基于隐含狄列克雷分配的短文本分类方法

张志飞　苗夺谦　高灿

同济大学
计算机科学与技术系

Nov. 17, 2012

汇报内容

汇报内容

短文本层出不穷


有急事，请帮我
充值，并把充值卡
密码发给我。


新浪微博
姚晨v


renren


facebook


Barack Obama
@BarackObama
Following
We're all in this together. That's how we campaigned, and that's who we are. Thank you. -bo
Reply
Retweeted
Favorite
11,208
3,558

主要困难

短文本挖掘面临的两大困难:

- 特征非常稀疏
- 上下文依赖性强

仅根据词语的共现程度不能很好地度量文本之间的相似性!

主要困难

短文本挖掘面临的两大困难:

- 特征非常稀疏
- 上下文依赖性强

仅根据词语的共现程度不能很好地度量文本之间的相似性!

解决方法

主要有两类解决方法:

- 借助外部文本扩展，如搜索引擎
  - 消耗时间长，且非常依赖搜索结果
- 借助知识库扩展，如WordNet、Wikipedia
  - 对于知识库中不存在的词无能为力

还有借助主题模型，如Latent Dirichlet Allocation

- 利用主题挖掘词语之间的关系

解决方法

主要有两类解决方法:

- 借助外部文本扩展，如搜索引擎
  - 消耗时间长，且非常依赖搜索结果
- 借助知识库扩展，如WordNet、Wikipedia
  - 对于知识库中不存在的词无能为力

还有借助主题模型，如Latent Dirichlet Allocation

- 利用主题挖掘词语之间的关系

汇报内容

向量空间模型

Vector Space Model(SVM)由Salton等在1968年提出，满足“词袋”假设。

符号定义

- 词汇表V: $V = \{v_1, v_2, ..., v_N\}$
- 文本集D: $D = \{d_1, d_2, ..., d_M\}$
- 文本$d_i \in D$的向量表示: $V^{(i)} = (w_1^{(i)}, w_2^{(i)}, ..., w_N^{(i)})$

向量中的特征权重通常采用TF-IDF评估，

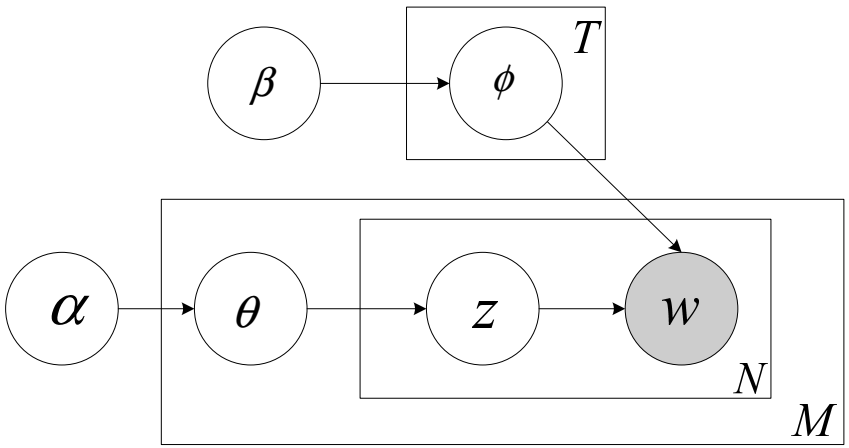
$$w_k^{(i)} = tf_{ki} \times \log \frac{M}{df_k} \tag{1}$$

采用余弦距离度量文本之间的相似度，

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{N} w_k^{(i)} \times w_k^{(j)}}{\sqrt{\sum_{k=1}^{N} (w_k^{(i)})^2} \times \sqrt{\sum_{k=1}^{N} (w_k^{(j)})^2}} \tag{2}$$

隐含狄列克雷分配

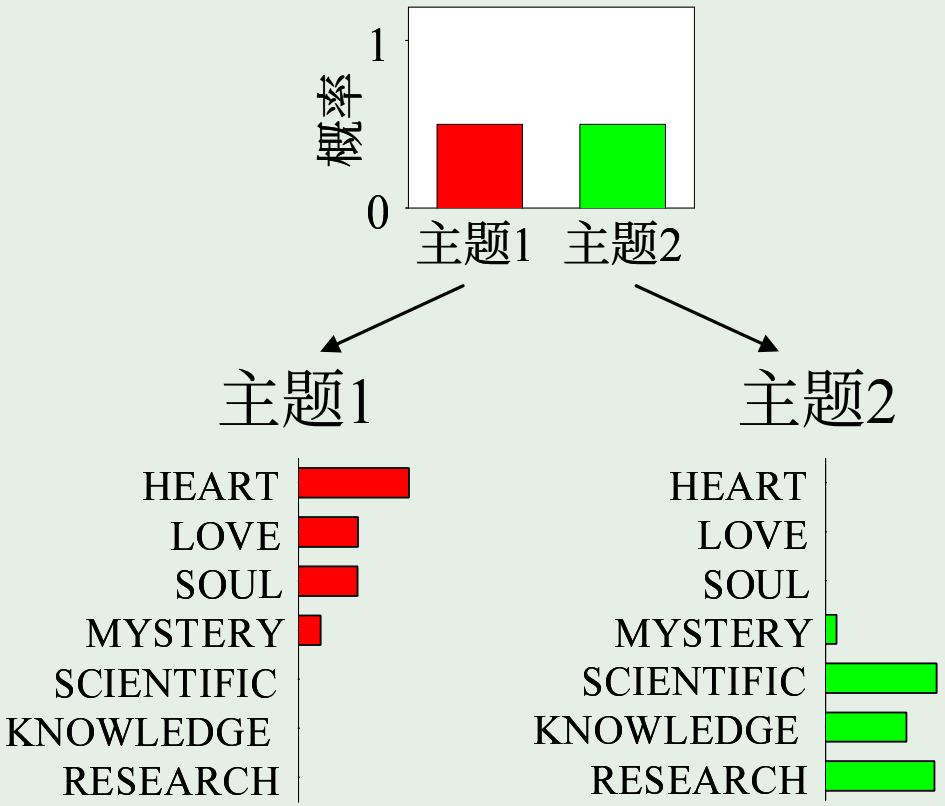
Latent Dirichlet Allocation(LDA)由Blei等在2003年提出，是“文本–主题–词”的三层贝叶斯产生式模型。



符号	含义
α	θ的超参数
β	ϕ的超参数
θ	文本的主题概率分布
ϕ	主题的词概率分布
z	词的主题分配
w	词
M	文本数
N	词数
T	主题数

隐含狄列克雷分配(续)

例子(文本生成过程)



一篇文本: LOVE SCIENTIFIC SOUL LOVE KNOWLEDGE ...

隐含狄列克雷分配(续)

对变量z进行Gibbs抽样，间接计算出两个概率分布θ和ϕ。

LDA参数估计

$$\theta_{mk} = \frac{n_m^{(k)} + \alpha}{\sum_{j=1}^{T} n_m^{(j)} + T\alpha}$$

$$\phi_{kt} = \frac{n_k^{(t)} + \beta}{\sum_{i=1}^{N} n_k^{(i)} + N\beta}$$

注:

(1) $n_m^{(j)}$为文本d_m中赋予主题j的词的总数，$n_k^{(i)}$为词v_i被赋予主题k的总次数。

(2) 通常两个超参数设置为：$\alpha = \frac{50}{T}$，$\beta = 0.01$。

基于主题的相似性

Topic Based Similarity(TBS)由Quan等在2010年提出，借助“主题”比较两篇短文本，解决稀疏性问题。

给定两篇短文本d_1和d_2，其向量表示为:
$V^{(1)} = (w_1^{(1)}, w_2^{(1)}, ..., w_N^{(1)})$，$V^{(2)} = (w_1^{(2)}, w_2^{(2)}, ..., w_N^{(2)})$

定义(可区分词集)

短文本d_1和d_2的可区分词集分别为:
$Dist(d_1) = \{v | v \in d_1 \wedge v \notin d_2\}$
$Dist(d_2) = \{v | v \in d_2 \wedge v \notin d_1\}$

基于主题的相似性(续)

对于每个主题$k(1 \leq k \leq T)$，v_m是$Dist(d_1)$中主题–词概率值最大的词，其值为ϕ_{km}；v_n是$Dist(d_2)$中主题–词概率值最大的词，其值为ϕ_{km}。

如果$\phi_{km} \geq \lambda$且$\phi_{kn} \geq \lambda$，那么v_m和v_n认为在该主题上非常相关。

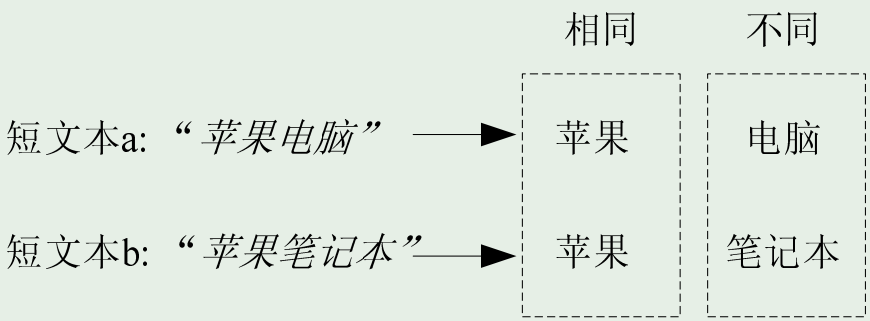
权重更新(特征稀疏性)

$$\begin{aligned} w_n^{(1)} &= w_n^{(1)} + w_n^{(2)} \times \phi_{in}, \\ w_m^{(2)} &= w_m^{(2)} + w_m^{(1)} \times \phi_{im} \end{aligned} \tag{3}$$

汇报内容

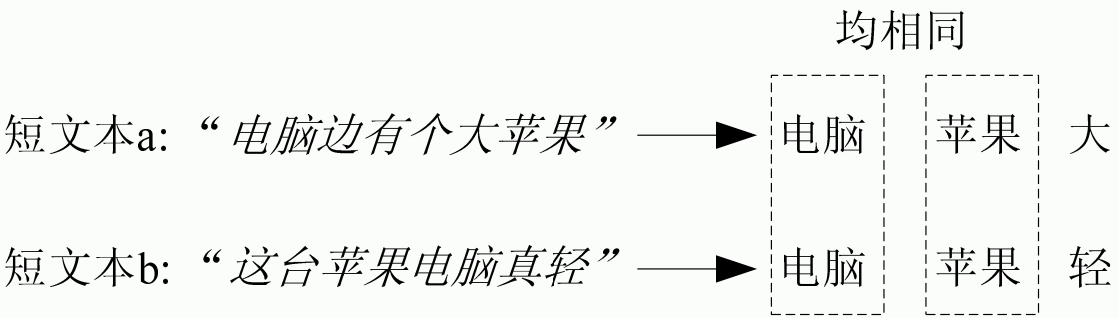
问题描述

例子(特征稀疏性)



TBS将“电脑”和“笔记本”关联起来，解决稀疏性问题。

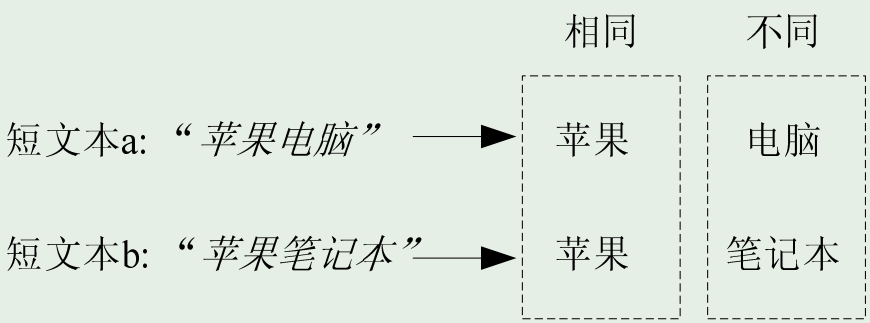
例子(上下文依赖性)



VSM和TBS均认为其相似性很大，不能解决上下文依赖性问题。

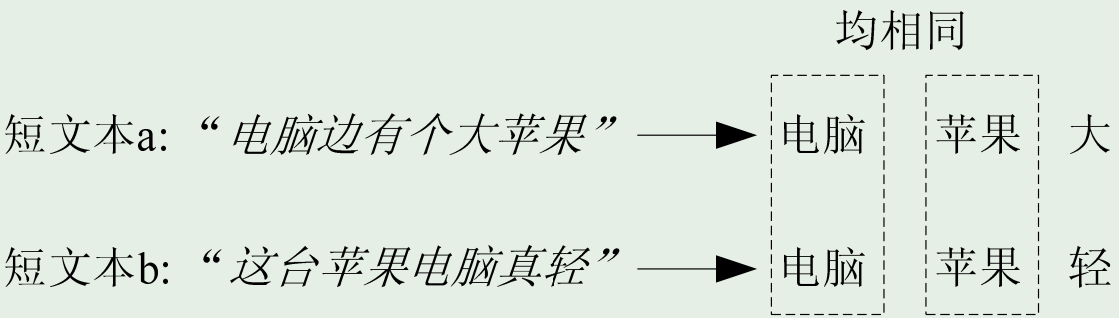
问题描述

例子(特征稀疏性)



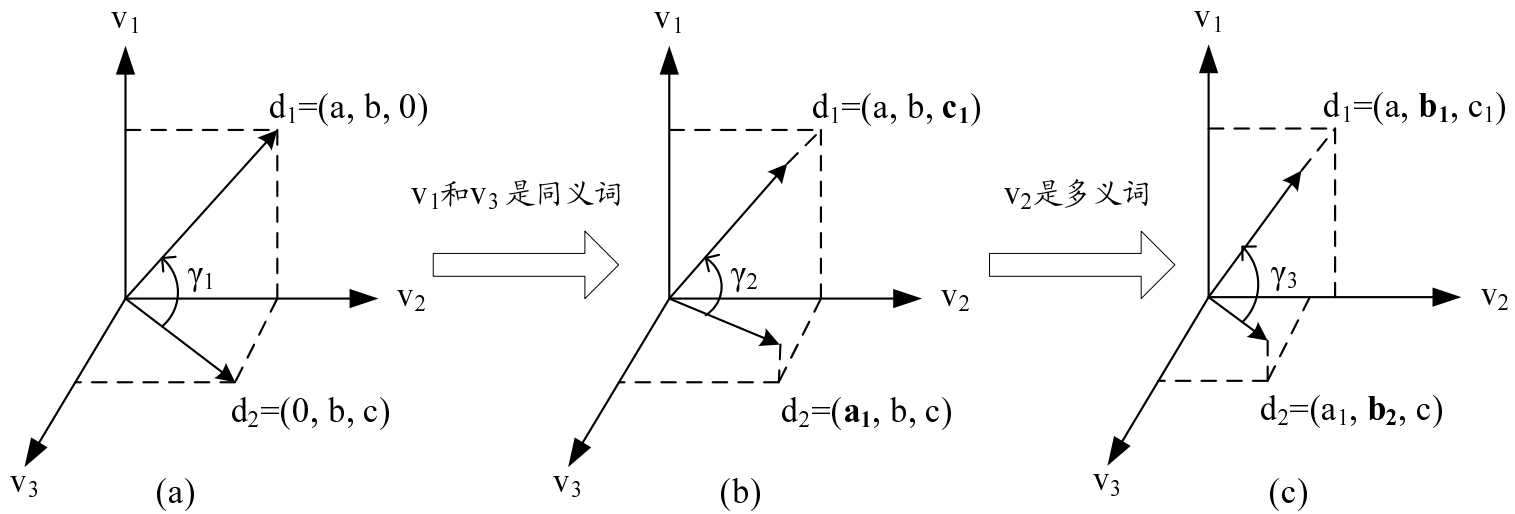
TBS将“电脑”和“笔记本”关联起来，解决稀疏性问题。

例子(上下文依赖性)



VSM和TBS均认为其相似性很大，不能解决上下文依赖性问题。

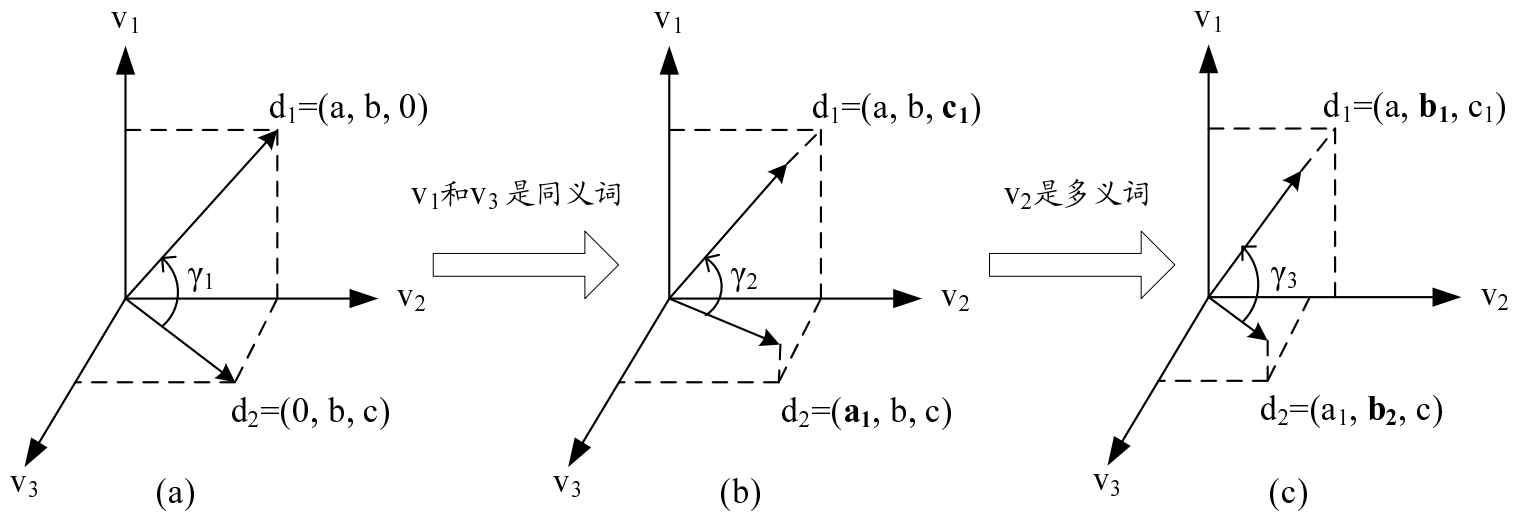
解决思路



注：γ_1、γ_2和γ_3是向量夹角；(b)，$a_1 > 0$且$c_1 > 0$；(c)，$0 < b_1 < b$且$0 < b_2 < b$。

- 若v_1和v_3是同义词，TBS增加其权重，使得$\gamma_2 < \gamma_1$，提高相似性。
- 若v_2是多义词，需要减少其权重，使得$\gamma_3 > \gamma_2$，降低相似性。

解决思路



注：γ_1、γ_2和γ_3是向量夹角；(b)，$a_1 > 0$且$c_1 > 0$；(c)，$0 < b_1 < b$且$0 < b_2 < b$。

- 若v_1和v_3是同义词，TBS增加其权重，使得$\gamma_2 < \gamma_1$，提高相似性。
- 若v_2是多义词，需要减少其权重，使得$\gamma_3 > \gamma_2$，降低相似性。

相似性度量

TBS依据LDA中的主题–词概率分布ϕ解决稀疏性问题，我们再依据LDA中的文本–主题概率分布θ解决上下文依赖性问题。

定义(共有词集)

短文本d_1和d_2的共有词集为:

$$Comm(d_1, d_2) = \{v | v \in d_1 \wedge v \in d_2\}$$

对$Comm(d_1, d_2)$中满足一定条件的词降低其权重，

条件1 根据θ，提取d_1和d_2各自的最大主题，两者不一致；

条件2 在各自的最大主题下，该词的主题–词概率值排名前60%。

相似性度量

TBS依据LDA中的主题–词概率分布ϕ解决稀疏性问题，我们再依据LDA中的文本–主题概率分布θ解决上下文依赖性问题。

定义(共有词集)

短文本d_1和d_2的共有词集为:

$$Comm(d_1, d_2) = \{v | v \in d_1 \wedge v \in d_2\}$$

对$Comm(d_1, d_2)$中满足一定条件的词降低其权重，

条件1 根据θ，提取d_1和d_2各自的最大主题，两者不一致;

条件2 在各自的最大主题下，该词的主题–词概率值排名前60%。

相似性度量

TBS依据LDA中的主题–词概率分布ϕ解决稀疏性问题，我们再依据LDA中的文本–主题概率分布θ解决上下文依赖性问题。

定义(共有词集)

短文本d_1和d_2的共有词集为:

$$Comm(d_1, d_2) = \{v | v \in d_1 \wedge v \in d_2\}$$

对$Comm(d_1, d_2)$中满足一定条件的词降低其权重，

条件1 根据θ，提取d_1和d_2各自的最大主题，两者不一致;

条件2 在各自的最大主题下，该词的主题–词概率值排名前60%。

相似性度量(续)

权重更新(上下文依赖性)

$$
\begin{aligned}
w_c^{(1)} &= \left| w_c^{(1)} - w_c^{(2)} \times \theta_{1t_{\max}^{(1)}} \times \phi_{t_{\max}^{(1)}c} \right|, \\
w_c^{(2)} &= \left| w_c^{(2)} - w_c^{(1)} \times \theta_{2t_{\max}^{(2)}} \times \phi_{t_{\max}^{(2)}c} \right|
\end{aligned}
\tag{4}
$$

其中:

$$
\begin{aligned}
t_{\max}^{(1)} &= \arg\max_k \{\theta_{1k} | 1 \le k \le T\}, \\
t_{\max}^{(2)} &= \arg\max_k \{\theta_{2k} | 1 \le k \le T\}
\end{aligned}
\tag{5}
$$

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

相似性度量(续)

新相似度度量算法

输入: 短文本$d_1 : V^{(1)}$和短文本$d_2 : V^{(2)}$;
输出: d_1和d_2的相似度$Sim(d_1, d_2)$;

1. 获取d_1和d_2的共有词集$Comm(d_1, d_2)$;
2. 如果$Comm(d_1, d_2) = \emptyset$, 转至步骤(6);
3. 根据公式(5)提取d_1和d_2各自的最大主题$t_{\max}^{(1)}$和$t_{\max}^{(2)}$;
4. 如果$t_{\max}^{(1)} = t_{\max}^{(2)}$，转至步骤(6);
5. 对于$Comm(d_1, d_2)$中的每个词v_c，如果其主题-词概率值在各自的最大主题下均排在前60%，则根据公式(4)更新权重;
6. 获取d_1和d_2的可区分词集$Dist(d_1)$和$Dist(d_2)$;
7. 如果$Dist(d_1) = \emptyset$或者$Dist(d_2) = \emptyset$, 转至步骤(10);
8. 对于每个主题，找出$Dist(d_1)$和$Dist(d_2)$中具有最大主题–词概率值的词v_m和v_n，其概率值为ϕ_{im} 和ϕ_{in};
9. 如果$\phi_{im} \geq \lambda$且$\phi_{in} \geq \lambda$，则根据公式(3)更新权重;
10. 根据公式(2)计算d_1和d_2的相似性。

算法举例

$$V^{(1)} = (0.6, 0.4, 0), \quad V^{(2)} = (0, 0.4, 0.6)$$

$$\theta = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}, \quad \phi = \begin{bmatrix} 0.3 & 0.35 & 0.35 \\ 0.2 & 0.55 & 0.25 \end{bmatrix}$$

- VSM: $\cos\gamma_1 = 0.3077$
- TBS:
  $V^{(1)} = (0.6, 0.4, 0.21)$, $V^{(2)} = (0.18, 0.4, 0.6)$, $\cos\gamma_2 = 0.7058$
- 新方法:
  $V^{(1)} = (0.6, 0.288, 0.21)$, $V^{(2)} = (0.18, 0.246, 0.6)$, $\cos\gamma_3 = 0.6491$
- $\cos\gamma_2 > \cos\gamma_3 > \cos\gamma_1$成立!

算法举例

$$V^{(1)} = (0.6, 0.4, 0), \quad V^{(2)} = (0, 0.4, 0.6)$$

$$\theta = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}, \quad \phi = \begin{bmatrix} 0.3 & 0.35 & 0.35 \\ 0.2 & 0.55 & 0.25 \end{bmatrix}$$

- VSM: $\cos\gamma_1 = 0.3077$
- TBS:
  $V^{(1)} = (0.6, 0.4, 0.21)$, $V^{(2)} = (0.18, 0.4, 0.6)$, $\cos\gamma_2 = 0.7058$
- 新方法:
  $V^{(1)} = (0.6, 0.288, 0.21)$, $V^{(2)} = (0.18, 0.246, 0.6)$, $\cos\gamma_3 = 0.6491$
- $\cos\gamma_2 > \cos\gamma_3 > \cos\gamma_1$成立!

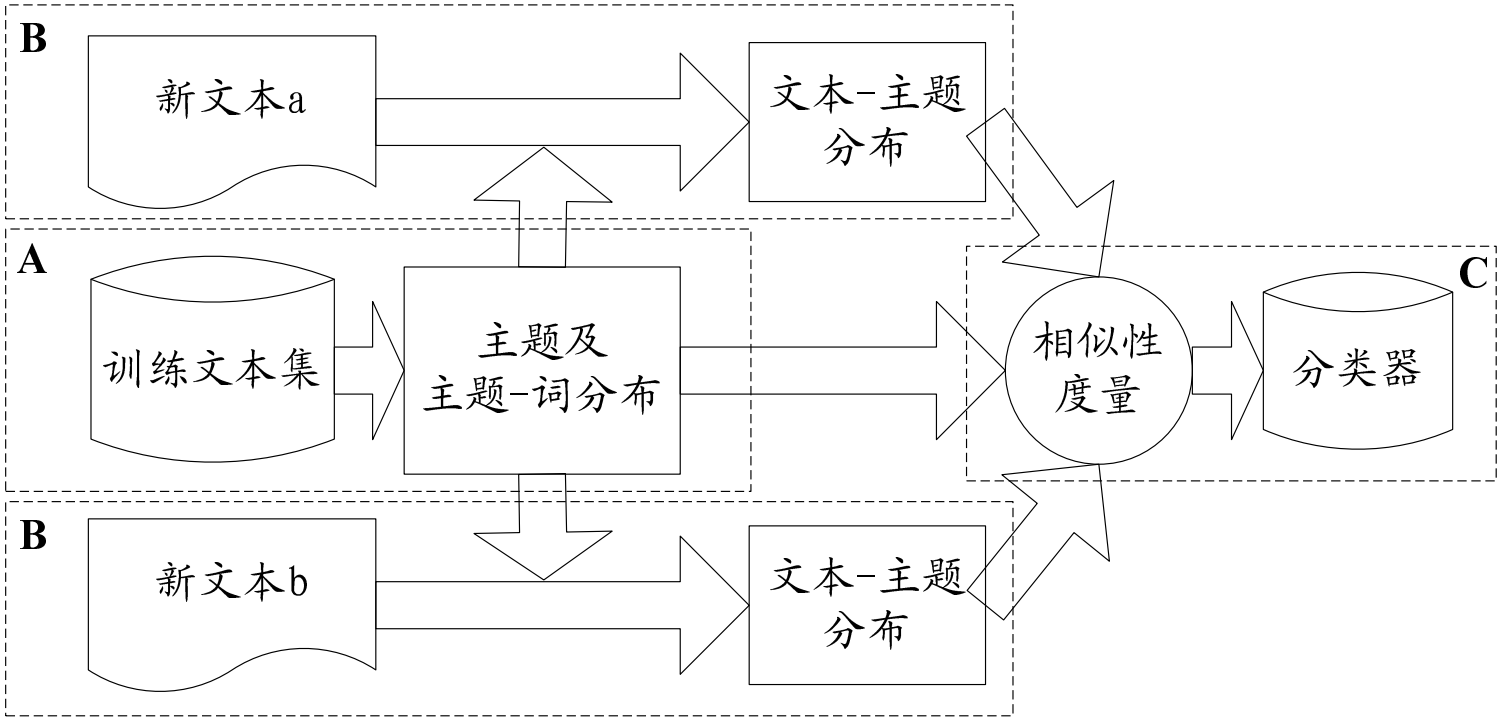
算法举例

$$V^{(1)} = (0.6, 0.4, 0), \quad V^{(2)} = (0, 0.4, 0.6)$$

$$\theta = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}, \quad \phi = \begin{bmatrix} 0.3 & 0.35 & 0.35 \\ 0.2 & 0.55 & 0.25 \end{bmatrix}$$

- VSM: $\cos\gamma_1 = 0.3077$
- TBS:
  $V^{(1)} = (0.6, 0.4, 0.21)$, $V^{(2)} = (0.18, 0.4, 0.6)$, $\cos\gamma_2 = 0.7058$
- 新方法:
  $V^{(1)} = (0.6, 0.288, 0.21)$, $V^{(2)} = (0.18, 0.246, 0.6)$, $\cos\gamma_3 = 0.6491$
- $\cos\gamma_2 > \cos\gamma_3 > \cos\gamma_1$成立!

基本框架



汇报内容

数据集

采用爬虫抓取网易页面(含标题和正文)，将页面标题作为短文本，共计5892篇。

类别	文本数
教育	518
经济	701
军事	1871
科技	505
商务	501
社会	483
体育	808
娱乐	505

实验设置

主题数的设定

将网易页面正文数据的2/3用于学习，1/3用于预测，并利用困惑度Perplexity指标确定主题数:

$$Perplexity = \exp\{-\frac{\sum_m \ln p(v_m)}{\sum_m N_m}\}$$

阈值的设定

TBS和我们的方法中均需要阈值λ，将所有主题下的最大主题–词概率值累加求平均，并以60%作为分界点:

$$\lambda = 0.6 \times \frac{1}{T} \sum_{k=1}^{T} \max_{1 \le j \le N} \phi_{kj}$$

分类器及评估

分类器

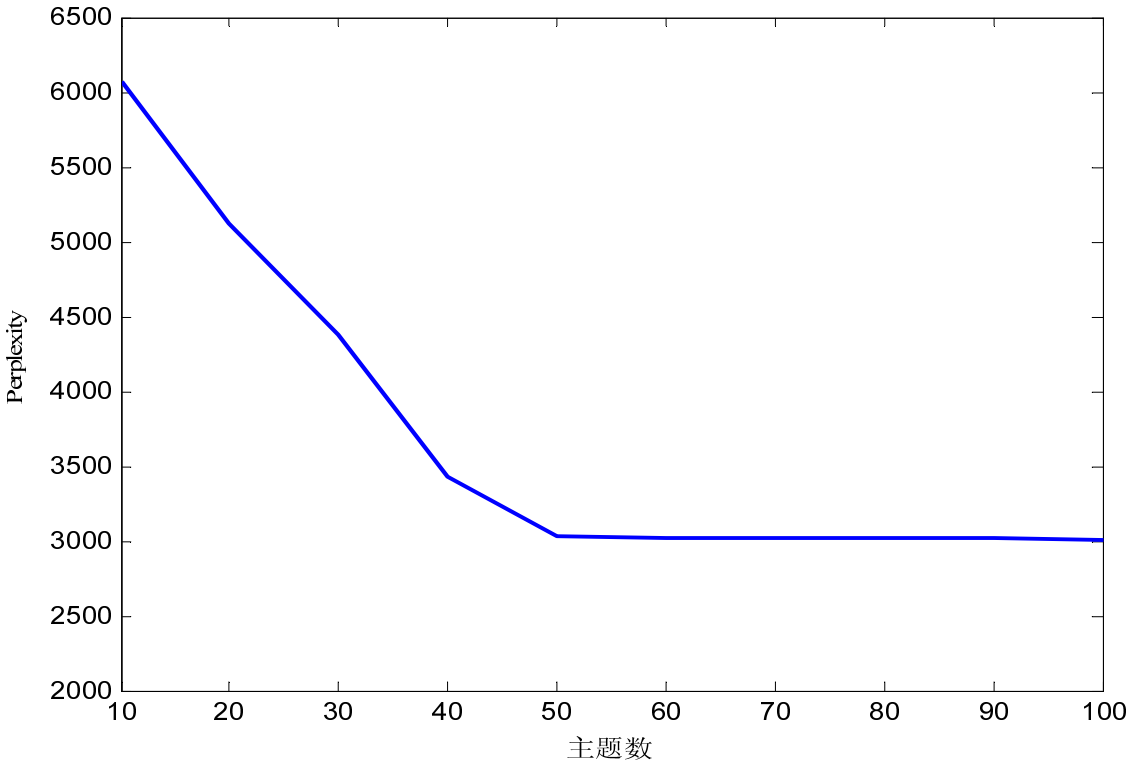
采用KNN对网易页面标题数据分类，结合五折交叉验证。KNN中的近邻数为VSM方法取得最好性能时的近邻数。

评估

采用文本分类中常见的指标来评估性能，如查全率Re和查准率Pr，而F_1是两者的综合评价(本文准确描述为宏F_1):

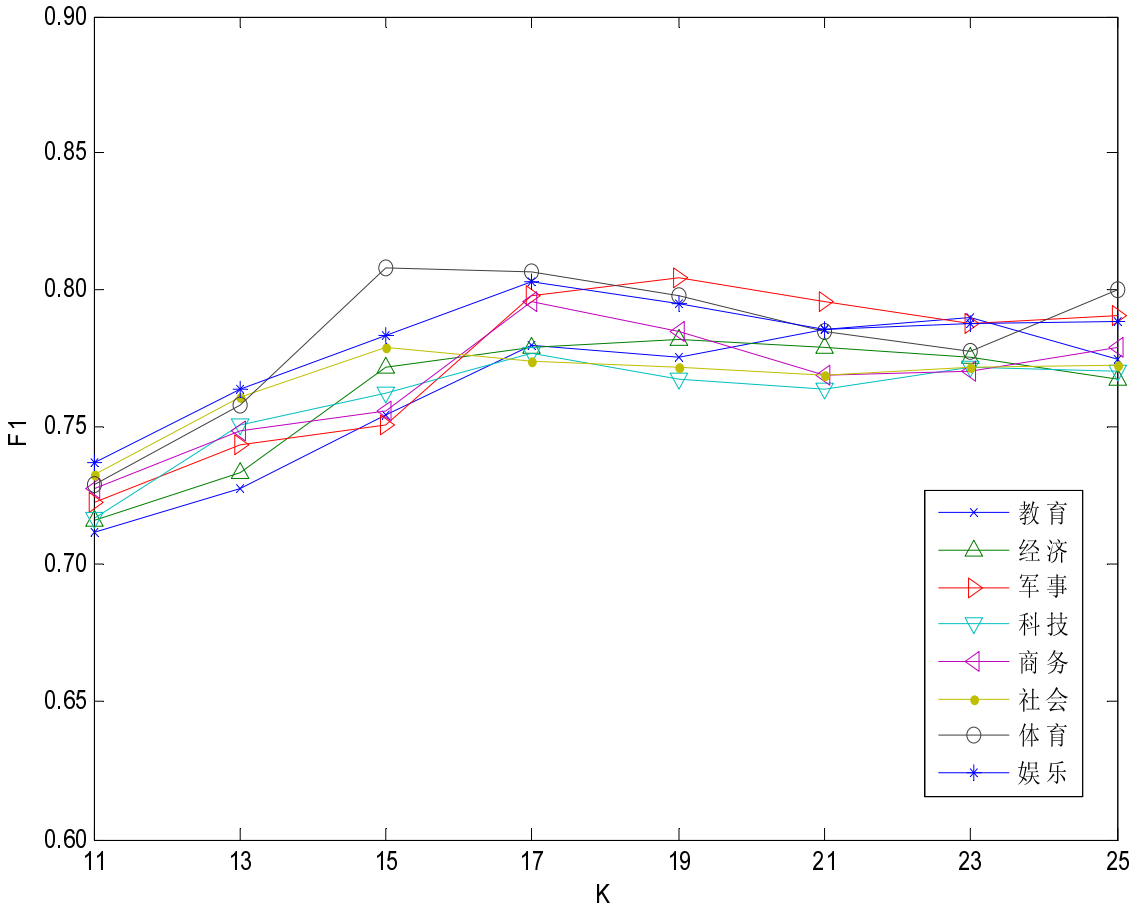
$$F_1 = \frac{2 \times Pr \times Re}{Pr + Re}$$

主题数



主题数设置为：$T = 50$

K近邻数



近邻数设置为：$K = 17$

查全率和查准率的比较

	VSM		TBS		新方法	
	Re	Pr	Re	Pr	Re	Pr
教育	0.782	0.779	0.801	0.802	0.824	0.831
经济	0.778	0.780	0.792	0.791	0.812	0.815
军事	0.803	0.794	0.820	0.823	0.829	0.830
科技	0.776	0.778	0.795	0.798	0.821	0.820
商务	0.791	0.802	0.812	0.816	0.820	0.821
社会	0.772	0.776	0.799	0.801	0.825	0.827
体育	0.812	0.803	0.820	0.823	0.838	0.840
娱乐	0.804	0.803	0.821	0.822	0.837	0.839

新方法各类的查全率和查准率均最优，TBS次之，VSM最差。

同/多义词分布

	同义词个数	多义词个数
教育	251	85
经济	144	126
军事	182	26
科技	236	132
商务	178	31
社会	240	109
体育	119	122
娱乐	239	94

- 上下文依赖性(或多义词)的确影响短文本的相似性度量;
- 军事和商务两类中的多义词个数较少，所以优势不明显。

同/多义词分布

	同义词个数	多义词个数
教育	251	85
经济	144	126
军事	182	26
科技	236	132
商务	178	31
社会	240	109
体育	119	122
娱乐	239	94

- 上下文依赖性(或多义词)的确影响短文本的相似性度量;
- 军事和商务两类中的多义词个数较少，所以优势不明显。

综合比较

	Re	Pr	F_1
VSM	0.790	0.789	0.789
TBS	0.808	0.810	0.809
新方法	0.826	0.828	0.827

从分类性能上看，新方法优于TBS(+2%)，也优于VSM(+4%)。

汇报内容

总结

- 综合解决短文本的两个问题：特征稀疏性和上下文依赖性
- 在文本分类中，所提出的方法性能优于VSM和TBS
- 展望：社交网络下表述的口语化和不规范化

总结

- 综合解决短文本的两个问题：特征稀疏性和上下文依赖性
- 在文本分类中，所提出的方法性能优于VSM和TBS

- 展望：社交网络下表述的口语化和不规范化

总结

- 综合解决短文本的两个问题：特征稀疏性和上下文依赖性
- 在文本分类中，所提出的方法性能优于VSM和TBS

- 展望：社交网络下表述的口语化和不规范化

Thank you !

Q & A?

Email:tjzhifei@163.com
Weibo:@同济志飞