# 基于Amazon Comprehend的文本分析开发实践

奚文俊，AWS 技术客户经理
Wenjun Xi, Technical Account Manager, Amazon Web Services

2018年5月15日
May 15, 2018

# 自然语言处理的趋势

# 自然语言处理的研究方向

# 自然语言处理场景无处不在

- 公众公开的内容
  - 社交媒体
  - 新闻

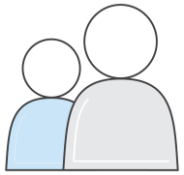
- 客户Engagement
  - 产品评论
  - 产品支持（电话、电子邮件、反馈）

# 自然语言处理模型训练的挑战

# Amazon Comprehend

# Amazon Comprehend: 自然语言处理

情感



实体

语言

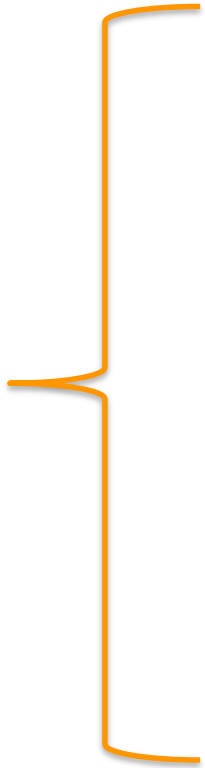
关键短语

主题建模

基于深度学习

# 文本分析

Amazon.com, Inc. is located in Seattle, WA and was founded July 5th, 1994 by Jeff Bezos. Our customers *love buying everything* from books to blenders at great prices



**Named Entities**

- Amazon.com: Organization
- Seattle, WA : Location
- July 5th,1994: Date
- Jeff Bezos : Person

**Keyphrases**

- Our customers
- books
- blenders
- great prices

**Sentiment**

- *Positive*

**Language**

- English

# 主题分类

主题关键词

| Topic | Term | Weight |
|---|---|---|
| 0 | Washington | .89 |
| 1 | Silicon Valley | .67 |
| 2 | Roasting | .91 |

文档按主题归类

| Document | Topic | Proportion |
|---|---|---|
| Doc.txt | 0 | .89 |
| Doc.txt | 1 | .07 |
| Doc.txt | 2 | .04 |

# 常见使用场景

- 客户反馈分析
  - 实时分析客户对于贵公司品牌、产品和服务的情绪

- 基于情感的搜索
  - 让您的搜索功能更智能：基于关键短语、情感和主题

- 知识发现和管理
  - 根据主题来管理文本 / 文档，个性化内容推荐

# 如何使用Amazon Comprehend

# API概览

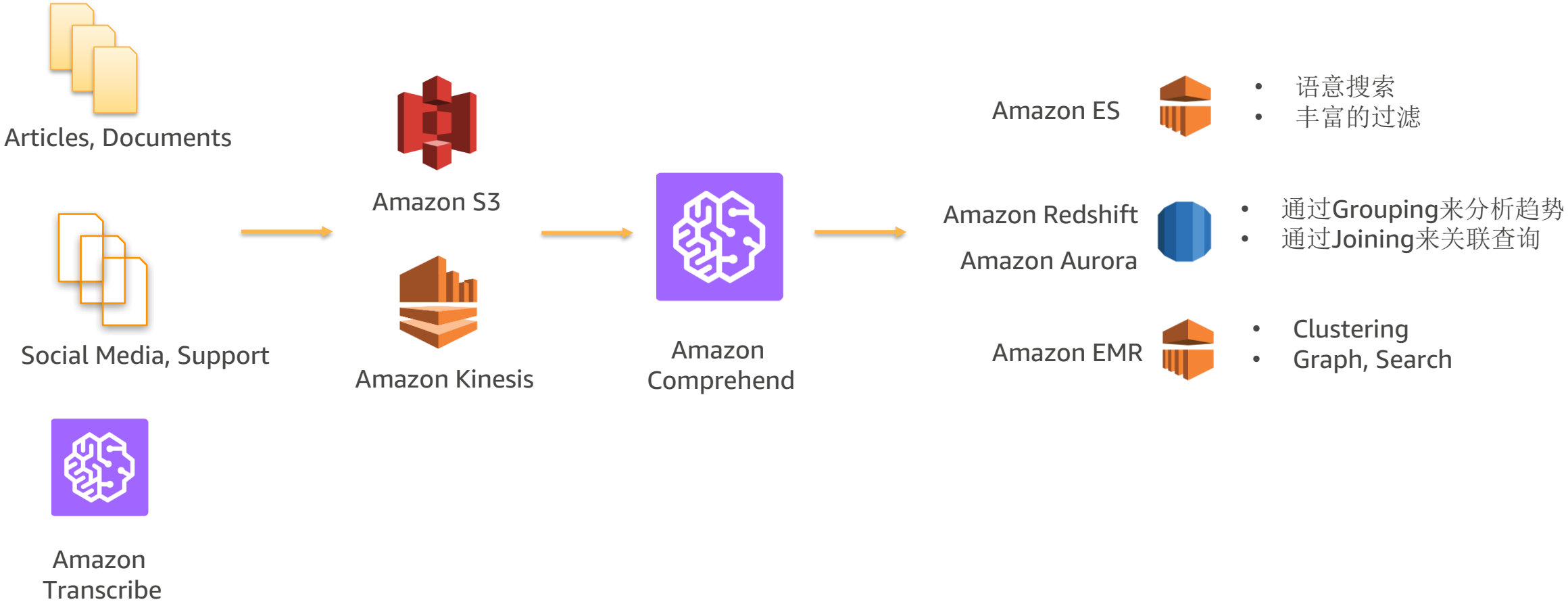
## Synchronous

| | |
|---|---|
| DetectDominantLanguage<br>BatchDetectDominantLanguage | 检测语言，可以检测100种语言 |
| DetectEntities<br>Batch DetectEntities | 检测命名实体，比如人名、地点、组织等 |
| DetectKeyPhrases<br>Batch DetectKeyPhrases | 检测能表征文本内容特点的关键名词短语 |
| DetectSentiment<br>Batch DetectSentiment | 检测文本所表示的情感：正面、负面、混合（正面负面皆有）、中性 |

## Asynchronous

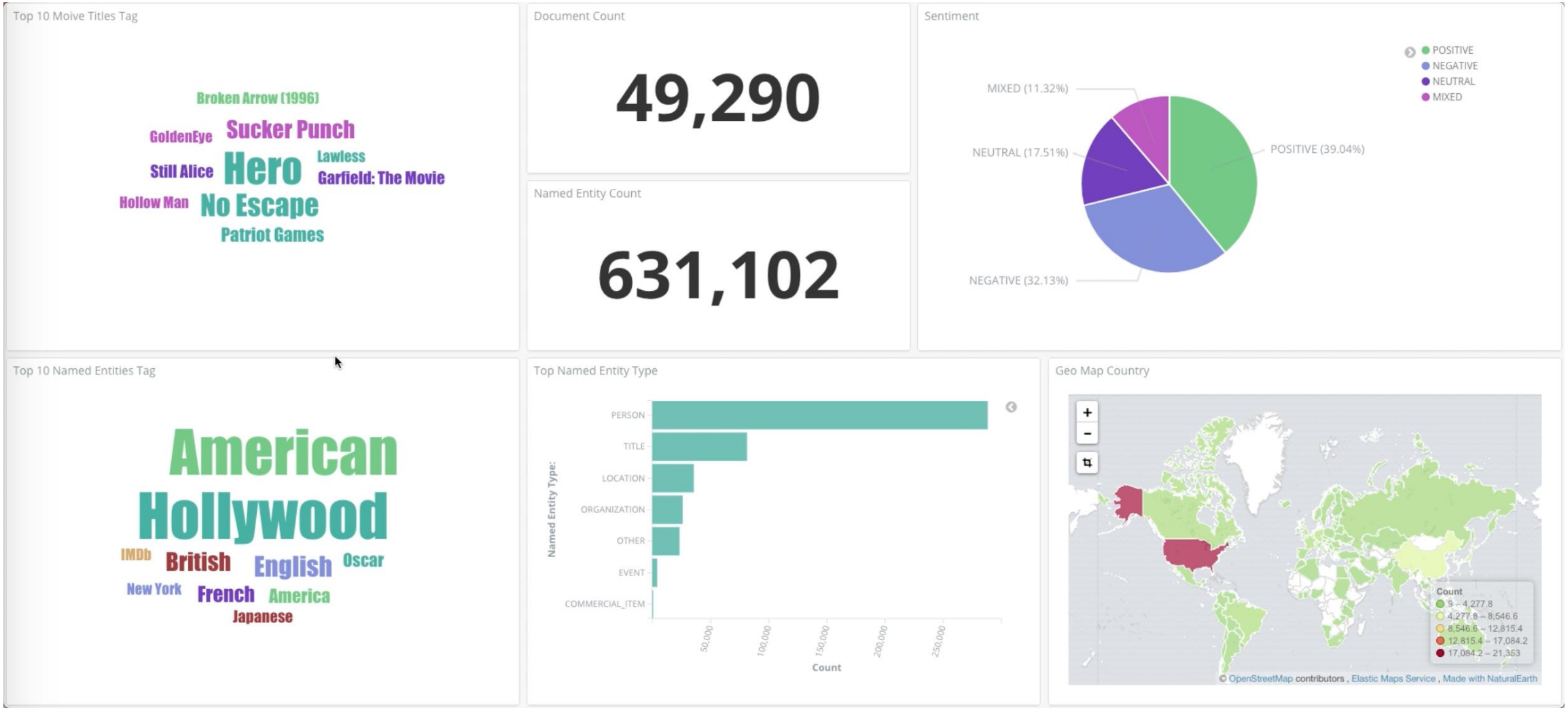
| | |
|---|---|
| StartTopicDetection | 开始主题分类任务 |
| ListTopicDetection | 列举所有您提交的主题分类任务 |
| DescribeTopicDetection | 获取主题分类任务的状态等信息 |

# Comprehend和AWS服务结合

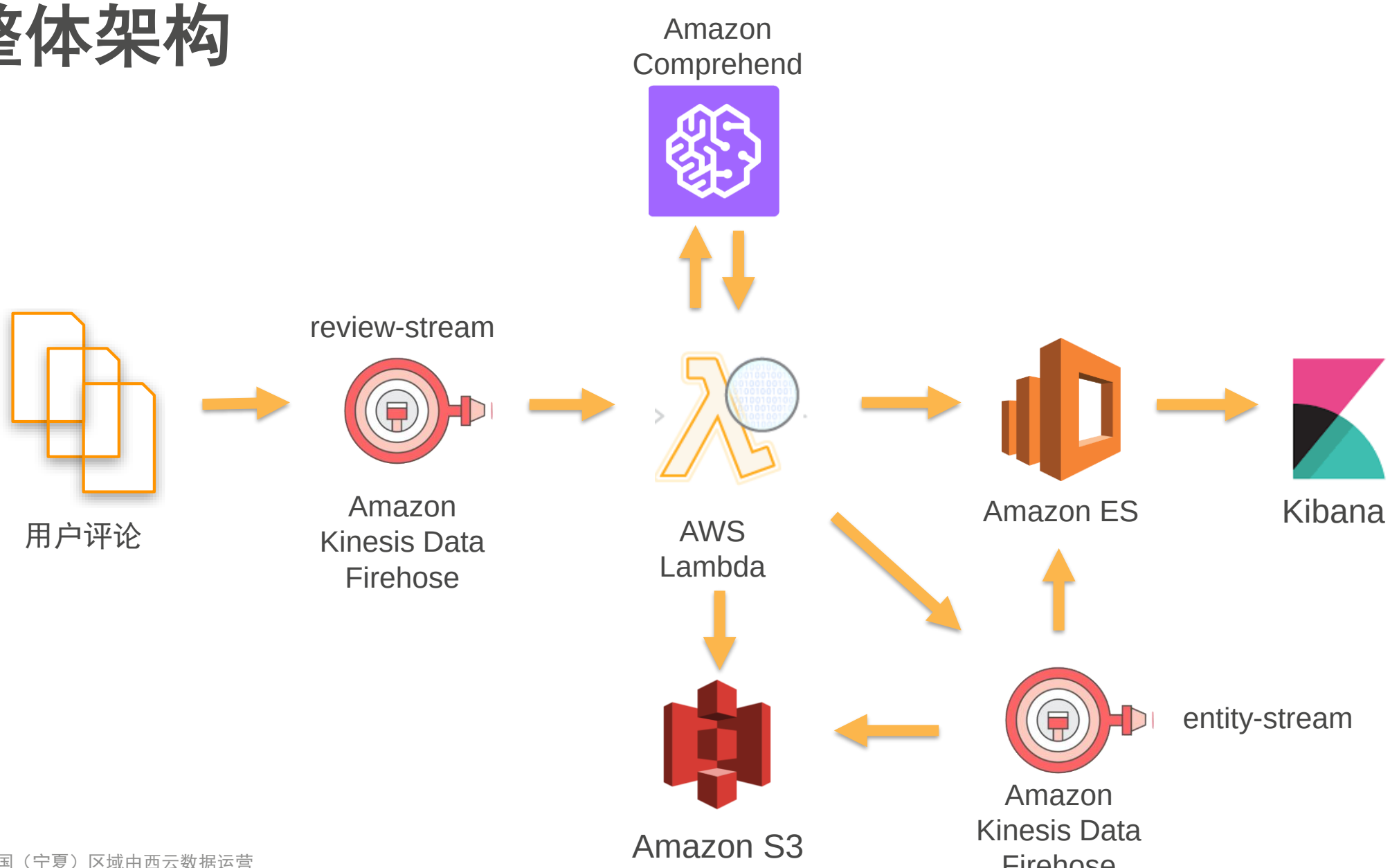


# 演示：近实时电影评论分析仪表盘

# 仪表盘概览

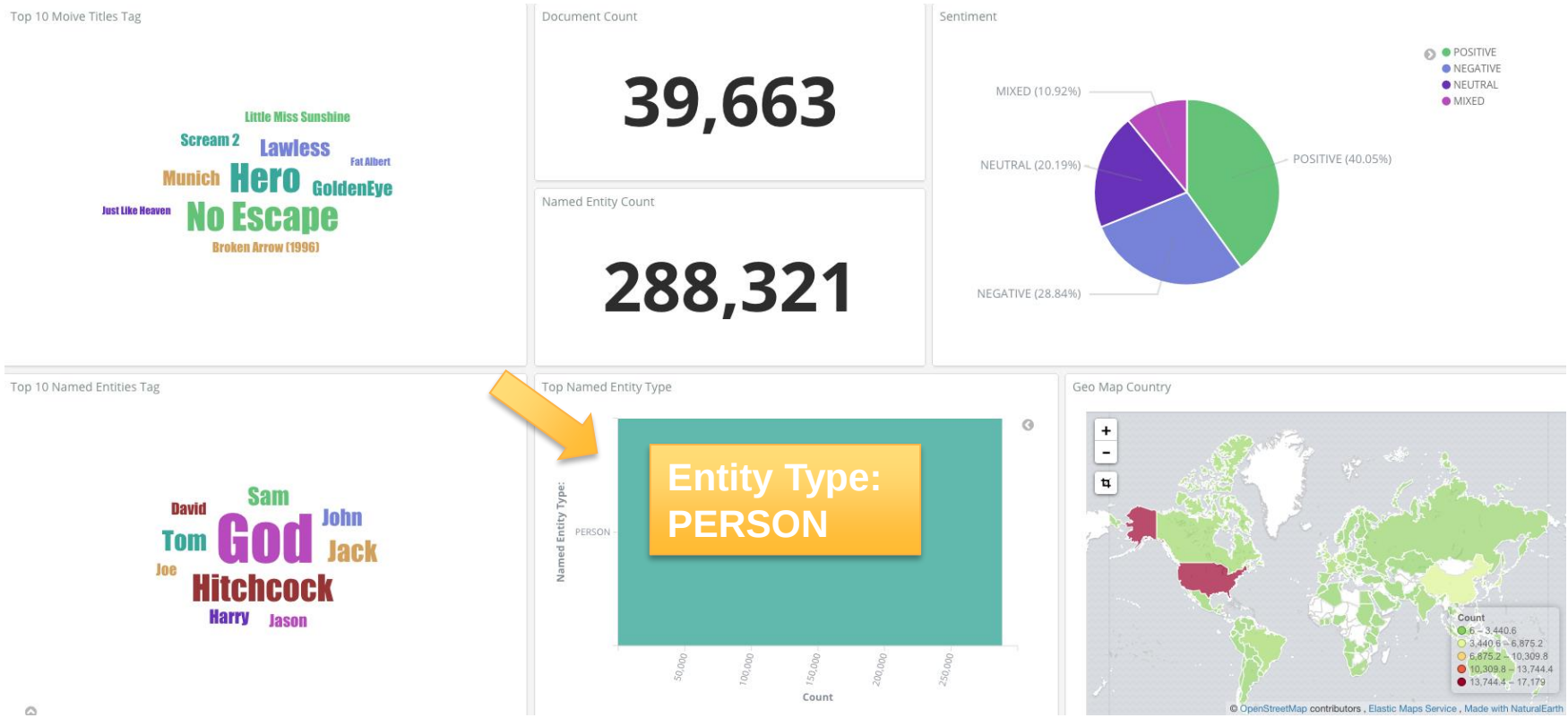

Top 10 Moive Titles Tag
Broken Arrow (1996)
GoldenEye
Sucker Punch
Still Alice
Hero
Lawless
Garfield: The Movie
Hollow Man
No Escape
Patriot Games
Document Count
49,290
Named Entity Count
631,102
Sentiment
POSITIVE
NEGATIVE
NEUTRAL
MIXED
MIXED (11.32%)
NEUTRAL (17.51%)
POSITIVE (39.04%)
NEGATIVE (32.13%)
Top 10 Named Entities Tag
American
Hollywood
IMDb
British
English
Oscar
New York
French
America
Japanese
Top Named Entity Type
PERSON
TITLE
LOCATION
ORGANIZATION
OTHER
EVENT
COMMERCIAL_ITEM
Named Entity Type:
50,000
100,000
150,000
200,000
250,000
Count
Geo Map Country
Count
9 – 4,277.8
4,277.8 – 8,546.6
8,546.6 – 12,815.4
12,815.4 – 17,084.2
17,084.2 – 21,353
© OpenStreetMap contributors , Elastic Maps Service , Made with NaturalEarth

# 整体架构


Amazon
Comprehend
review-stream
用户评论
Amazon
Kinesis Data
Firehose
AWS
Lambda
Amazon ES
Kibana
Amazon S3
entity-stream
Amazon
Kinesis Data
Firehose

# 设计要点

- 为什么用到两个Kinesis Firehose?
  - 需求：点击Dashboard中的一个视图，其他视图相应作过滤



# 设计要点

- 设计一：使用nested类型

```
{
  "properties": {
    "doc_type": {"type": "keyword"},
    "parent": {"type": "keyword"},
    "text": {"type": "text"},
    "title": {"type": "keyword"},
    "entity": {
        "type": "nested"
    },
    "key_phrase": {"type": "text"},
    "sentiment": {"type": "keyword"},
    "ip_addr": {"type": "ip"},
    "geoip": {
```


Kibana目前不支持复合字段的聚合查询

https://www.elastic.co/guide/en/kibana/current/nested-objects.html

# 设计要点

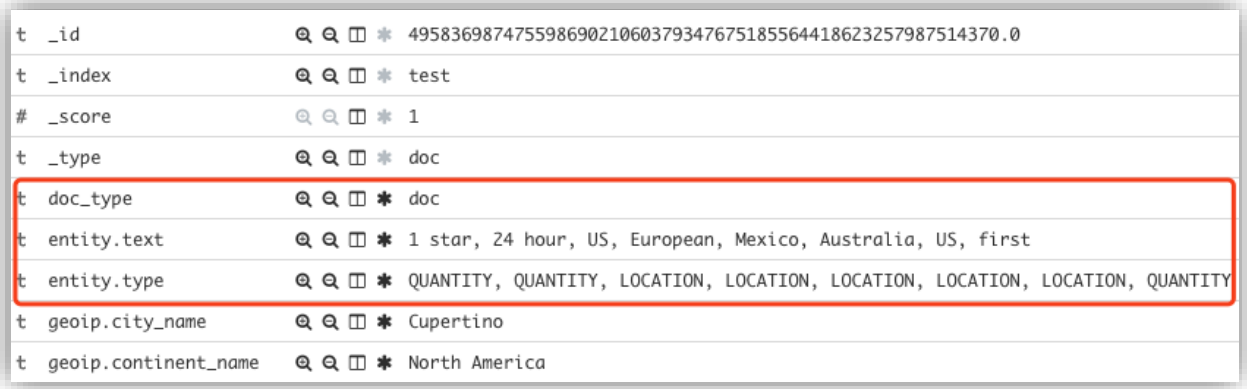
- 设计二： 
  - 命名实体文本和类型作为两个List

| | | |
|---|---|---|
| t | entity.text | Jack Black, Owen Wilson, Ben Stiller, Doctor Who, Metallica |
| t | entity.type | PERSON, PERSON, PERSON, TITLE, ORGANIZATION |

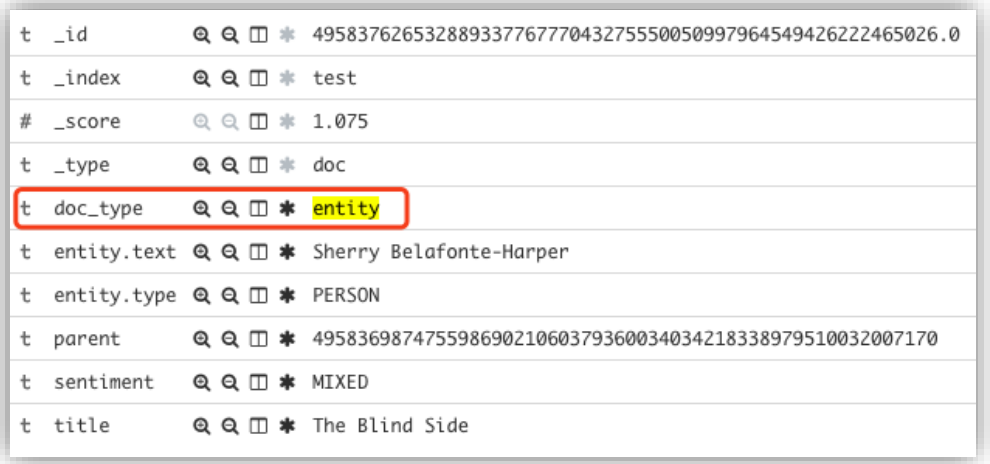
  - 缺陷：命名实体文本和类型之间的关联丢失
    无法通过类型来过滤命名实体

# 设计要点

- 设计三：
  - 增加一个字段doc_type用来区分文档类型:
    - doc：该文档包含entity name和type两个List
    - entity：每个entity作为一个独立文档



| | | |
|---|---|---|
| t | _id | 4958369874755986902106037934767518556441862325798751437 0.0 |
| t | _index | test |
| # | _score | 1 |
| t | _type | doc |
| t | doc_type | doc |
| t | entity.text | 1 star, 24 hour, US, European, Mexico, Australia, US, first |
| t | entity.type | QUANTITY, QUANTITY, LOCATION, LOCATION, LOCATION, LOCATION, LOCATION, QUANTITY |
| t | geoip.city_name | Cupertino |
| t | geoip.continent_name | North America |



| | | |
|---|---|---|
| t | _id | 4958376265328893377677704327555005099796454942622246 5026.0 |
| t | _index | test |
| # | _score | 1.075 |
| t | _type | doc |
| t | doc_type | entity |
| t | entity.text | Sherry Belafonte-Harper |
| t | entity.type | PERSON |
| t | parent | 4958369874755986902106037936003403421833897951003200 7170 |
| t | sentiment | MIXED |
| t | title | The Blind Side |

# 设计要点

- 使用Comprehend批量API提升吞吐量，一次分析25条文本

```python
# batch detect
if (i % BATCH_SIZE == 0 or i >= num_record):
    print('processing batch #{}'.format(i / BATCH_SIZE))

    # detect key phrases
    key_phrase_list = get_key_phrase_list(text_list)

    # detect sentiments
    sentiment_list = get_sentiment_list(text_list)

    # detect named entities and send to another firehose
    entity_list = send_entity(text_list, sentiment_list, recordId_list, title_list)
```

```python
def get_key_phrase_list(text_list):
    response = client_comprehend.batch_detect_key_phrases(TextList=text_list, LanguageCode='en')
    key_phrase_list = []

    for result in response['ResultList']:
        key_phrases = list(set(x['Text'] for x in result['KeyPhrases']))
        key_phrase_list.append(key_phrases)
    return key_phrase_list

def get_sentiment_list(text_list):
    response = client_comprehend.batch_detect_sentiment(TextList=text_list, LanguageCode='en')
    sentiment_list = []

    for result in response['ResultList']:
        sentiment = result['Sentiment']
        sentiment_list.append(sentiment)
    return sentiment_list
```

```
{
   "ErrorList": [
      {
         "ErrorCode": "string",
         "ErrorMessage": "string",
         "Index": number
      }
   ],
   "ResultList": [
      {
         "Index": number,
         "KeyPhrases": [
            {
               "BeginOffset": number,
               "EndOffset": number,
               "Score": number,
               "Text": "string"
            }
         ]
      }
   ]
}
```

```
{
   "ErrorList": [
      {
         "ErrorCode": "string",
         "ErrorMessage": "string",
         "Index": number
      }
   ],
   "ResultList": [
      {
         "Index": number,
         "Sentiment": "string",
         "SentimentScore": {
            "Mixed": number,
            "Negative": number,
            "Neutral": number,
            "Positive": number
         }
      }
   ]
}
```

# 动手实践

## Blog

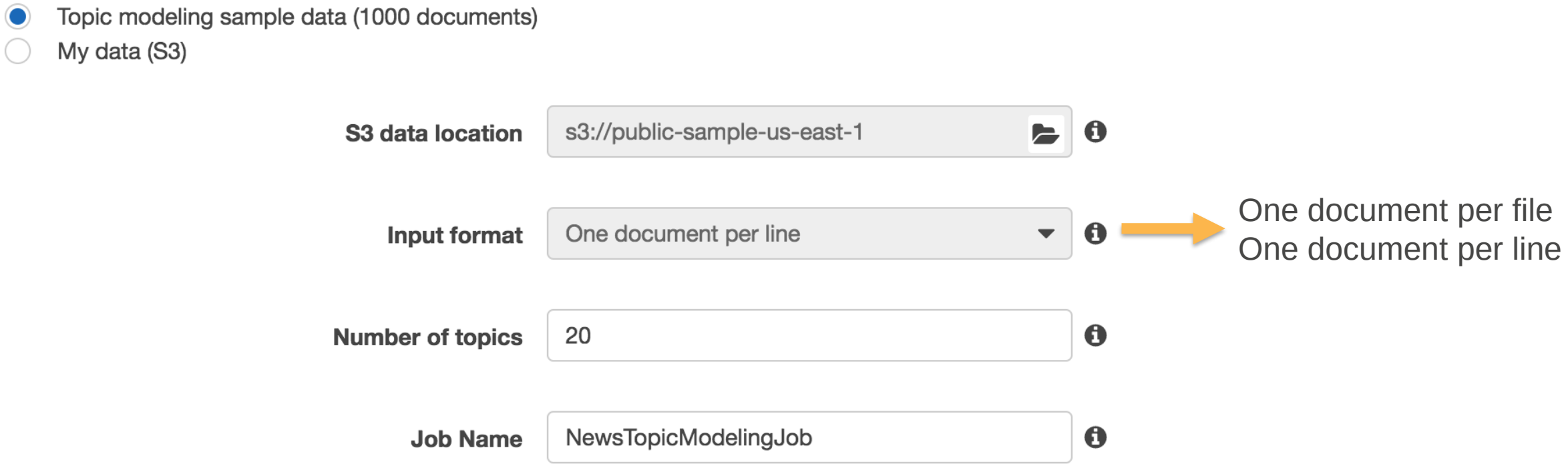
https://aws-blogs-prod.amazon.com/china/realizing-near-real-time-text-sentiment-analysis-with-aws-comprehend/

## 相关资源

- 代码：https://github.com/gddezero/realtime-text-analysis
- 数据集：http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz

# 演示：文本主题分类

# 输入数据


Topic modeling sample data (1000 documents)
My data (S3)
S3 data location
s3://public-sample-us-east-1
Input format
One document per line
Number of topics
20
Job Name
NewsTopicModelingJob
One document per file
One document per line

# 输出结果和IAM role

## Select output location

Select the preferred output format for your analysis. S3 data output location, and the format of data as CSV.

**S3 data location** s3://xwj-files-use1/News/

## Select an IAM role

The topic modeling job will use the IAM role to access your Amazon S3 input and output buckets.

- ○ Select an existing IAM role
- ◉ Create a new IAM role

**Permissions to access** input and output S3 buckets

Your role will have access to resources in buckets: "xwj-files-use1"

**Name suffix** xwj

# 查看任务运行状态和结果

| Job name | Job ID | Start time | End time | Status |
|---|---|---|---|---|
| NewsTopicModelingJob | 31417b3e6150ae7d37b7b4ac05c7329c | 4/26/2018, 4:03:08 PM | 4/26/2018, 4:09:14 PM | Completed |

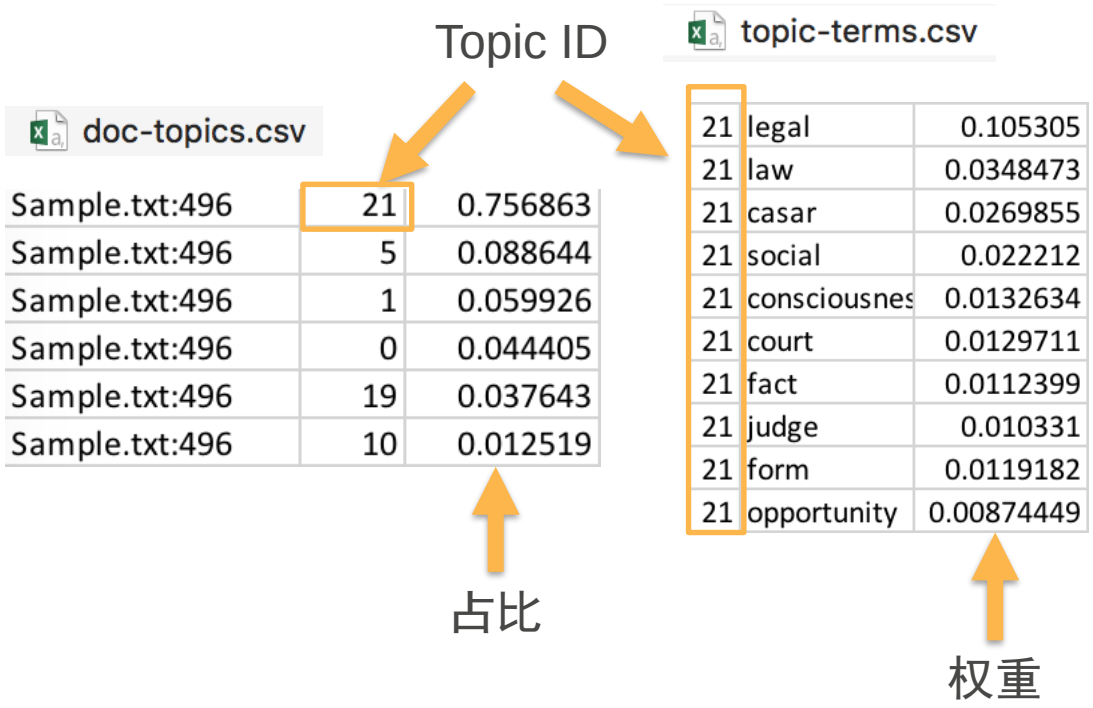
| Name | Last modified | Size |
|---|---|---|
| output.tar.gz | Apr 26, 2018 4:08:43 PM GMT+0800 | 16.9 KB |

- doc-topics.csv
- topic-terms.csv

# 查看任务运行状态和结果

| Job name | Job ID | Start time | End time | Status |
|---|---|---|---|---|
| NewsTopicModelingJob | 31417b3e6150ae7d37b7b4ac05c7329c | 4/26/2018, 4:03:08 PM | 4/26/2018, 4:09:14 PM | Completed |

Legal consciousness is a collection of understood and/or imagined to have understood, legal awareness of ideas, views, feelings and traditions imbibed through legal socialization; which reflects as legal culture among given individual, or a group, or a given society at large. The legal consciousness evaluates the existing law and also bears in mind an image of the desired or ideal law.Consciousness is not an individual trait nor solely ideational; legal consciousness is a type of social practice reflecting and forming social structures. The study of legal Consciousness documents the forms of participation and interpretation through which act or sustain, reproduce, or amend the circulating contested or hegemonic structures of meanings concerning law. Legal consciousness is the way in which law is experienced and interpreted by specific individuals as they engage, avoid, resist or just assume the law and legal meanings.Legal consciousness is a state of being, legal socialisation is the process to Legal consciousness; where as legal awareness & legal mobilisation are means to achieve the same.



# 免费试用

- 现已在如下区域可用
  - US East (N. Virginia)
  - US East (Ohio)
  - US West (Oregon)
  - EU (Ireland)
- 免费试用额度
  - 命名实体识别、情绪分析、关键短语抽取、语言检测
    - 每月50K 文本单位 (5M 字母)
  - 主题模型
    - 每月5个任务（每个任务最多1MB）

# Thank You!