User Preference-aware Fake News Detection

摘要

近年来，虚假信息和假新闻对个人和社会造成了不利影响，引起了人们对假新闻检测的广泛关注。现有的大多数假新闻检测算法都侧重于挖掘新闻内容和/或周围的外生上下文，以发现欺骗性信号;而忽略了用户在决定是否传播一条假新闻时的内生偏好。确认偏差理论表明，当一条假新闻证实了用户现有的信念/偏好时，用户更有可能传播这条假新闻。

用户的历史、社交活动(如帖子)提供了有关用户对新闻偏好的丰富信息，并具有推进假新闻检测的巨大潜力。然而，在探索用户对假新闻检测的偏好方面的工作是有限的。因此，在本文中，我们研究了利用用户偏好进行假新闻检测的新问题。我们提出了一个新的框架UPFD，它通过联合内容和图形建模同时捕获来自用户偏好的各种信号。在实际数据集上的实验结果证明了该框架的有效性。

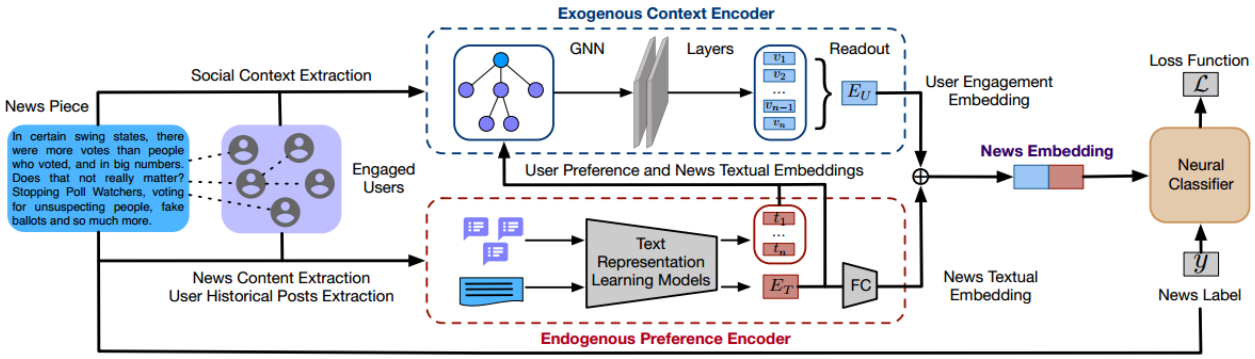
关键词：数据挖掘;社会媒体分析;假新闻检测

1.引言

由于我的研究方向是社会科学计算，该篇论文用户偏好感知假新闻检测与我的方向非常合适，同时我自己也对假新闻检测有很大的兴趣。谣言信息检测也有很大的应用价值，谣言信息如果广泛传播，往往都会造成破坏。例如：“XX社区检测到新的 COVID-19感染病例！！！”、 “XX药品是治疗新冠的特效药，来自权威期刊的最新研究！！！”等热点公共卫生事件消息，往往会引起当地和附近居民哄抢生活和医疗用品， 导致附近的大型超市和药店出现哄抢事件。哄抢行为会造成真正需要这些生活和医疗用品的人无法及时获取。这是一种典型的通过激起人们内心恐惧而传播的谣言，多数 参与者虽然心怀正义却不知不觉成了它的帮凶。如果不及时识别和制止，自然会扩散到更大的范围。因此，谣言处理得越及时，造成的危害就越小。

2.相关工作

提出了一个端到端的假新闻检测框架，名为用户偏好感知假新闻检测(UPFD)，以联合建模内生偏好和外生上下文(如下图所示)。具体而言，UPFD由以下主要组成部分组成:①内生偏好编码器 ②外生内容编码器 ③二者信息融合的部分[3,7]，最后将得到的News Embedding(新闻嵌入)通过二分类器得到最终的判断结果。



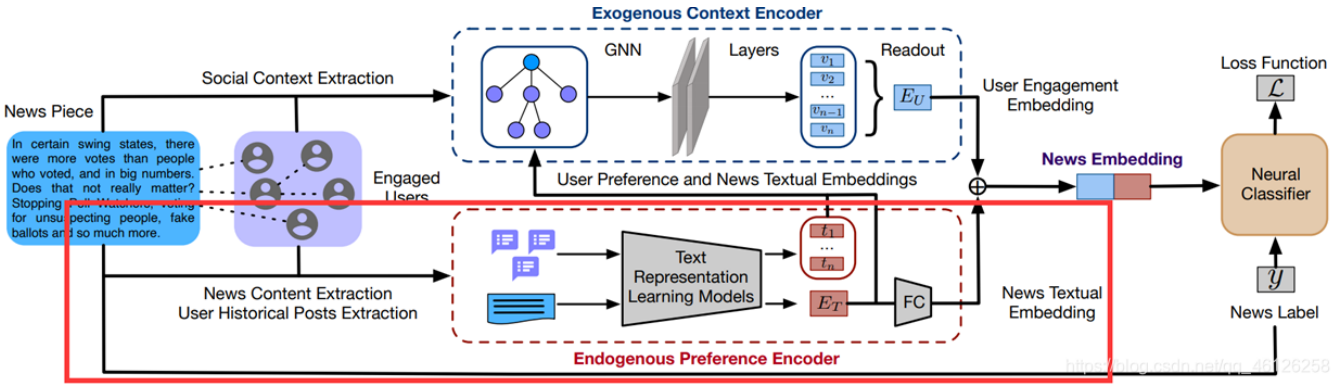
本文主要贡献可以概括如下:1.研究了一种基于用户偏好的社交媒体假新闻检测新问题;2.提出了一种原则性的方法，同时利用内生偏好和外生语境来检测假新闻;3.在真实世界的数据集上进行了大量实验，以证明UPFD在检测假新闻方面的有效性。

3.本文方法

3.1本文方法概述

3.1.1内生偏好编码器

在此部分会得到新闻内容、相关用户的偏好表示，将这两部分进行拼接得到news Texual Embedding(新闻文本嵌入)[1,5,8]。



相关用户的偏好表示：

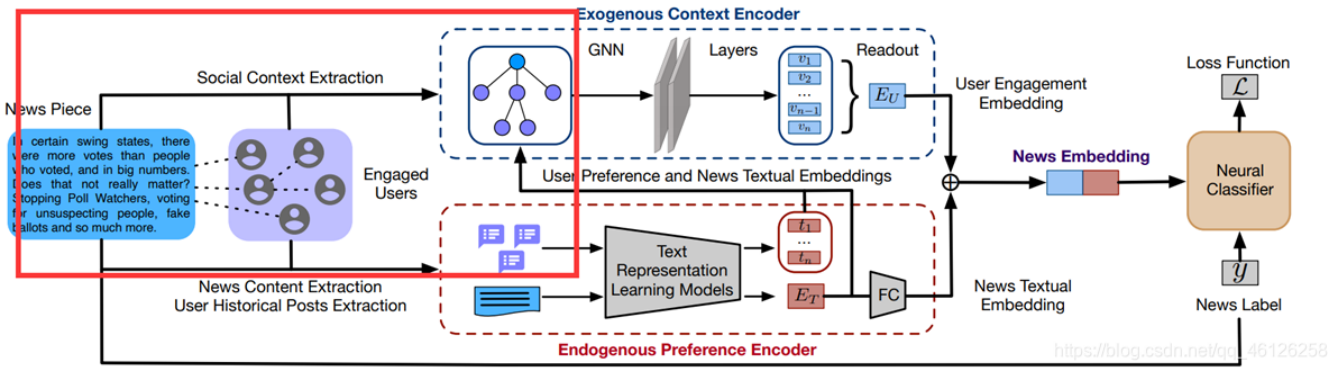
首先在 FakeNewsNet[9] 数据集中找到该新闻对应的用户在twitter上的社交参与信息，根据此信息在twitter上爬取200个他们之前发过的帖子，共得到2000w的推文。对于已被注销（不可访问）的用户，使用随机抽样可访问用户的推文来代替[2]，同时控制变量使这些抽取的用户参与的新闻与其相应的历史帖子相同。这里用了两种预训练模型：word2vec 在spaCy语料库选取68w单词的预先训练向量，合并该用户的200条帖子再对其向量进行平均以获得用户偏好表示。对于BERT模型，将对200条历史推文分别进行编码再进行平均得到偏好表示。

新闻内容的向量表示：

在两个模型中直接编码即可。

3.1.2外生内容编码器

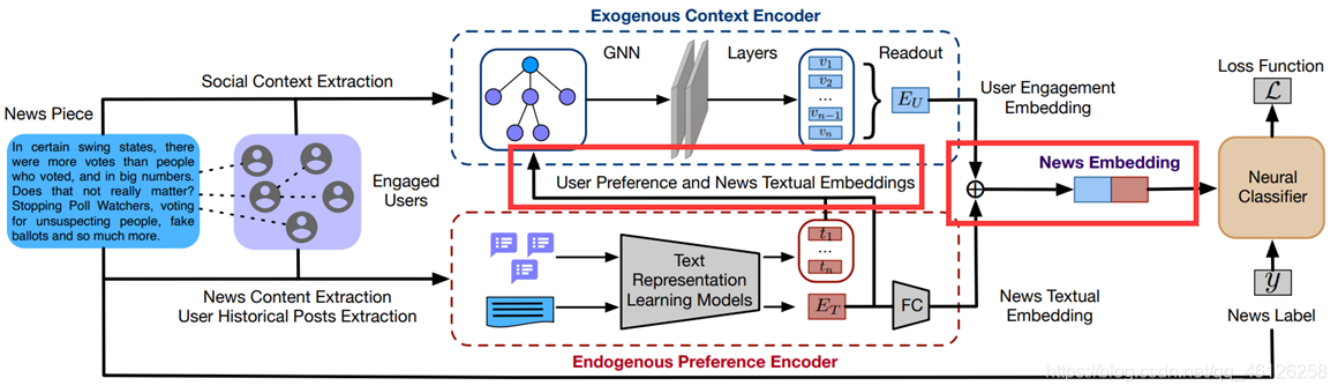
在此部分我们利用图神经网络得到新闻的传播路径对应的图嵌入。[4,7,10]



我们按照时间顺序构建传播图，其中根节点v1表示新闻片段，其他节点{v2,…,vn}表示共享根新闻的用户，树形关系则表示用户之间的转发关系。还记得上一步部分对不可访问的用户推文进行了随机抽取来做代替，实际上其主要目的是为了配合融合部分，防止直接删除破坏传播图的级联关系（将该用户删掉则其子节点的用户也会被删除）。

3.1.3二者信息融合

此部分的目的是将3.1.1和3.1.2得到的内容进行融合[4,6,7]，得到User Engagement Embedding(用户参与嵌入)。融合主要包含两个部分，如下：



已知在3.1.1得到了新闻内容、以及用户的偏好表示，在3.1.2我们构建出了新闻的传播图：其中根节点v1表示新闻片段，其他节点{v2,…,vn}表示共享根新闻的用户。所以在这部分将对应的新闻内容、用户的偏好表示作为节点的特征向量，使传播图融合用户及新闻的信息。经络两层图卷积层后在经过readout函数，readout函数对所有节点嵌入进行平均池操作，得到图嵌入(即User Engagement Embedding)。

其次，由于新闻内容通常包含了关于新闻可信度的更明确信号，所以我们将新闻文本嵌入和用户参与嵌入串联起来作为最终的新闻嵌入，来丰富新闻嵌入信息（实际上就是加大了文本嵌入的所占权重）。

4.复现细节

4.1与已有开源代码对比

与已有的开源代码对比，我多使用了个CNN模型来增加对比基准。

4.2实验环境搭建

实验环境需满足Python>=3.6, PyTorch>=1.6, and PyTorch-Geometric>=1.6.1

keras>=2.2.4，scikit-learn>=0.22.1，tqdm>=4.31.1，numpy>=1.19.4，scipy>=1.5.2

4.3界面分析与使用说明

使用文本输出界面，输出检测结果。

```
1532  90%|████████▉ | 72/80 [01:42<00:11,  1.41s/it]loss_train: 0.2181, acc_train: 0.9973,
      recall_train: 0.9980, auc_train: 1.0000, loss_val: 0.3915, acc_val: 0.9707, recall_val: 0.
      9762, auc_val: 0.9955
1533
1534  91%|█████████ | 73/80 [01:43<00:09,  1.40s/it]loss_train: 0.2315, acc_train: 0.9963,
      recall_train: 0.9957, auc_train: 0.9999, loss_val: 0.5389, acc_val: 0.9615, recall_val: 0.
      9859, auc_val: 0.9956
1535
1536  92%|█████████▏| 74/80 [01:45<00:08,  1.41s/it]
1537  94%|█████████▍| 75/80 [01:46<00:07,  1.41s/it]loss_train: 0.3323, acc_train: 0.9881,
      recall_train: 0.9904, auc_train: 0.9996, loss_val: 0.6958, acc_val: 0.9414, recall_val: 0.
      8941, auc_val: 0.9955
1538  loss_train: 0.6515, acc_train: 0.9762, recall_train: 0.9818, auc_train: 0.9966, loss_val: 0.
      8279, acc_val: 0.9341, recall_val: 0.8760, auc_val: 0.9949
1539
1540  95%|█████████▍| 76/80 [01:48<00:05,  1.41s/it]loss_train: 0.6278, acc_train: 0.9771,
      recall_train: 0.9788, auc_train: 0.9977, loss_val: 0.5890, acc_val: 0.9579, recall_val: 0.
      9897, auc_val: 0.9963
1541
1542  96%|█████████▌| 77/80 [01:49<00:04,  1.41s/it]loss_train: 0.1854, acc_train: 0.9936,
      recall_train: 0.9923, auc_train: 0.9999, loss_val: 0.4680, acc_val: 0.9689, recall_val: 0.
      9830, auc_val: 0.9951
1543
1544  98%|█████████▊| 78/80 [01:50<00:02,  1.40s/it]
1545  99%|█████████▉| 79/80 [01:52<00:01,  1.40s/it]loss_train: 0.1776, acc_train: 0.9936,
      recall_train: 0.9945, auc_train: 0.9999, loss_val: 0.4251, acc_val: 0.9670, recall_val: 0.
      9620, auc_val: 0.9947
1546  loss_train: 0.1272, acc_train: 0.9991, recall_train: 1.0000, auc_train: 1.0000, loss_val: 0.
      3894, acc_val: 0.9725, recall_val: 0.9762, auc_val: 0.9952
1547
1548  100%|██████████| 80/80 [01:53<00:00,  1.40s/it]
1549  gossipcop sage Testing Results:
1550  acc: 0.9754, f1_macro: 0.9753, f1_micro: 0.9754, precision: 0.9759, recall: 0.9749, auc: 0.
      9945, ap: 0.9945
1551
```

图1. 操作界面示意

4.4创新点

多使用了个CNN模型来增加对比基准

5.实验结果分析

UPFD和6种模型的假新闻检测性能：

```
politifact Original GCNFN Testing Results:
acc: 0.8688, f1_macro: 0.8681, f1_micro: 0.8688, precision: 0.8977, recall: 0.8406,
auc: 0.9276, ap: 0.9365
 politifact GNNCL Testing Results:
 acc: 0.6018, f1_macro: 0.5922, f1_micro: 0.6018, precision: 0.6477, recall: 0.4646,
 auc: 0.7257, ap: 0.7293
politifact UPFD GCNFN Testing Results:
acc: 0.8235, f1_macro: 0.8224, f1_micro: 0.8235, precision: 0.8844, recall: 0.7515,
auc: 0.8869, ap: 0.8864

  politifact BiGCN Testing Results:
  acc: 0.8326, f1_macro: 0.8318, f1_micro: 0.8326,precision: 0.8715, recall: 0.7867,
  auc: 0.8783, ap: 0.8888

  politifact gcn Testing Results:
  acc: 0.8190, f1_macro: 0.8182, f1_micro: 0.8190, precision: 0.8444, recall: 0.7845,
  auc: 0.8913, ap: 0.9110

 politifact gat Testing Results:
 acc: 0.8281, f1_macro: 0.8269, f1_micro: 0.8281, precision: 0.8852, recall: 0.7598,
 auc: 0.8881, ap: 0.8868
 politifact sage Testing Results:
 acc: 0.8462, f1_macro: 0.8453, f1_micro: 0.8462, precision: 0.8897, recall: 0.7950,
 auc: 0.8859, ap: 0.8862

gossipcop GNNCL Testing Results:
acc: 0.9360, f1_macro: 0.9356, f1_micro: 0.9360, precision: 0.9083, recall: 0.9705,
auc: 0.9735, ap: 0.9637

 gossipcop Original GCNFN Testing Results:
 acc: 0.9590, f1_macro: 0.9588, f1_micro: 0.9590, precision: 0.9578, recall: 0.9607,
 auc: 0.9881, ap: 0.9889

gossipcop UPFD GCNFN Testing Results:
acc: 0.9611, f1_macro: 0.9609, f1_micro: 0.9611, precision: 0.9549, recall: 0.9683,
auc: 0.9895, ap: 0.9904

gossipcop BiGCN Testing Results:
acc: 0.9127, f1_macro: 0.9123, f1_micro: 0.9127,precision: 0.9267, recall: 0.8971,
auc: 0.9680, ap: 0.9668
```

```
gossipcop gcn Testing Results:
acc: 0.9511, f1_macro: 0.9509, f1_micro: 0.9511, precision: 0.9647, recall: 0.9364,
auc: 0.9864, ap: 0.9856

gossipcop gat Testing Results:
acc: 0.9652, f1_macro: 0.9651, f1_micro: 0.9652, precision: 0.9545, recall: 0.9776,
auc: 0.9928, ap: 0.9927

gossipcop sage Testing Results:
acc: 0.9754, f1_macro: 0.9753, f1_micro: 0.9754, precision: 0.9759, recall: 0.9749,
auc: 0.9945, ap: 0.9945
```

图 2. 实验结果示意

UPFD在两个数据集上的表现都优于最佳基线GCNFN约1%。
得出结论：使用用户的历史帖子作为偏好表示，并将其作为图神经网络的节点特征可提高假新闻的检测效果。

6.总结与展望

尽管取得了良好的实验结果，但这篇论文也可能存在一些局限性。例如，对于用户偏好的建模可能仍然存在一些挑战，尤其是在处理大规模数据时。此外，该研究的数据集和实验设置也可能对结果产生影响。在未来的研究中，可能需要进一步考虑如何更好地建模用户偏好，以及如何处理不同领域或文化中用户的差异性。此外，与其他假新闻检测方法的比较和对比也是一个值得关注的方向，以验证该方法的相对优势

参考文献

[1] Nadeem Ahmad and Jawaid Siddique. 2017. Personality assessment using Twittertweets. Procedia computer science 112 (2017), 1964−1973.

[2] Twitter Developer. 2021. Twitter API. https://developer.twitter.com/.

[3] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In NeurIPS.

[4] Yi Han, Shanika Karunasekera, and Christopher Leckie. 2020. Graph Neural Networks with Continual Learning for Fake News Detection from Social Media.arXiv preprint arXiv:2007.03316 (2020).

[5] Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, and Mark Carman. 2015.Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis. 25−30.

[8] Jing Qian, Mai ElSherief, Elizabeth M Belding, and William Yang Wang. 2018.Leveraging intra-user and inter-user representation learning for automated hate speech detection. arXiv preprint arXiv:1804.03124 (2018).

[9] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu.2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data 8,3 (2020), 171−188.

[10] Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2020. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In Proceedings of the International AAAI Conference on Web and Social Media.