Online Mental Disorder Analysis

Improving feature engineering and analysis with Elasticsearch and Kibana

About me

Elvis Saravia

Belize & Taiwan

ML & NLP / Lecturer / Blogger

7+ years: Financial & Graph Data Analysis

What role does ML play in Search ?

What role does Search play in ML ?

Journey

- Overview
- Introduce the Data
- Index Mapping
- Data Preprocessing & Ingest Pipeline
- Custom Analyzers
- Demo (Querying and Visualizations)

Motivation

- Mental disorders impair ability to conduct daily functions
- Leverage *search* and *analytics* to *extract* and *explore* hidden and complex linguistic behaviour from natural language data (e.g., slang, emoticon, stopwords, misspelling, etc.)
- Use *insights* to improve machine learning systems that power chatbots (e.g., monitor and alleviate mood)

time for our first daily checkin! 🙌

Let's do it!

What are you doing now (besides talking to lil ol me)? ✏️

I'm at work

So tell me, how are you feeling today? ✏️

I'm feeling a bit overwhelmed today

Woebot.io (mood tracker)

Overview

Goals:

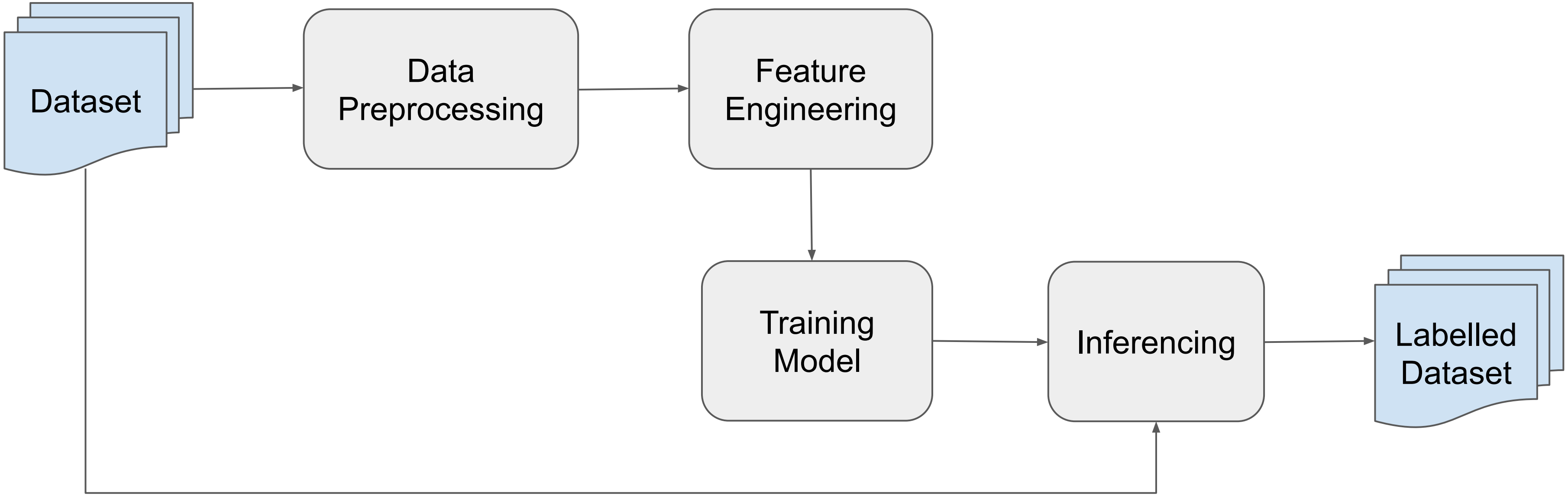
- To broadly demonstrate how to leverage Elasticsearch’s *ingest pipeline* and *custom analyzers* for *preprocessing* and *feature engineering*
- To introduce *common best practices* for dealing with natural language data
- To discover *insights* that assist to improve feature engineering and ML models

Target Audience: Data Scientists / Data Engineers

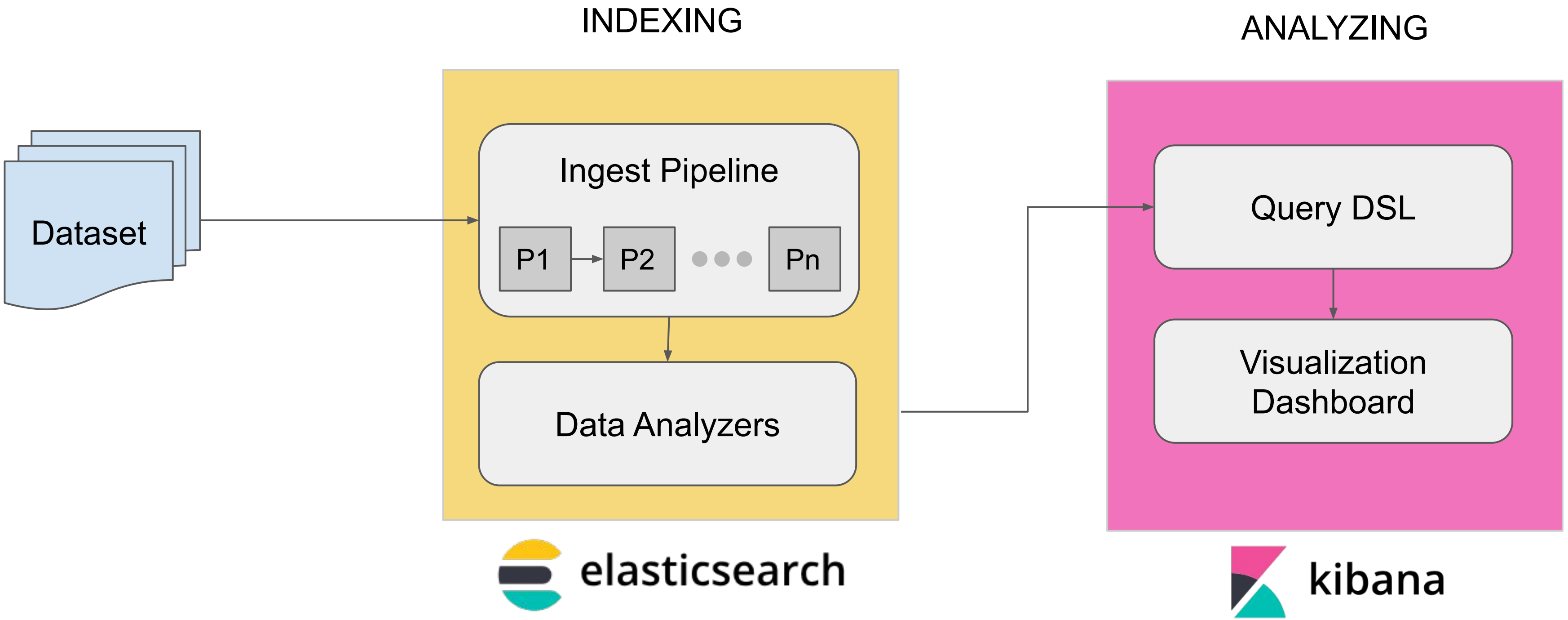
Prerequisites: Assumes basic knowledge of Elasticsearch, Kibana, and Python

Duration: 30 minutes (15 minute demo included)

Scenario - Typical Machine Learning Pipeline

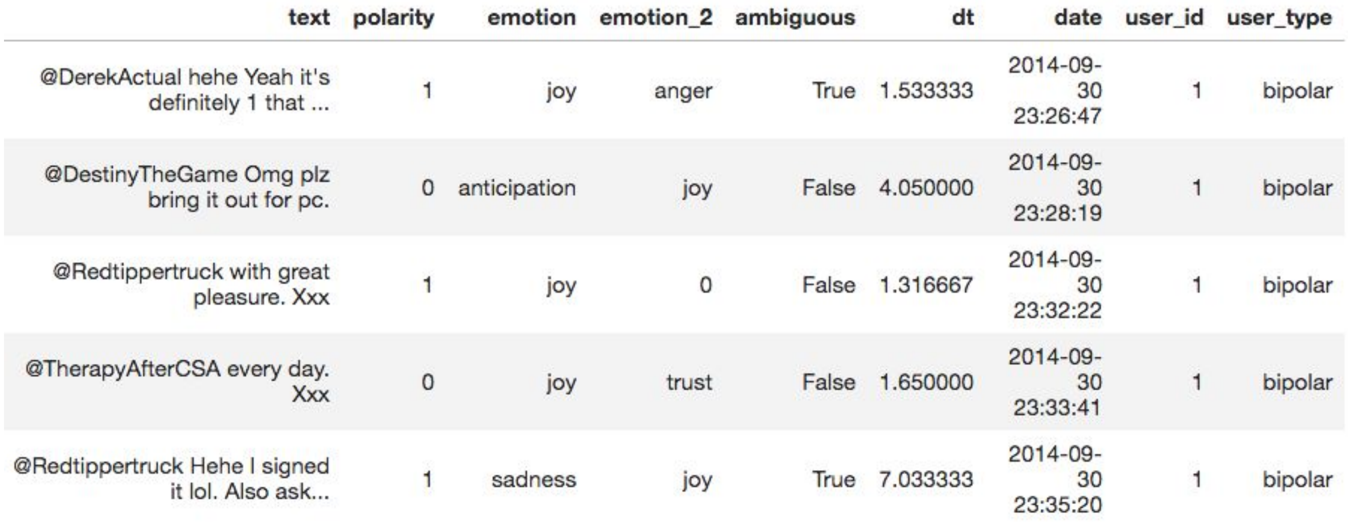


Framework


INDEXING
ANALYZING
Dataset
Ingest Pipeline
P1
P2
Pn
Data Analyzers
Query DSL
Visualization Dashboard
elasticsearch
kibana

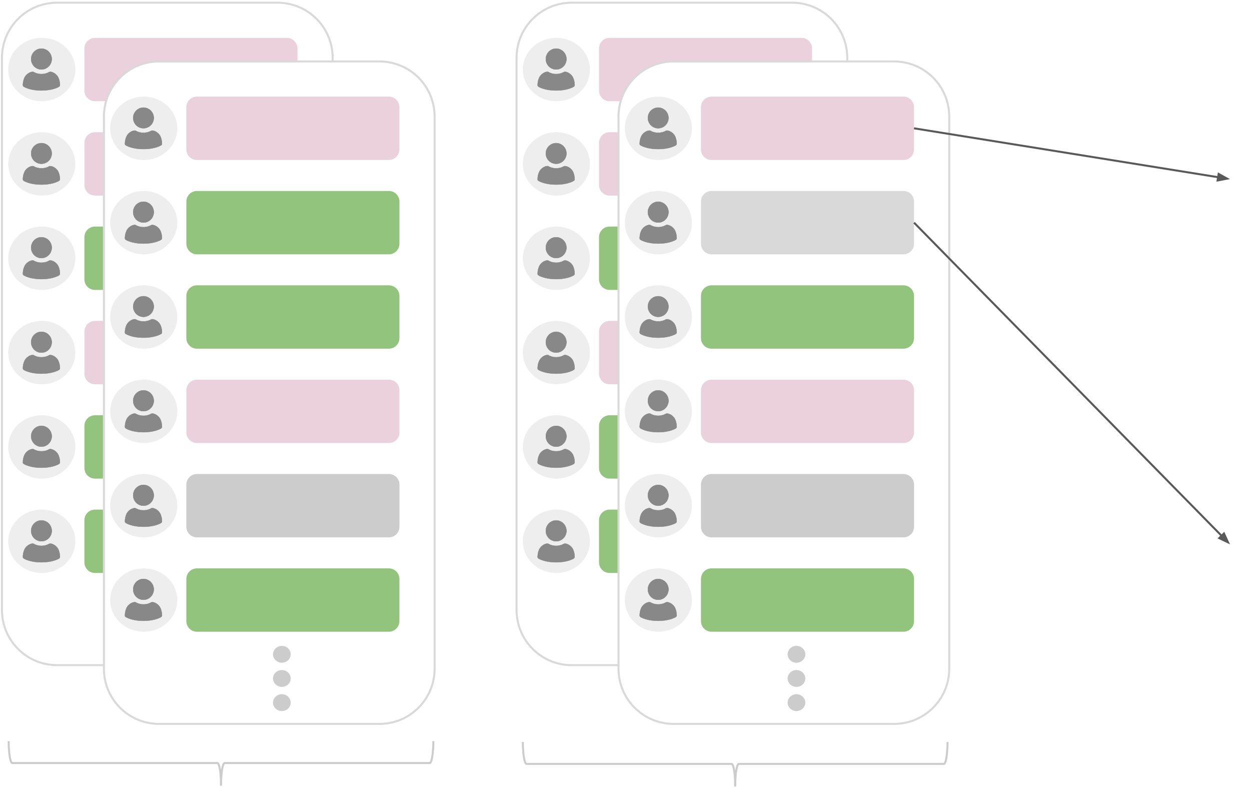
Data Collection

- Online self-reported, mental disorder cases (via “I am diagnosed with X”) :
  - Bipolar disorder: *periods of depression and abnormally elevated mood* (278)
  - Border personality disorder: *longstanding mood swings* (203)
- Normal user profiles (548)



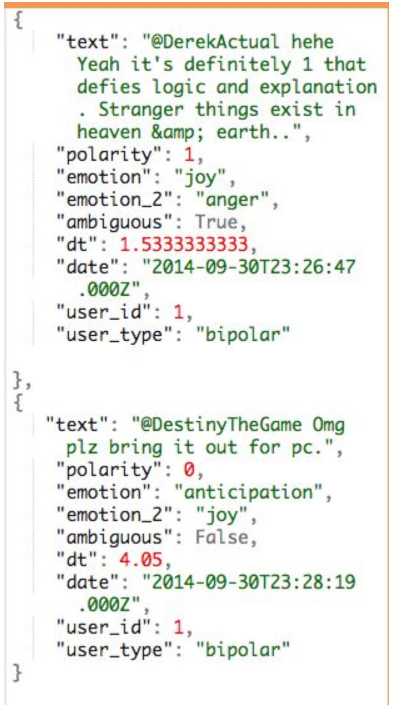
text	polarity	emotion	emotion_2	ambiguous	dt	date	user_id	user_type
@DerekActual hehe Yeah it's definitely 1 that ...	1	joy	anger	True	1.533333	2014-09-30 23:26:47	1	bipolar
@DestinyTheGame Omg plz bring it out for pc.	0	anticipation	joy	False	4.050000	2014-09-30 23:28:19	1	bipolar
@Redtippertruck with great pleasure. Xxx	1	joy	0	False	1.316667	2014-09-30 23:32:22	1	bipolar
@TherapyAfterCSA every day. Xxx	0	joy	trust	False	1.650000	2014-09-30 23:33:41	1	bipolar
@Redtippertruck Hehe I signed it lol. Also ask...	1	sadness	joy	True	7.033333	2014-09-30 23:35:20	1	bipolar

Data - User Timeline



Control group

Diagnosed group

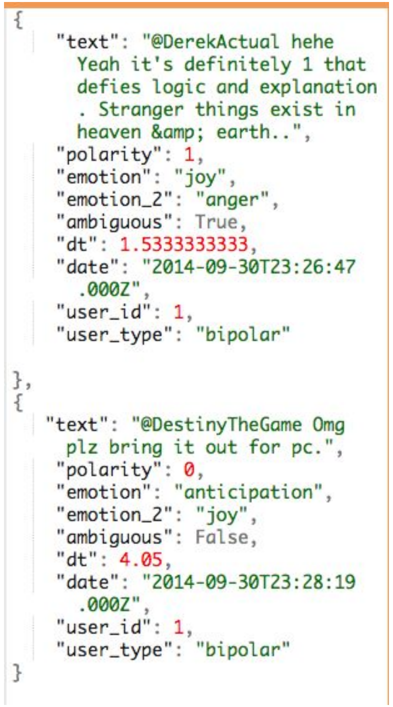


```
{
    "text": "@DerekActual hehe
      Yeah it's definitely 1 that
      defies logic and explanation
      . Stranger things exist in
      heaven &amp; earth..",
    "polarity": 1,
    "emotion": "joy",
    "emotion_2": "anger",
    "ambiguous": True,
    "dt": 1.5333333333,
    "date": "2014-09-30T23:26:47
      .000Z",
    "user_id": 1,
    "user_type": "bipolar"

},
{
   "text": "@DestinyTheGame Omg
     plz bring it out for pc.",
    "polarity": 0,
    "emotion": "anticipation",
    "emotion_2": "joy",
    "ambiguous": False,
    "dt": 4.05,
    "date": "2014-09-30T23:28:19
      .000Z",
    "user_id": 1,
    "user_type": "bipolar"
}
```

Storing Data in Elasticsearch

Considerations before indexing data:

- How to transfer & index data?
  - Logstash / API client (python)
- What scheme or mapping should the data follow?
  - Fields, types, index mapping, preprocessing, etc.
- Any data transformations?
  - Ingest, Analyzers, etc.

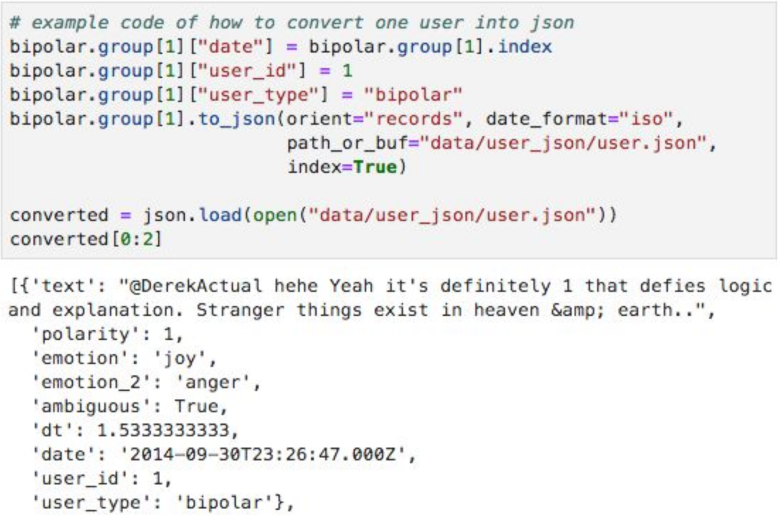


```
{
    "text": "@DerekActual hehe
      Yeah it's definitely 1 that
      defies logic and explanation
      . Stranger things exist in
      heaven &amp; earth..",
    "polarity": 1,
    "emotion": "joy",
    "emotion_2": "anger",
    "ambiguous": True,
    "dt": 1.5333333333,
    "date": "2014-09-30T23:26:47
      .000Z",
    "user_id": 1,
    "user_type": "bipolar"

},
{
   "text": "@DestinyTheGame Omg
     plz bring it out for pc.",
    "polarity": 0,
    "emotion": "anticipation",
    "emotion_2": "joy",
    "ambiguous": False,
    "dt": 4.05,
    "date": "2014-09-30T23:28:19
      .000Z",
    "user_id": 1,
    "user_type": "bipolar"
}
```

Indexing

How to transfer index data?

- API client (Python library)
- Data is available in dataframe format
- Convert data to JSON
- Bulk insert data with Python library
  - Fast / Efficient
  - Flexibility in fields to include
  - Perform any transformations
  - (link to notebook)



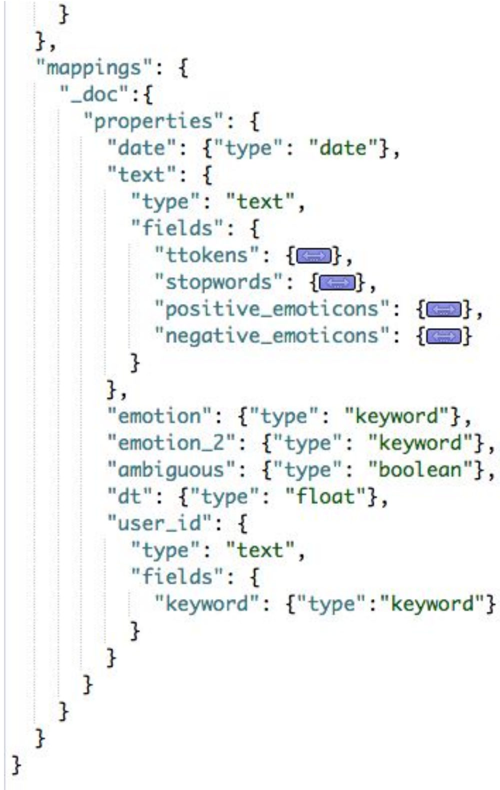
```
# example code of how to convert one user into json
bipolar.group[1]["date"] = bipolar.group[1].index
bipolar.group[1]["user_id"] = 1
bipolar.group[1]["user_type"] = "bipolar"
bipolar.group[1].to_json(orient="records", date_format="iso",
                         path_or_buf="data/user_json/user.json",
                         index=True)

converted = json.load(open("data/user_json/user.json"))
converted[0:2]
```

```
[{'text': "@DerekActual hehe Yeah it's definitely 1 that defies logic
and explanation. Stranger things exist in heaven &amp; earth..",
  'polarity': 1,
  'emotion': 'joy',
  'emotion_2': 'anger',
  'ambiguous': True,
  'dt': 1.5333333333,
  'date': '2014-09-30T23:26:47.000Z',
  'user_id': 1,
  'user_type': 'bipolar'},
```

Index Mapping

Index mapping provides a way of formatting or schematizing data:

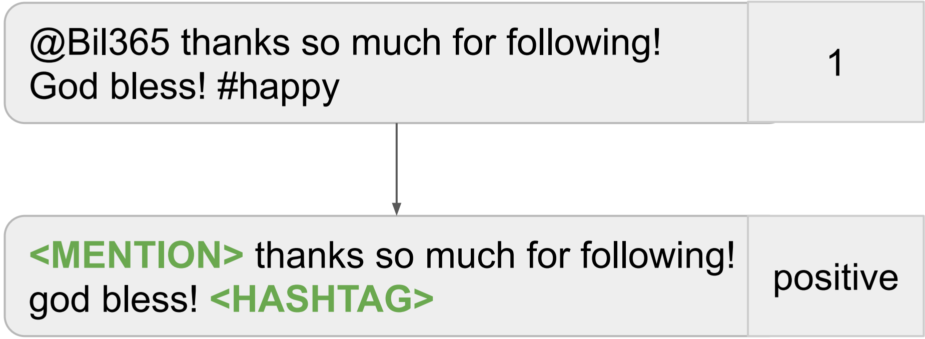
- Configure default pipeline of processors
- Declare field types
- Configure custom analyzers
- …



```
    }
  },
  "mappings": {
    "_doc":{
      "properties": {
        "date": {"type": "date"},
        "text": {
          "type": "text",
          "fields": {
            "ttokens": {↔},
            "stopwords": {↔},
            "positive_emoticons": {↔},
            "negative_emoticons": {↔}
          }
        },
        "emotion": {"type": "keyword"},
        "emotion_2": {"type": "keyword"},
        "ambiguous": {"type": "boolean"},
        "dt": {"type": "float"},
        "user_id": {
          "type": "text",
          "fields": {
            "keyword": {"type":"keyword"}
          }
        }
      }
    }
  }
}
```

Ingest Pipeline

- Provides a mechanism to preprocess data before indexing it
- An ingest pipeline is made of **processors:**
    - Convert labels with ‘set’
    - Lowercase with ‘lowercase’
    - Extracts structured field with regex using ‘grok’
    - Replace text with regex using ‘gsub’



@Bil365 thanks so much for following! God bless! #happy

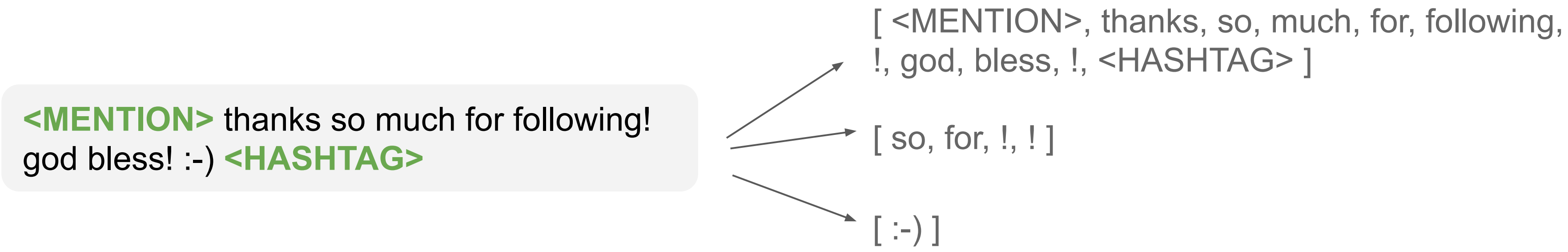
1

<MENTION> thanks so much for following! god bless! <HASHTAG>

positive

Analyzers

- Analyzers provide a way to improve search and conduct special analyses on data
- We will use analyzers to **discover linguistic phenomena:**
  - Twitter special tokenizer
  - Extract stopwords from predefined list
  - Obtain positive and negative emoticons



<MENTION> thanks so much for following! god bless! :-) <HASHTAG>

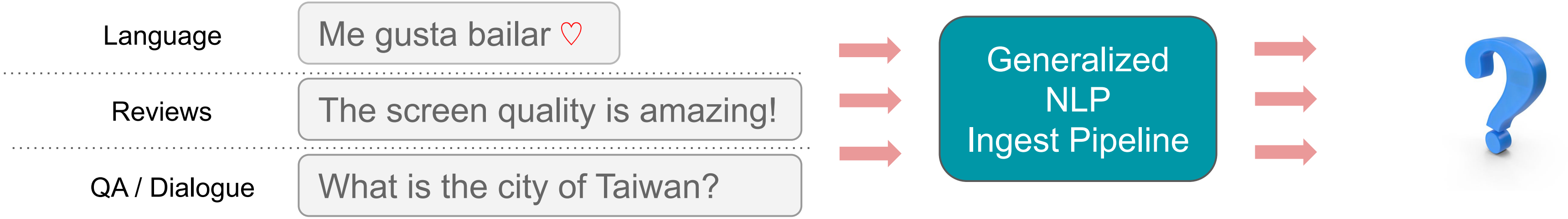
[<MENTION>, thanks, so, much, for, following, !, god, bless, !, <HASHTAG>]

[so, for, !, !]

[:-)]

Future Ideas

- Build and train ML model based on processed text and features
- Store ML model and use Logstash to ingest real-time profiles of online mental disorder cases via “I am diagnosed with X” filter
- What can we learn from natural language that generalizes to logs, metrics, etc.)? `55.3.244.1 GET /index.html 15824 0.043`
- Generalize pipeline to different conversations (chatbot, reviews, language etc.)



Language — Me gusta bailar ♡

Reviews — The screen quality is amazing!

QA / Dialogue — What is the city of Taiwan?

Generalized NLP Ingest Pipeline

?

References

- Elasticsearch 6.6 Reference
- Elastic Resources and Training
- Clinical NLP with Elasticsearch
- OpenNLP with the Elastic stack
- MIDAS: Mental illness detection and analysis via social media

Q&A

Demo