件，并保存训练好的模型（含 TF–IDF 向量器与 OvR 分类器）以支持完全复现。

3.2 TP2：分块表示的拓扑结构特征

为在词项统计之外刻画文档内部的组织结构，我们采用参考文献中提出的 TP2 管线：将长文本按顺序分块（block），在分块层面构造相似性结构，并将其压缩为固定维度的拓扑/结构特征向量。该方法的核心直觉是：即便两篇文档在全局词频上相似，它们在叙事推进、主题切换与信息分布的局部结构上仍可能不同；分块表示能够部分捕捉这种 “局部–全局” 的组织差异。

输入与分块（Input & Blocking） 给定一篇文档的 token 序列长度为 T，我们将其按原始顺序划分为 B 个连续分块（本文取 $B = 10$），每个分块包含约 T/B 个 token。对第 b 个分块，使用与基线一致的 TF–IDF 词表与权重规则，将其表示为稀疏向量

$$\mathbf{v}_b \in \mathbb{R}^{|V|}, \qquad b = 1, \ldots, B,$$

从而每篇文档对应一组分块向量 $\{\mathbf{v}_1, \ldots, \mathbf{v}_B\}$。需要强调的是：分块 TF–IDF 并非对整篇文档 TF–IDF 向量进行切片，而是先在文本层面按顺序切分，再对每个分块独立计算 TF–IDF 表示。

块间关系：由相似性到度量结构（Block Relations） 为了刻画分块之间的相似性，我们对任意两块 (i, j) 计算余弦相似度，并将其转化为距离：

$$\mathrm{dist}(i, j) = 1 - \cos(\mathbf{v}_i, \mathbf{v}_j).$$

由此得到一个 $B \times B$ 的距离矩阵，可视为一个 B 点的度量空间。直观上，当两块包含相近的主题词或语义片段时，它们的距离更小；距离矩阵因此编码了文档内部不同片段之间的 “接近/分离” 结构。

分块 TF–IDF 表示与避免信息泄漏 对每个分块文本计算 TF–IDF 向量表示。需要强调的是，为避免验证集/测试集信息泄漏，我们仅使用训练集的所有分块文本来 `fit` TF–IDF 向量器；随后对 train/val/test 的分块分别进行 `transform` 得到分块向量矩阵 $X_b \in \mathbb{R}^{10\times|V|}$。本实验中 TF–IDF 参数与基线保持一致（如 $(1,2)$-gram、`min_df`、`max_df`、`max_features` 等），以确保对比公平。

固定维特征输出（Output: 14-D Vectorization） 在该度量结构上，我们使用持续同调（persistent homology）对随尺度变化的连通性与环结构进行摘要，并将其向量化为固定维度特征。具体而言，我们提取 0 维与 1 维的持续信息：

- H_0（连通分支）：由于 B 个点最终合并为一个连通分支，H_0 提供 $B-1$ 个“死亡时间”（death times）。当 $B=10$ 时得到 9 个标量。
- H_1（环结构）：环的数量与持续时间不固定，为获得固定维表示，我们使用若干统计量对其总结，例如环数量、birth 与 duration 的均值与标准差等。

综合上述两部分，TP2 为每篇文档输出一个固定长度的向量（本文为 14 维），可与非拓扑特征拼接，或单独输入到下游分类器（如 XGBoost）中。该设计使得拓扑/结构信息能够以“传统机器学习友好”的方式接入分类任务，并支持与基线模型进行公平对比与消融分析。

3.3 深度序列模型：BiLSTM

作为序列建模基线，我们实现了一个面向多标签分类的 BiLSTM 模型。该模型以 token 序列为输入，通过可学习的词嵌入层将离散 token 映射为稠密向量序列，再由双向 LSTM 汇聚上下文信息，最终输出四个标签的 logits，并经 sigmoid 转换为概率。

Tokenize 与词表（Vocabulary） 在完成基础清洗后，我们采用空格分词（`split`）得到 token 序列，并仅使用训练集构建词表以避免信息泄漏。词表包含两个特殊符号：`<PAD>` 与 `<UNK>`；出现频次低于 `min_freq` 的 token 被过滤。输入序列统一截断/填充到最大长度 `max_len`，并保存词表文件以支持复现。

模型结构 模型由以下模块组成：

- **Embedding：**将 token id 映射到 $\mathbb{R}^d$（本实验 $d=128$）；
- **BiLSTM：**隐藏维度为 `hidden`，双向输出；
- **Pooling：**取最后一层的正向与反向隐藏状态拼接作为文档表示；
- **Linear：**线性层映射到 4 维 logits，对应四个标签。

为减少 padding 位置对计算的影响，训练中使用 `pack_padded_sequence` 按真实长度打包序列。

损失函数、优化与训练策略 多标签任务采用 `BCEWithLogitsLoss`。考虑标签不均衡，我们根据训练集正负样本比例估计 `pos_weight`，以提升稀有标签的学习信号。优化器使用 Adam，学习率为 `lr`，训练 `5` 个 epoch。每个 epoch 后在验证集上评估 macro-F1，并保存验证 macro-F1 最优的模型参数作为最终模型。

3.4 模型融合：Stacking

为融合不同模型的互补性信息，我们采用概率级 stacking：将多个基学习器（base learners）的四维概率输出拼接为 meta 特征，再训练一个二层融合器（meta-learner）进行最终预测。该策略的关键规范是：**二层模型仅使用验证集进行训练与阈值调优，测试集仅用于一次性最终评估**，从而避免评估偏差。

Meta 特征构造 对每个样本，我们收集若干基模型的四维概率输出并按 `doc_id` 对齐；将这些概率按列拼接形成 meta 特征向量。若包含 M 个基模型，则 meta 特征维度为 $4M$（例如融合 BiLSTM + XGB(TP2) + baseline LR 时，维度为 12）。

Meta-learner：OvR Logistic Regression 二层融合器采用 One-vs-Rest 逻辑回归：对每个标签训练一个二分类器，输出最终四维概率。逻辑回归使用 `liblinear` 求解器、`class_weight=balanced`，并限制最大迭代步数以保证收敛。

阈值选择与评估规范 由于多标签预测需要将概率转为 0/1，我们在验证集上进行阈值选择，并在测试集上固定该阈值评估。本文同时报告两种阈值策略：

- **全局阈值：**在预设网格上搜索单一阈值，使验证集 macro-F1 最大；
- **逐标签阈值：**分别为每个标签选择使该标签 F1 最大的阈值向量。

最终在测试集上分别给出两种阈值策略下的 micro/macro F1 与每标签 F1，并保存融合模型与阈值配置文件以支持复现。

消融实验（Ablation） 为量化各基模型的边际贡献，我们进一步对不同融合组合进行消融：baseline-only、baseline+BiLSTM、baseline+XGB(TP2)、baseline+BiLSTM+XGB(TP2)。每个组合均使用相同的二层学习与阈值选择流程，确保对比公平。

4 实验结果

4.1 主要结果

表2和3汇总了本项目的主要对比结果，包含：强基线 TF–IDF+LR、仅使用 TP2 特征的 XGBoost（TP2-only）、深度基线 BiLSTM，以及最终的 stacking 融合模型。为保证评估公平，我们遵循多标签任务的常见做法：在验证集上选择阈值（包含全局阈值或逐标签阈值两种策略），并在测试集上仅进行一次最终评估，报告 micro/macro-F_1 及各标签的 F_1。

从整体表现看，TF–IDF+LR 作为强基线在该任务上非常具有竞争力；单独使用 TP2 的 14 维结构特征（TP2-only）难以达到基线水平，但其输出与其他模型具有互补性。BiLSTM 能利用序列信息，但在本项目设置下单模表现仍低于强基线。最终 stacking 融合模型在 macro-F_1 上取得最优结果（相比基线有稳定提升），说明来自不同建模范式的概率输出能够互补，从而提升整体性能。

模型	micro-F_1	macro-F_1
TF–IDF+LR (baseline)	0.698	0.646
XGB (TP2-only)	0.494	0.429
BiLSTM	0.557	0.482
Stacking (LR + XGB(TP2) + BiLSTM)	0.707	0.657

表 2: 主要模型在测试集上的整体表现

模型	F_1(action)	F_1(comedy)	F_1(drama)	F_1(romance)
TF–IDF+LR (baseline)	0.652	0.726	0.740	0.466
XGB (TP2-only)	0.303	0.570	0.635	0.206
BiLSTM	0.396	0.580	0.627	0.326
Stacking (LR + XGB(TP2) + BiLSTM)	0.668	0.740	0.739	0.482

表 3: 主要模型在测试集上的逐标签 F_1

4.2 消融实验（Ablation）

为分析不同信号来源的边际贡献，我们进一步进行消融实验：以 TF–IDF+LR 为基线，分别加入 BiLSTM 概率、加入 XGB(TP2) 概率，以及同时加入二者，并保持相同的二层融合器与阈值选择流程。表 4和5给出了测试集结果。

可以观察到：在 macro-F_1 上，加入 BiLSTM 带来约 +0.003 的小幅提升；加入 XGB(TP2) 带来约 +0.006 的提升；两者同时加入时提升达到约 +0.011。从逐标签表现看，TP2 更明显地改善了 comedy/romance 等标签，而 BiLSTM 对 action 有一定帮助但对 drama 并非单调提升；二者共同加入时 romance 的提升最为明显，体现了“结构特征 + 序列特征”的互补性。

组合	micro-F_1	macro-F_1	Δmacro
Baseline (LR)	0.698	0.646	0.000
+ BiLSTM	0.696	0.649	0.003
+ XGB(TP2)	0.704	0.652	0.006
+ BiLSTM + XGB(TP2)	0.707	0.657	0.011

表 4: 消融实验：在基线之上不同信号的边际贡献

组合	F_1(action)	F_1(comedy)	F_1(drama)	F_1(romance)
Baseline (LR)	0.652	0.726	0.740	0.466
+ BiLSTM	0.666	0.726	0.733	0.470
+ XGB(TP2)	0.657	0.739	0.740	0.474
+ BiLSTM + XGB(TP2)	0.668	0.740	0.739	0.482

表 5: 消融实验：逐标签 F_1

5 讨论

5.1 为什么 TF–IDF 基线很强

从实验结果看，TF–IDF+LR 在本任务上表现非常具有竞争力。其原因在于电影类型（genre）预测高度依赖关键词与局部词组模式：例如动作片常出现 *battle, gun, chase* 等词汇，爱情片常出现 *love, marriage, relationship* 等词汇。TF–IDF 本质上对“类别指示词”赋予较高权重，并通过线性分类器在高维稀疏空间中进行有效分离；同时其训练代价低、超参数较少、对中等规模数据集更稳定。因此，TF–IDF+ 线性模型常被视为文本主题/风格任务中的强基线，这也解释了深度模型在本项目设置下未必能够超过该基线。

5.2 为什么 TP2-only 较弱但在融合中有效

TP2-only（XGB on TP2 features）单独使用 14 维结构/拓扑特征时性能显著低于基线，这在一定程度上是可预期的：与 TF–IDF 数万维的词项特征相比，TP2 的表示容量更小，且其信号来源并非直接建模语义，而是通过分块关系捕捉文档内部的组织结构（例如主题切换、片段相似度模式等）。因此 TP2-only 更像是“弱但不同”的信息源。

然而，消融与 stacking 的结果表明：当 TP2 的概率输出与 TF–IDF 基线、BiLSTM 等模型的输出融合时，macro-F_1 能获得稳定增益（例如在基线之上约 +0.011）。这说明 TP2 与传统词频/序列建模在错误模式上存在互补：基线容易在弱标签（如 romance）或语义边界模糊样本

上产生系统性偏差，而 TP2 在分块层面引入的结构信号可能帮助区分某些 “叙事组织” 相近或相异的样本，从而在融合框架中提升整体性能。

5.3 BiLSTM 与融合收益的解释

BiLSTM 能利用 token 序列信息建模局部上下文，但其在本项目中单模型表现仍弱于 TF–IDF 基线。可能原因包括：(i) 词袋特征对 genre 任务已非常有效；(ii) 深度模型训练对超参数与数据规模更敏感；(iii) 类别不均衡导致模型更偏向频繁标签。尽管如此，BiLSTM 的输出在 stacking 中仍带来一定边际贡献，表明序列建模捕捉到的部分信息与 TF–IDF 并不完全重合。

5.4 局限性

本文以复现实验为主要目标，仍存在若干局限：(i) 仅实现并验证了 TP2 管线，未进一步实现 TP1 或更丰富的拓扑向量化方式；(ii) 预测阈值对 macro-F_1 影响较大，尽管我们在验证集上选择阈值以避免测试泄漏，但不同阈值策略仍会影响结论的稳定性；(iii) 数据来自 Kaggle 的 Wikipedia Movie Plots 子集，并经过 tokens$\geq$ 200 的过滤，结论对其他数据域的可迁移性仍需进一步验证。

6 结论

本文围绕 “计算拓扑能否帮助传统模型更聪明” 这一问题，在电影剧情简介的多标签分类任务上复现并实现了基线模型（TF–IDF+LR）、结构/拓扑特征管线（TP2）、深度序列基线（BiLSTM）以及 stacking 融合框架。实验结果表明：TP2-only 作为单独特征源的性能有限，但其与 TF–IDF、BiLSTM 的输出具有互补性；通过概率级 stacking 融合可以获得稳定的宏平均指标提升，且消融实验验证了 TP2 与序列模型均能提供边际贡献。

参考文献

[1] Shafie Gholizadeh, Ketki Savle, Armin Seyeditabari, and Wlodek Zadrozny. *Topological Data Analysis in Text Classification: Extracting Features with Additive Information.* arXiv:2003.13138, 2020.

A 关键超参数汇总

表 6: 关键超参数与实验设置（按模块汇总）

模块	参数	取值/说明
数据预处理与划分		
预处理	随机种子	`SEED=0`
预处理	目标标签	`{action, comedy, drama, romance}`
预处理	过滤规则	`MIN_TOKENS=200` 且命中目标标签 ≥ 1（多标签数据）
划分	比例	`train/val/test = 70%/15%/15%`（先 30% 留作 temp，再平分为 val/test）
Baseline：TF–IDF + One-vs-Rest Logistic Regression		
TF–IDF	`ngram_range`	`(1, 2)`
TF–IDF	`min_df`	`2`
TF–IDF	`max_df`	`0.9`
TF–IDF	`max_features`	`50000`
LR(OVR)	`solver`	`liblinear`
LR(OVR)	`class_weight`	`balanced`
LR(OVR)	`max_iter`	`2000`
LR(OVR)	C 搜索网格	`{0.25, 0.5, 1.0, 2.0, 4.0}`(按 VAL macro-F_1 选最优)
评估	阈值	`THRESHOLD=0.5`（sigmoid 概率 ≥ 0.5 判为正）
TP2 特征构造：分块 TF–IDF → 距离矩阵 → PH 向量		
分块	分块数	`N_BLOCKS=10`（将每篇文档 token 序列等分为 10 段）
块表示	块级 TF–IDF	与 baseline 相同的 `TFIDF_PARAMS`（在训练集所有 blocks 上拟合）
距离	距离度量	余弦距离：`cosine_distances`，得到 10×10 距离矩阵 D
PH	计算工具	`ripser(D, distance_matrix=True, maxdim=1)`
向量化	输出维度	14 维：H_0 取 9 个最大 death；H_1 取 5 个统计量（count / birth mean / pers mean / birth std / pers std）
TP2-only 分类器：XGBoost (One-vs-Rest)		
XGB	`n_estimators`	`500`
XGB	`max_depth`	`5`
XGB	`learning_rate`	`0.05`
XGB	`subsample`	`0.8`
XGB	`colsample_bytree`	`0.8`
XGB	`reg_lambda`	`1.0`
XGB	`objective`	`binary:logistic`（逐标签二分类，OVR 包装）
XGB	`eval_metric`	`logloss`
XGB	`n_jobs`	`-1`
评估	阈值	`0.5`（与 baseline 对齐）
深度基线：BiLSTM		
词表	最小词频	`MIN_FREQ=2`（训练集统计词频构建词表）
序列	最大长度	`MAX_LEN=400`（截断/补齐到 400）
训练	batch size	`BATCH_SIZE=64`
模型	embedding 维度	`EMB_DIM=128`
模型	hidden size	`HIDDEN=128`（双向 $\Rightarrow$ 输出维度 2×128）
模型	层数	`NUM_LAYERS=1`
模型	dropout	`DROPOUT=0.2`（层数为 1 时 LSTM 内部 dropout 关闭，仅保留输出 dropout）