事件发现与跟踪项目

一、项目背景

- 事件排重

在现阶段的微博头条推荐中，只有物料排重模块，没有针对事件的处理模块。当出现重大事件时，推荐流中会出现多篇同一事件的报道文章，这是在事件上的重复。合理的情况是针对同一事件同一阶段的报道最好只出一篇。

- 热点时效性

目前的热点流时效性较差，需要提升。

- 热点流点击率

目前热点流点击率较低，可以通过热点聚合和事件露出文章选择，提升热点流点击率

- 提升推荐整体效果

热点在资讯推荐中有很大作用，全局热点可以帮助人们筛选最近发生的重大事件，而垂直热点则是兴趣推荐中的重要物料。

二、项目分解

通常情况下，热点事件是一定时间、一定范围内，公众最为关心的热点问题。热点事件发现就是从微媒体文本的传播数据流中自动发现热点内容并将与之相关的其他信息联系在一起。热点事件跟踪则是从微媒体文本传播数据中分析热点话题的发

展规律。(引用自微媒体文本热点事件的发现与跟踪)

1、 热点发现

热点发现包括热点事件的检测和聚合。在这部分要解决两个问题：第一，如何很好的检测出热点事件，包括突发事件、热度上升事件；第二，在得到热点事件候选集后怎么把同一事件相关文章聚合到一起。

2、 热点跟踪

一般讨论的是在某个时间区域内热点事件的发现，这只是即时的热点事件。有些热点事件随着时间的推移,环境的改变，不断地发酵。因此对热点事件的跟踪是对近一段时期内（若干个连续的时间区域）的热点事件的趋势分析与预测。

以时序序列为基准的微媒体文本热点事件的传播可以视作一个信息循环模型，分为形成期(事件被曝光)、爆发期(微博、传统媒体、其他新媒体形成舆论合力)、缓解期(政府的介入)、平复期(问题的解决)和消失期(热点消失)五个阶段。任何热点事件都不会无限期传播下去，所以增加一个消失期，才符合热点事件的生命周期。一般来讲，讨论热点事件的演变过程可以反映到某一日期间隔的连续时间区域上。根据热点事件强度的变化趋势，判断热点事件生命周期内的变化。热点事件跟踪就是判定热点话题的走势。

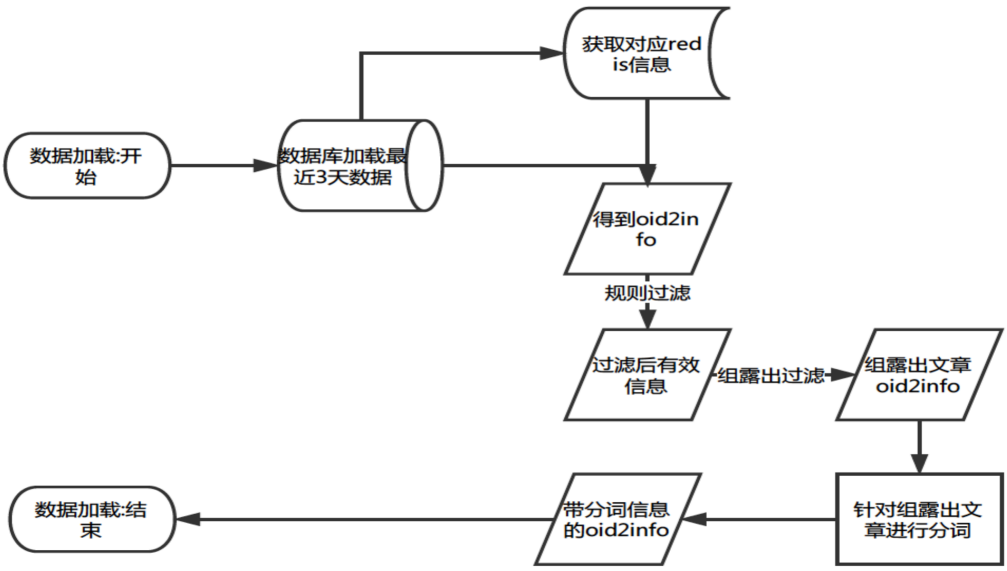
三、项目迭代

1、 baseline 版本

◊ **事件生产：**

➢ 数据处理(分词)

取最近两天的文本数据，对文章标题进行分词，词性过滤，停止词过滤。



➢ 聚类：

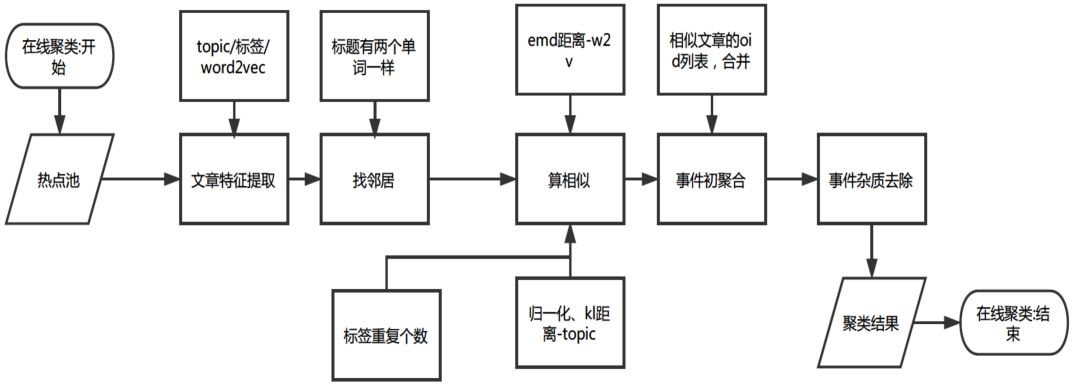
✓ 聚类中心确定：根据 gsize,top 特征值确定聚类中心

✓ 文章特征提取：topic 特征(标签和权重)、二级/三级标签特征(标签和权重)、word2vec 特征

✓ 聚类：

✧ 找邻居：每篇文章要与邻居算相似度,先把邻居找到；要求标题最少有两个单词一致

- ✧ 算相似：
  - 语义相似度：emd 距离
  - 主题相似度：计算 topic 相似度,获取文章在所有 topic 上的概率分布,没有的则权重为 0。归一化 topic 权重向量,然后计算 Kl 距离。
  - 二级/三级标签相似度：标签重复个数
- ✧ 事件初聚合
  - 在得到一篇文章的相似文章列表后,再聚合就是事件
- ✧ 事件聚合杂质去除
- ➢ 事件判断：
  - ✓ 根据规则判断得到的是否为真正的事件
- ➢ 分配 eventid
  - ✓ 一个文章有事件 id 后不再改变
  - ✓ 第一次启动，计算 event，存到 redis。下次计算如果一个新文章符合多个事件的要求，随机选择一个事件，但原来的事件不改变。事件 id 可以随机使用事件文章的 gid



◊ **在线事件抽取：**

➢ 事件露出文章优选

➢ 事件召回流

2、 第二版方案(增加了热点池)

事件发现

➢ 构建候选子集

✓ 热词：突发词检测系统开发、热词榜

✓ 热门特征：gsize,top

➢ 构建热门文章池：热门子集融合，过滤

热点事件聚合

➢ 单遍聚类算法

✓ 文本特征提取

✓ 阈值模型

3、 第三版方案(增加了热点池；seq2seq 模型提特征)

为了更好的计算文章语义相似度，尝试了 seq2seq 模型，该

模型的作用就是提取文章表示向量，用来计算相似度。

4、 第四版方案(利用 simnet 网络完成文本匹配)

◊ **事件生产：**

➢ 数据处理(分词)

取最近两天的文本数据，对文章标题进行分词，词性过滤，停止词过滤。

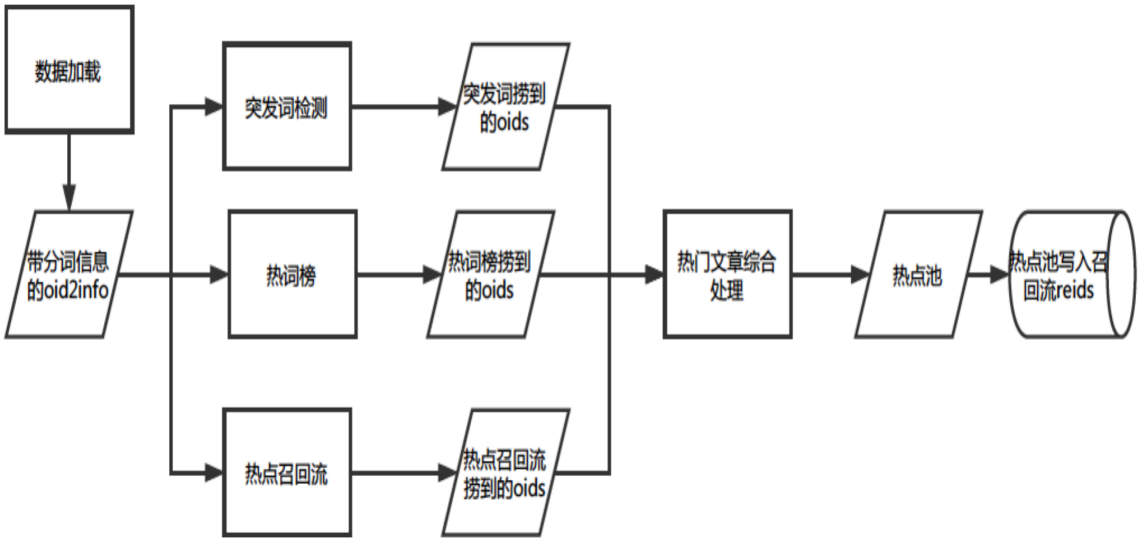
➢ 事件发现(构造热点池)

✧ 构建候选子集

✓ 热词：突发词检测系统开发、热词榜

✓ 热门特征：gsize,top, 热门文章召回流(媒体热门、微博热门、全局热门、各领域垂直热门)

✧ 构建热门文章池：热门子集融合，过滤



准确简单的介绍突发词检测模型：

A、 算法背景

kleinberg 状态机模型，将事件突发表征为状态的改变。

B、 算法输入、输出

输入为可以为不同时间段单词出现总次数，单词出现文档数，时间段划分可以以分钟为单位，小时为单位，天为单位。

算法输出为状态数组，需要根据状态数组来进行判断单词是否为突发词。

C、 模型

两个状态的自动机：

- Two states automaton A: q_0, q_1

$$f_0(x) = \alpha_0 e^{-\alpha_0 x} \quad f_1(x) = \alpha_1 e^{-\alpha_1 x}$$

- Based on a set of messages to estimate a state sequence
  - Maximum likelihood

- *n* inter-arrival gaps: $\mathbf{x} = (x_1, x_2, \mathrm{L}\ \ x_n)$
- A state sequence: $\mathbf{q} = (q_{i_1}, q_{i_2}, \mathrm{L}\ \ q_{i_n})$
- b denotes the number of state transitions in the sequence **q**

$$\Pr[\mathbf{q} \mid \mathbf{x}] = \frac{\Pr[\mathbf{q}] f_{\mathbf{q}}(\mathbf{x})}{\sum_{\mathbf{q}'} \Pr[\mathbf{q}'] f_{\mathbf{q}'}(\mathbf{x})}$$

$$= \frac{1}{Z} \left(\frac{p}{1-p} \right)^{b} (1-p)^{n} \prod_{t=1}^{n} f_{i_t}(x_t)$$

- Finding a state sequence q maximizing previous probability is equivalent to finding one that minimizes

$$-\ln \Pr[\mathbf{q} \mid \mathbf{x}] = b \ln \left(\frac{p}{1-p} \right) + \left(\sum_{t=1}^{n} -\ln f_{i_t}(x_t) \right) - n \ln(1-p) + \ln Z$$

- Equivalent to minimize the following *cost function*:

$$c(\mathbf{q} \mid \mathbf{x}) = b \ln \left(\frac{p}{1-p} \right) + \left(\sum_{t=1}^{n} -\ln f_{i_t}(x_t) \right)$$

D、 参数估计

em 算法

E、 推理

维特比算法

- 事件聚合
  - 找邻居：每篇文章要与邻居算相似度,先把邻居找到；要求标题最少有一个单词一致；得到文章和邻居组合 pair-list。评估下预测数据规模。
  - 算相似：

语义相似度：针对文章和邻居组合 pair-list,利用 simnet 判断是否相似。

  - 事件初聚合

在得到一篇文章的相似文章列表后,再聚合就是事件

  - 事件聚合杂质去除

找到事件组每篇文章都出现的单词或者出现次数前 3 的单词，如果文章没有不满足单词限制，就从组内删掉。

- 事件判断：

根据规则判断得到的是否为真正的事件，主要判断事件组内 gsize,top 是否满足限制

- 分配 eventid
  - 一个文章有事件 id 后不再改变
  - 第一次启动，计算 event，存到 redis。下次计算如果一个

新文章符合多个事件的要求，随机选择一个事件，但原来的事件不改变。事件 id 可以随机使用事件文章的 gid

◊ **事件消费：**

- 事件露出文章优选

  规则选取，综合文章特征/热门特征/用户反馈特征/时间衰减，去除事件重复

- 事件召回流

  热点池文章存入召回流 redis;

  召回有事件 id 的文章，按照事件组大小排序。