

基于Ngboost、 BERT、交叉注意力机制的 社交媒体评论热度预测

以小米SU7微博数据为例

2025年

研究背景

社交媒体热度预测的重要性

- 舆情监控与品牌管理
- 内容推荐系统优化
- 信息传播规律研究

小米SU7发布引发热议

- 2024年3月28日正式发布
- 小米首款新能源汽车
- 社交媒体广泛讨论

数据特点

- 长尾分布：少数热门评论获得大量互动
- 时效性强：热度随时间快速衰减

研究挑战

长尾分布问题

- 传统MSE不适用
- 大数预测微小误差被过度惩罚
- 需要关注相对误差

不确定性量化

- 模型需要知道"自己不知道什么"
- 输出预测置信度
- 概率预测而非点估计

特征表示

- 多模态特征融合
- 文本、用户、时序信息
- 重复/相似内容检测

数据集构建

数据来源

- 目的：原数据集点赞数据缺失，重新爬虫
- 来源：新浪微博平台
- 时间：2025.3.27 - 2025.4.14
- 方式：隧道代理绕过反爬虫
- 机制：检查点机制支持断点续传

271,452

条评论数据

数据集划分 (8:1:1)

数据集	样本数	占比
训练集	217,162	80%
验证集	27,145	10%
测试集	27,145	10%

重复数据（跟风/引用/复读 现象）

G	H	I	J	K
用户昵称	评论内容	发布时间	子评论数	点赞数
身骑白马	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的	2025/4/1 23:06	1856	29471
悻然江木	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险公司，而是找汽车商家的	2025/4/1 23:22	0	3
超级宇宙霸	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的	2025/4/1 23:24	0	1
宝你铭	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的	2025/4/1 23:29	0	1
临江雪_	是这样的//@身骑白马:我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，	2025/4/1 23:57	0	0
伤口wound	@雷军 我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商	2025/4/2 6:03	0	2
璇律66	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的	2025/4/2 6:42	54	429
肖顺东	我的荣放有次跟车过近突然提醒我注意危险，我都没有发现有预碰撞提醒功能，丰田的销售	2025/4/2 9:10	0	0
快_Ke	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的---	2025/4/2 11:45	1	4
感悟瞬间8	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的	2025/4/2 11:57	2	8

第一次出现热度最高
重复数据第一次出现强制划入训练集

重复数据（跟风/引用/复读 现象）

代码：

```
# 9. 评论重复内容分析（跟风复制）
# =====
print("=" * 60)
print("【评论重复内容分析 - 跟风复制】")
print("=" * 60)

# 清洗评论内容
comments_df['内容清洗'] = comments_df['评论内容'].fillna('').str.strip()

# 按评论内容分组
comment_content_groups = comments_df.groupby('内容清洗').agg({
    '评论ID': 'count',
    '发布时间': ['min', 'max'],
    '用户昵称': lambda x: list(x.unique())[:5], # 只取前5个用户
    '子评论数': ['first', 'last', 'mean'],
    '点赞数': ['first', 'last', 'mean']
}).reset_index()

comment_content_groups.columns = ['内容', '出现次数', '首次发布', '最后发布', '用户列表',
                                  '子评论_首次', '子评论_末次', '子评论_均值',
                                  '点赞_首次', '点赞_末次', '点赞_均值']

# 筛选重复出现的内容
dup_comments_content = comment_content_groups[comment_content_groups['出现次数'] > 1].copy()
print(f"总不同评论内容数: {len(comment_content_groups):,}")
print(f"重复出现的内容: {len(dup_comments_content):,} ({len(dup_comments_content)/len(comment_content_groups)*100:.1f}%)")

# 出现次数分布
print("\n--- 评论内容出现次数分布 ---")
print(comment_content_groups['出现次数'].value_counts().sort_index().head(10))

# 查看出现次数最多的评论内容
print("\n--- 出现次数最多的10条评论内容 ---")
top_dup_comments = dup_comments_content.nlargest(10, '出现次数')[['内容', '出现次数', '用户列表']]
for idx, row in top_dup_comments.iterrows():
    content = row['内容'][:50] + '...' if len(row['内容']) > 50 else row['内容']
```

结果：

=====

【评论：首次发布 vs 跟风指标对比】

=====

重复内容的所有评论记录数：33,569

首次发布(原创)：5,664

后续跟风：27,905

首次发布(有效点赞)：4,507

跟风发布(有效点赞)：21,603

--- 指标对比 ---

子评论数：

首次发布 - 均值：2.24，中位数：0

跟风发布 - 均值：0.17，中位数：0

✓ 首次发布更优（均值是跟风的 13.5 倍）

点赞数：

首次发布 - 均值：15.95，中位数：0

跟风发布 - 均值：1.13，中位数：0

✓ 首次发布更优（均值是跟风的 14.2 倍）

MinHash相似度检测

算法原理

- 通过Jaccard相似度估计文本相似度
- 文本转换为N-gram集合 (N=3)
- 使用128个哈希函数计算MinHash签名

优化策略

- 滑动窗口：维护近10,000条评论的签名
- 全局TopK：记录高频重复文本（出现>3次）
- "老梗"记忆机制

核心公式

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{交集元素个数}}{\text{并集元素个数}}$$

复杂度对比

暴力计算
 $O(n^2)$



MinHash
 $O(n \cdot k)$

k = 哈希函数数量 (128)

MinHash相似度检测

包含“我这辈子”的评论：17 条

时间索引	BGE相似度	BGE重复	MH相似度	MH重复	评论摘要
32062	0.9020	1	0.0859	0	这，可能是我这辈子开过加速最快情绪价值最高的车...
58320	0.9110	2	0.1094	0	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
59338	0.9984	4	0.8594	1	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险公司，而是找汽车商家的...
59448	0.9981	3	1.0000	2	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
59736	1.0000	7	1.0000	3	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
61065	0.9662	4	0.7500	4	是这样的//@身骑白马:我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
62879	0.8693	0	0.5312	5	是的，调查结果还没出来，就让人家车企负责[允悲]我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的，目的地太明显了，一堆傻子还在嘿嘿嘿...
66067	0.9189	5	0.7734	6	@雷军 我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的.....人不能太老实...
66115	0.9820	8	0.8984	7	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商厂家的...
66557	1.0000	12	1.0000	9	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
73041	0.7771	0	0.3438	0	我的荣放有次跟车过近突然提醒我注意危险，我都没有发现有预碰撞提醒功能，丰田的销售也没有给我介绍。//@听龙卷风吹:雷军脾气太好了，换大众丰田试试[允悲] //@璇律66:我这
82118	0.8198	0	0.5078	5	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的----这母亲想用舆论影响结果，耐心等待调查结果吧。中国就是会哭的有奶吃，这个博弈要改改
82736	0.9071	4	1.0000	8	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
93435	0.9141	2	0.8125	7	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的[doge]...
101921	0.9972	8	1.0000	8	我这辈子第一次听说，出了交通事故在第一时间不找警察，不找保险，而是找汽车商家的...
194806	0.9078	4	0.0703	0	我这辈子最佩服的就是雷总了，亏钱做生意都能做成首富[悲伤]...
246524	0.7555	0	0.0312	0	友哥们别急哈，正常人可以过几个月半年蹲一下我微博，这边不是网红网黑，放心蹲哈。不正常的等我下班一个投诉哈。我不是雷总粉丝哈，雷总好基友黑过我偶像，我这辈子都讨厌

不确定性

--- 出现次数最多的10条评论内容 ---

[出现2485次]

[出现595次] 转发微博

[出现547次] 感谢分享

[出现500次] 图片评论 查看图片

[出现408次] [doge]

[出现391次] 是的

[出现231次] 安全第一

[出现225次] [允悲]

[出现223次] 下午好呀

[出现202次] [打call]

转发/评论/转发本身具有不确定性和随机性，
预测1000个和预测1005个没有太大差别

语义具有不确定性和模糊性，
比如“查看图片”的评论在缺少图片信息的情况下，
语义具有极大的模糊性。

LDA主题分析

```
11 # 设置主题数
12 num_topics = 6
13
14 # 训练LDA模型 -- 使用更保守的参数
15 lda_model = LdaModel(
16     corpus=corpus,
17     id2word=dictionary,
18     num_topics=num_topics,
19     random_state=42,
20     passes=3, ..... # 减少迭代次数
21     alpha='symmetric', ..... # 使用对称alpha, 减少计算
22     eta='symmetric', ..... # 使用对称eta
23     chunksize=500, ..... # 减小批次大小
24     iterations=30, ..... # 限制每个文档的迭代次数
25     eval_every=None, ..... # 不在训练中评估, 节省内存
26     minimum_probability=0.01
27 )
```

=====

【主题关键词】

=====

主题 1: 速度(0.032), 电池(0.019), 电车(0.015), 油车(0.014), 碰撞(0.014), 不行(0.014), 起火(0.013), 刹车(0.013), 小米(0.013), 问题(0.012)

主题 2: 华为(0.032), 车祸(0.024), 确实(0.023), 出来(0.023), 人家(0.022), 小米(0.022), 别人(0.019), 米粉(0.018), 你们(0.017), 知道(0.015)

主题 3: 那么(0.029), 不要(0.025), 直接(0.024), 一下(0.022), 这么(0.020), 为啥(0.013), 出事(0.012), 理解(0.012), 标准版(0.012), 其实(0.011)

主题 4: 小米(0.118), 家属(0.025), 事故(0.023), 雷军(0.016), 联系(0.014), 现在(0.014), 流量(0.013), 出来(0.013), 他们(0.012), su7(0.012)

主题 5: 车门(0.029), 知道(0.025), 打不开(0.021), 开车(0.021), 手机(0.019), 汽车(0.019), 图片(0.018), 造谣(0.017), 相信(0.016), 打开(0.015)

主题 6: 智驾(0.046), 问题(0.025), 车主(0.024), 驾驶(0.022), 安全(0.021), 高速(0.018), 辅助驾驶(0.018), 责任(0.016), 司机(0.016), 不能(0.016)

特征工程 (4类17维)

基础特征 (7维)

用户评论数、是否认证、是否一级评论
微博评论数、发布时间 (小时/星期/工作日)

文本特征 (6维)

评论长度、感叹号数、问号数
表情数、话题标签、小米关键词数

LDA主题 (1维)

潜在狄利克雷分配主题模型
性能讨论 / 安全话题 / 品牌对比...

时间密度 (3维)

时间顺序索引、最大相似度、重复次数
MinHash算法高效检测相似评论

NGBoost模型

自然梯度提升 (Natural Gradient Boosting)

- 基于梯度提升的概率预测方法
- 直接拟合条件概率分布参数
- 同时输出预测均值 μ 和标准差 σ

对数尺度NLL损失函数

$$\mathcal{L} = 0.5 \cdot \log(\sigma^2) + [\log(y+10) - \log(\mu+10)]^2 / (2\sigma^2)$$

模型配置

参数	值
n_estimators	100
max_depth	10
learning_rate	0.1

BGE神经网络架构

模型架构流程



BGE-base-zh-v1.5

- 768维中文文本嵌入
- 编码4类文本：评论/微博/根评论/父评论
- 默认冻结，支持微调

Cross-Attention融合

- 评论向量作为Query
- 上下文（微博/父评论）作为Key/Value
- 自适应融合上下文信息

命名实体、特殊符号 训练嵌入向量

- @高频用户 保留原始ID
- @其他用户 替换为 _USER_
- 小米汽车相关词汇
- 表情符号
- 特殊字符

双预测头

- 均值头：预测子评论数
- 方差头：预测不确定性
- Softplus激活保证正值

双预测头

(μ, σ)

命名实体词典

≡ vip_users.txt X

≡ vip_users.txt > data

```
1 # VIP用户列表 (出现>=20次)
2 小米法务部 328
3 雷军 292
4 小米汽车 69
5 王化 66
6 鸿蒙智行法务 49
7 薛定谔的英短咕咕咕 33
8 余承东 32
9 小米公司发言人 29
10 小蒜苗长 29
11 万能的大熊 27
12 我是大彬同学 27
13 科技新一 26
14 美国驻华大使馆 26
15 AI逃逸 24
16 小米公司 24
17 羊驼的睡衣 24
18 卢伟冰 22
19 诗雨370491153 20
20 不会武功的武功李云飞 20
21 |
```

≡ xiaomi_word.txt U X

≡ xiaomi_word.txt > data

```
1 '小米汽车',
2 '小米SU7',
3 'SU7',
4 '雷军',
5 '保时捷',
6 'Taycan',
7 '智能驾驶',
8 '自动驾驶',
9 '辅助驾驶',
10 '智驾',
11 '续航',
12 '电池',
13 '充电',
14 '快充',
15 '超充',
16 '智能座舱',
17 '车机',
18 '大屏',
19 '澎湃os',
20 '性价比',
21 '质价比',
22 '定价',
23 '预售',
24 '交付',
25 '锁单',
26 '大定',
27 '小定',
28 '比亚迪',
29 '特斯拉',
30 'Model3',
31 '蔚来',
32 '小鹏',
33 '理想',
34 '问界',
35 '华为',
36 '遥遥领先',
37 '真香',
38 '割韭菜',
39 '智商税',
```

评价指标体系

精度指标

MSLE (均方对数误差)

$$MSLE = \frac{1}{n} \sum (\log(y_{true} + 10) - \log(y_{pred} + 10))^2$$

ACP@20% (容忍区间准确率)

$$|y_{pred} - y_{true}| \leq \max(\alpha \cdot y_{true}, \delta)$$

不确定性指标

Log NLL (负对数似然)

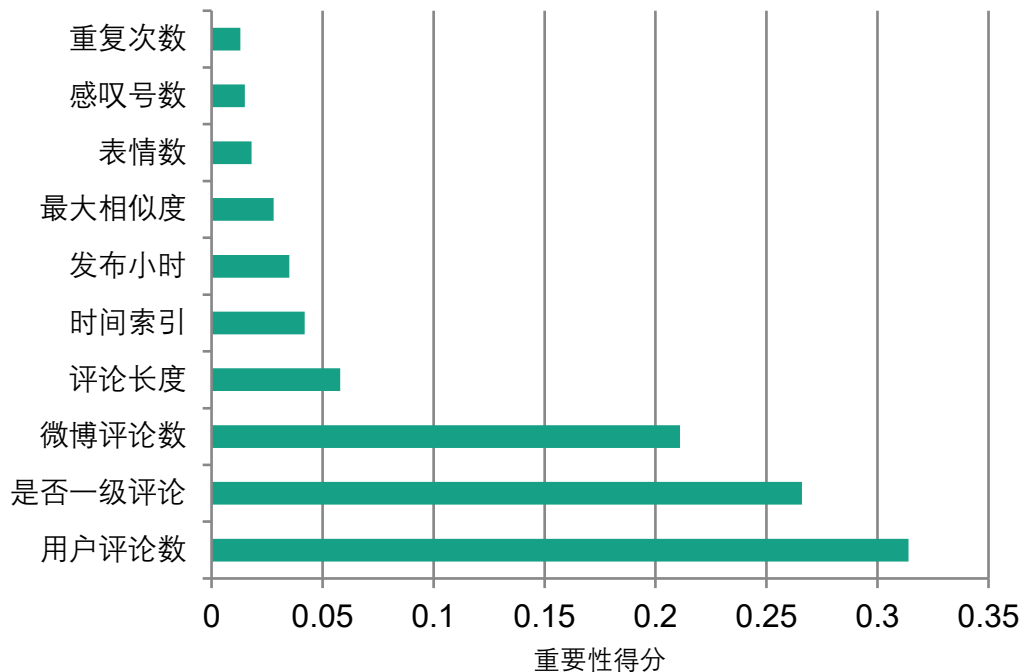
$$L = \frac{1}{2} \log(\sigma^2) + \frac{(\log(y + 10) - \log(\mu + 10))^2}{2\sigma^2}$$

PICP@95% (置信区间覆盖率)
+MPIW (平均预测区间宽度)

95% 置信区间 $[\mu - 1.96\sigma, \mu + 1.96\sigma]$

$$MPIW = \frac{1}{n} \sum (\text{Upper}_i - \text{Lower}_i)$$

特征重要性分析



关键洞察

Top 3 特征占比 79%

用户活跃度

活跃用户的评论更易获关注

评论层级

一级评论曝光度更高

微博热度

热门微博下的评论互动更多

实验结果

数据集	MAE	MSLE	ACP@20%	Log NLL	PICP@95%	MPIW
训练集	0.8717	0.0324	97.91%	-2.9121	97.85%	2.9835
验证集	0.8652	0.0329	97.77%	-2.8979	97.79%	2.9842
测试集	1.0406	0.0347	97.80%	-2.8250	97.75%	3.0050

关键发现

- ACP@20% 达到 97.80%: 绝大多数预测在20%容忍范围内
- PICP@95% 接近 95%: 不确定性估计校准良好
- MPIW较窄: 范围限定精确
- 训练集与验证集差距: 泛化性能良好

结论与展望

主要贡献

1. 数据集构建

27万条小米SU7微博评论

2. 特征工程

4类17维互补特征

3. 概率预测

NGBoost + 对数NLL损失

4. 评价体系

多维指标全面评估

未来工作

用户画像

社交网络结构、历史互动模式

时序建模

捕捉热度动态演化规律

跨平台验证

推广到其他社交媒体平台

感谢聆听

欢迎提问与讨论