Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

Using Sentiment Representation Learning to Enhance Gender Classification for User Profiling

刘宇　硕士

2018年12月02日

内容提要

- 背景简介
- 算法原理
  - 原理框架
  - 数据预处理
  - 情感表征学习
  - 性别分类
- 实验结果
- 参考文献

预期收获

- 预期收获
  - 1. 了解用户画像的基本概念
  - 2. 理解情感表征学习
  - 3. 理解迁移学习的应用

背景简介

- 用户画像
  - 用户信息标签化，利用机器学习等技术预测用户的基本属性（如人口统计学属性、兴趣属性等）
  - 使用高度精炼的特征对用户进行标签化
  - 用户画像是互联网时代实现精准化服务、营销和推荐的必经之路，在网络安全、管理和运营等领域也具有重要意义
- 性别标签
  - 性别是区分用户的一个重要属性
  - 个性化推荐产品，过滤不符性别的产品
  - 推荐特定性别的相关内容

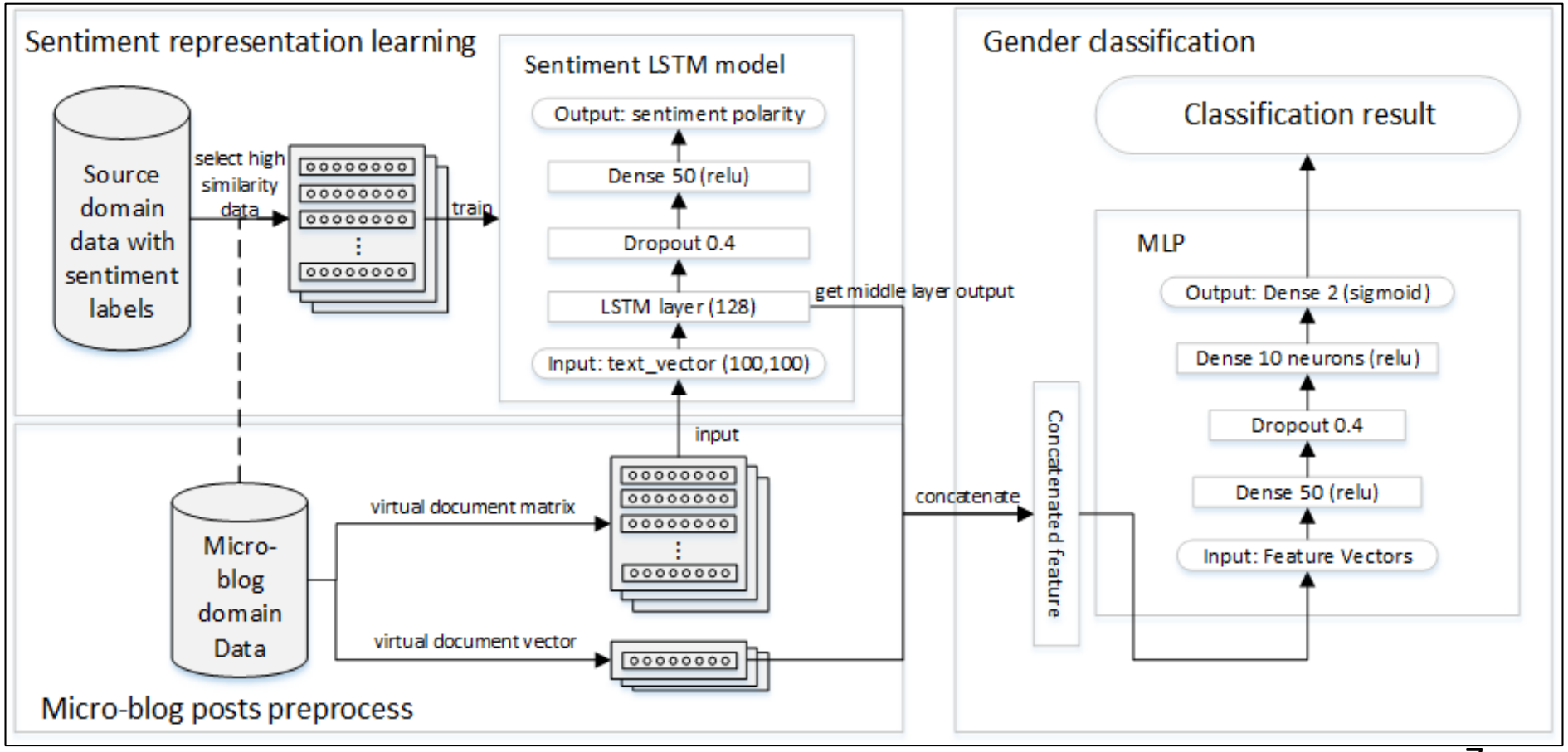
背景简介

- 情感分析
  - 情感分析多用于公开意见分析以及政治倾向分析等任务
  - 不同性别之间存在着情感差异
    - 在社交网络平台上，女性通常比男性表现的更加积极且情感更加丰富
    - 女性更喜欢展示好心情，男性则更倾向于平白直述
    - 女性比男性具有更大的情感跨度
- 通过融合情感表征来增强对用户画像中性别属性的构建

算法原理

算法原理—原理框架

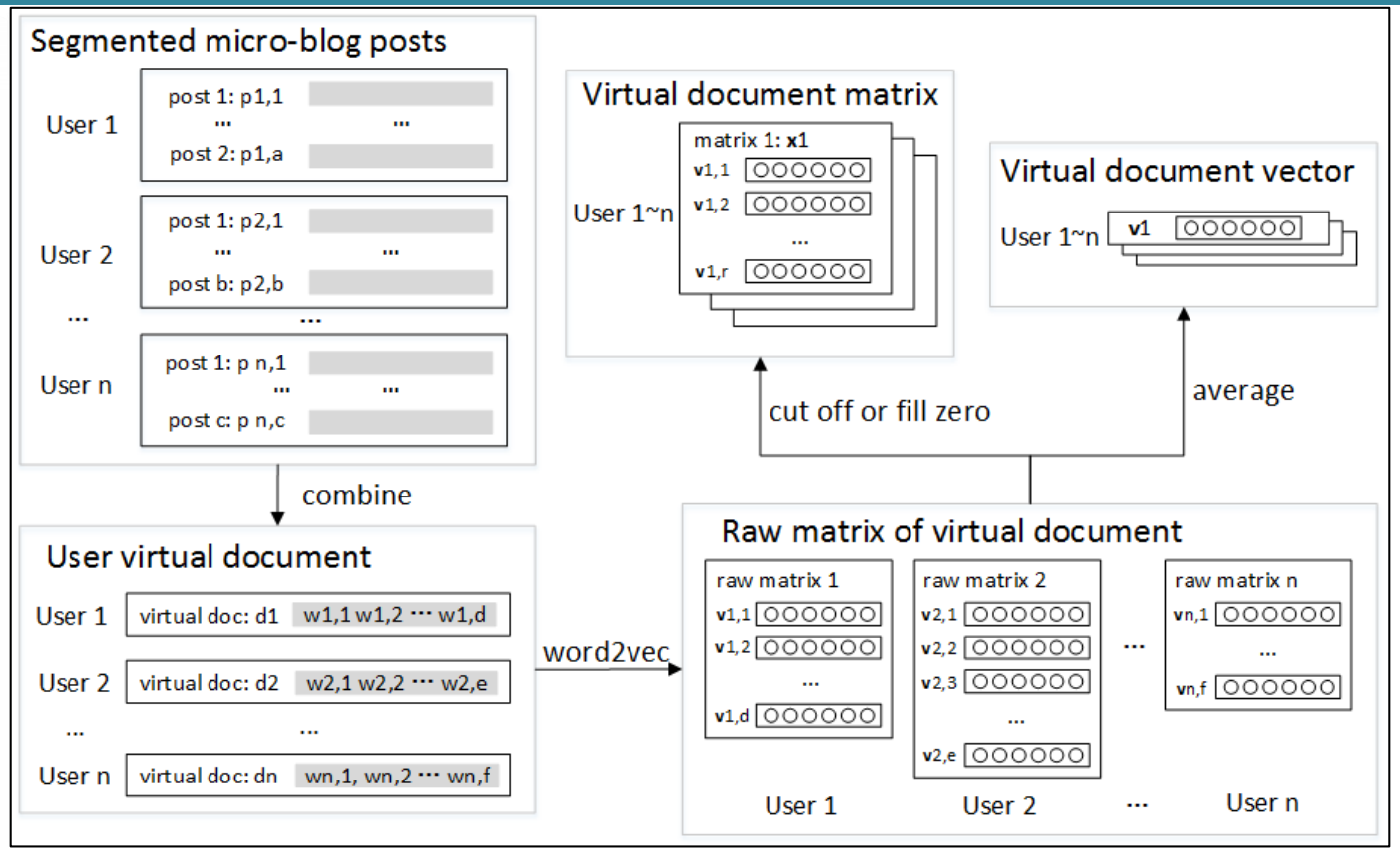
- 原理框架图
  - 主要有三部分：数据预处理、情感表征学习、性别分类



算法原理—数据预处理

- 每个用户的微博数量不同，每条微博的词数不同（上限140词），对于性别分类过短，同一用户的所有微博组合成一个虚拟文档，每个用户对应一个虚拟文档
- 文档表示
  - 虚拟文档矩阵
  - 虚拟文档向量

算法原理—数据预处理



T	对微博语料进行预处理，使其符合后续模型的输入
I	用户的原始微博语料数据集
P	采用word2vec，将微博语料转换为向量
O	虚拟文档矩阵、虚拟文档向量

算法原理—情感表征学习

- 如何获得社交媒体数据中准确的情感标签?
  - 手工标记：成本高，不切实际
  - 迁移学习：利用与目标域数据相近的带标签源域数据构建一个情感表征模型，通过该模型得到目标域数据的情感标签

- 源域数据的选择
  - 商品评论通常具有短文本特性，且带有1~5星的打分可以用于表征用户情感（积极、消极）

算法原理—情感表征学习

- 提高源域与目标域数据的相似度
  - 筛选源域数据中与目标域数据具有高相似度的数据，需要计算相似度

$$D_S \leftarrow \left\{ x_k, y_k \mid \frac{1}{n}\sum_{i=1}^{n} SIM\left(v_k, v_i\right) > z, z \in (0,1) \right\}$$

  - 将部分目标域数据进行手动标记，然后将其添加到源域数据

$$D_S \leftarrow \left\{ x_{mlt}, y_{mlt} \mid x_{mlt} \in D_t \right\}$$

- 情感表征
  - 深度神经网络中特征是可迁移的
  - 源域和目标域模型的参数是共享的
  - 除了最终的情感极性结果，还可以将模型的中间层作为情感表征

算法原理—情感表征学习

- 模型训练

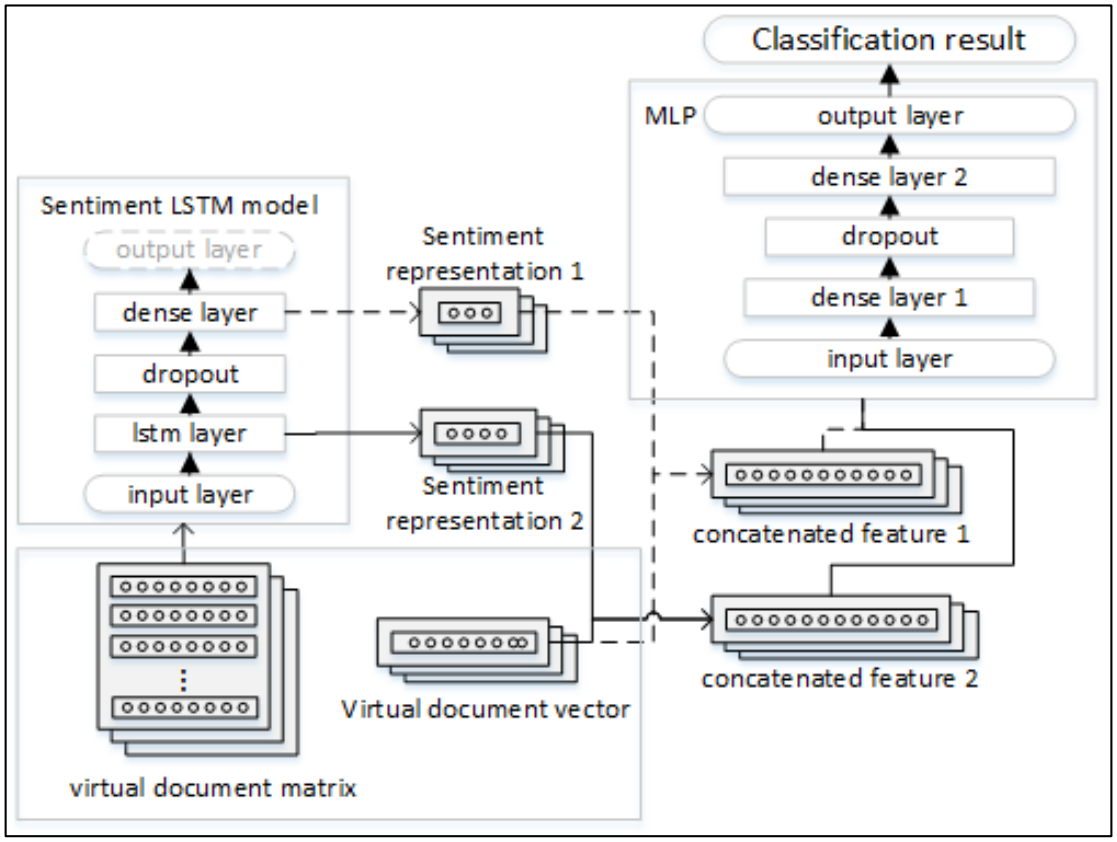
T	基于有标签评论数据，获得一个情感表征模型
I	与目标域数据高度相似的商品评论数据集
P	采用LSTM算法构建一个情感表征模型
O	文本情感表征模型

- 情感表征

T	基于无标签微博数据，获得对应的情感特征
I	经过预处理的用户虚拟文档矩阵
P	利用上面训练的模型，获取用户的情感特征
O	用户对应的情感特征向量

算法原理—性别分类

T	对微博用户进行性别分类
I	用户虚拟文档向量与情感表征的级联
P	利用MLP模型，进行性别分类
O	微博用户的性别（0：男性，1：女性）



算法原理

Algorithm 1 SRL-MLP

Input:

Segmented user micro-blog posts $p_{i,j} = \{w_{i,j}^{(1)}, w_{i,j}^{(2)}, ..., w_{i,j}^{(c_{i,j})}\}$, i=1,2,...n;

Segmented source domain documents $d_k = \{w_k^{(1)}, w_k^{(2)}, ..., w_k^{(c_k)}\}$, k=1,2,...,m;

Output:

Output: Gender of user i, G_i.

1: Documents vector representation;

1.1 Assemble user posts to form user virtual document:

$d_i' = \{p_{i,1}, p_{i,2}, ..., p_{i,j}, ..., p_{i,a}\}, i = 1, 2, ..., n;$

Regardless of which post the word comes from, renumber the word:

$d_i = \{w_i^{(1)}, w_i^{(2)}, ..., w_i^{(c_i)}\}, i = 1, 2, ..., n;$

1.2 Vectorize each word in virtual document to form raw matrix of document by word2vec[8][9], including micro-blog and source domain data.

$\mathbf{x}_i' = \{\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, ..., \mathbf{v}_i^{(c_i)}\}, i = 1, 2, ..., n;\ \mathbf{x}_k' = \{\mathbf{v}_k^{(1)}, \mathbf{v}_k^{(2)}, ..., \mathbf{v}_k^{(c_k)}\}, k = 1, 2, ..., m;$

1.3 Make average micro-blog word vectors:

$\mathbf{v}_i = \frac{1}{c_i}(\mathbf{v}_i^{(1)} + \mathbf{v}_i^{(2)} + ... + \mathbf{v}_i^{c_i});\ \mathbf{v}_k = \frac{1}{c_k}(\mathbf{v}_k^{(1)} + \mathbf{v}_k^{(2)} + ... + \mathbf{v}_k^{c_i});$

1.4 Cut off or fill zero to make document matrices having same shape $d \times r$:

$\mathbf{x}_i = \{\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, ..., \mathbf{v}_i^{(r)}\}, i = 1, 2, ..., n;\ \mathbf{x}_k = \{\mathbf{v}_k^{(1)}, \mathbf{v}_k^{(2)}, ..., \mathbf{v}_k^{(r)}\}, k = 1, 2, ..., m;$

2: LSTM Sentiment representation learning;

2.1 Select high similarity source domain data to be new source domain data $\mathcal{D}_s$:

$\mathcal{D}_s \leftarrow \{\mathbf{x}_k, y_k | \frac{1}{n}\sum_{i=1}^{n} SIM(\mathbf{x}_i, \mathbf{x}_k) > a, a \in (0, 1)\};$

2.2 Using source domain data $\mathcal{D}_s$ to train a LSTM sentiment model.

2.3 Put target domain data $\mathcal{D}_t = \{x_i\}, i = 1, 2, ..., n$, into LSTM, for each $\mathbf{x}_i$ get its lstm layer output $\mathbf{h}_i$ to be our target domain sentiment representation:

3: MLP gender classification;

3.1 Concatenate $\mathbf{v}_i$ and $\mathbf{h}_i$ to be final features $\mathbf{f}_i$:

$\mathbf{f}_i = (\mathbf{v}_i^T, \mathbf{h}_i^T)^T;$

3.2 Input $\mathbf{f}_i$ to MLP, get G_i:

$G_i = f(W\mathbf{f}_i + b);$

4: **return** G_i;

实验结果

实验结果

- 评价指标
  - 准确率，正确分类的样本占总样本的比例

- 实验一：文档表示方式的对比

Word representation	Accuracy(%)
TF-IDF	82.71
Keywords TF-IDF	80.49
LDA	79.15
Average word2vec	**84.20**
Average word2vec + LDA	84.01

- 实验二：性别分类器的对比

Classifiers	Accuracy(%)
Logistic Regression	67.06
Random Forest	72.15
Support Vector Machine	76.34
MLP	**84.20**
CNN	74.21
LSTM	73.67

实验结果

- 实验三：融合情感表征

train LSTM on	added features	accuracy(%)
entire JD reviews	frozen lstm	88.09
	frozen dense	87.98
	finetuned lstm	86.95
high similarity JD reviews	frozen lstm	**89.73**
	frozen dense	88.45
	finetuned lstm	87.38
entire JD reviews and manually labeled micro-blog	frozen lstm	88.87
	frozen dense	88.13
	finetuned lstm	86.75
high similarity JD reviews and manually labeled micro-blog	frozen lstm	89.31
	frozen dense	89.26
	finetuned lstm	87.22

参考文献

参考文献

- Using Sentiment Representation Learning to Enhance Gender Classification for User Profiling
  - https://arxiv.org/abs/1810.06645

道德经

上善若水。水善利万物而不争，处众人之所恶，故几於道。居善地，心善渊与善仁，言善信，正善治，事善能，动善时。夫唯不争，故无尤。

谢谢！

- 情感表征模型构建代码

```
print('Build model...')
model = Sequential()
model.add(Embedding(len(dict)+1, 256))
model.add(LSTM(128)) # try using a GRU instead, for fun
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'

model.fit(x, y, batch_size=16, nb_epoch=10) #训练时间为若干个小时

classes = model.predict_classes(xt)
acc = np_utils.accuracy(classes, yt)
print('Test accuracy:', acc)
```