Fusing Elasticsearch with Neural Networks to Identify Personal Data

Ryan Turner
Senior ML Researcher,
Twitter

Rakshit Wadhwa
Senior Software Engineer,
Twitter
Part I: The Use Case
Motivation: Tracing and accounting for personal data

- Microservice-based companies distribute accountability for data privacy throughout an organization.
- **Annotations**, having a standard taxonomy for referring data columns are a key part of PDP compliance.
Challenges and Goals

- Challenges:
  - Distributed across numerous datasets and storage systems
  - Adhere to evolving privacy and data governance policies

- Goals:
  - Understand what data exists in our systems, its sensitivity, and its permitted purpose of use
  - Optimize for storage, discover new usage patterns, improve data security, and optimize data handling
Ideal Solution: Standardized taxonomy for schema

Problems:

- Changing schemas in legacy datasets often results in heavy refactoring across the consumers and producers of data.
- Long, painful, and error-prone process
Real Solution: Annotate, or “tag”, the columns

- Annotate in the background
  - Minimal refactoring
  - No downtime to existing tools/services

- Provide a probabilistic uncertainty with recommendations in the tool
Our Solution and use cases

- We build a probabilistic model on top of Elasticsearch for a recommender system
- Why might this be useful for you?
  - Systems for annotation recommendations useful at *any company* handling lots of personal data
  - Anytime you need a text-based lookup with quantified uncertainty, this architecture is a solution
## Example annotations

<table>
<thead>
<tr>
<th>dataset name</th>
<th>column name</th>
<th>column description</th>
<th>annotation</th>
<th>annotation definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>id</td>
<td>Unique identifier for the user</td>
<td>UserId</td>
<td>The identifier used internally for the identification of end-users generated by Twitter products.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dataset name</th>
<th>column name</th>
<th>column description</th>
<th>annotation</th>
<th>annotation definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>timelines</td>
<td>user_id</td>
<td>Twitter handle</td>
<td>Username</td>
<td>User's handle that appears on Twitter entities.</td>
</tr>
</tbody>
</table>
Annotations
(also known as PDTs or Personal Data Types)
2m fields across 100K active and inactive datasets
2m

Across 100K active and inactive datasets

Annotations (Personal Data Types)
Recommendation Engine - What are we predicting?

- Engineers/product managers annotate data manually over time
- Manually created taxonomy of 504 possible annotations
- Twitter scale:
  - 2 million columns spread across
  - ~100K primary and derived datasets
  - Each column had to be mapped against these 504 annotations
- Need an automated recommender to annotate new and old data
- Final annotation comes from user
Annotation Recommendation Service: Architecture

- Data Collection Engine
- Recommendation Engine
- ML Refresh Pipeline
- Annotation Service APIs
- Integrations
Part II: The System
The training corpus

- Manually-labeled data acts as a training corpus to automatically predict annotations for other (unlabeled) datasets
- Training corpus of ~70K records
- Augment descriptions with metadata ⇒ text-based feature vector
  
  Dataset Name, Column Name, Column Description, Annotation Name, Owner Team]

- Input: [Dataset Name, Column Name, Column Description]
- Output: set of recommended annotations for the column
## Example annotations

<table>
<thead>
<tr>
<th>dataset name</th>
<th>column name</th>
<th>column description</th>
<th>annotation</th>
<th>annotation definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>id</td>
<td>Unique identifier for the user</td>
<td>UserId</td>
<td>The identifier used internally for the identification of end-users generated by Twitter products.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dataset name</th>
<th>column name</th>
<th>column description</th>
<th>annotation</th>
<th>annotation definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>timelines</td>
<td>user_id</td>
<td>Twitter handle</td>
<td>Username</td>
<td>User's handle that appears on Twitter entities.</td>
</tr>
</tbody>
</table>
Step 1: Reduce to a full-text search problem

- Converted our corpus into inverted search indexes
- Leverage existing solutions (Elasticsearch)
- Training: Turned the data into synthetic “documents” by concatenating the metadata for columns with the same annotations
- Test: Do multiple variations of the Elasticsearch queries
- ⇒ Need a calibration model to convert multiple confidence scores into a single probability
Reduce to a full-text search problem

<table>
<thead>
<tr>
<th>Text-based input features</th>
<th>Categorical Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>column names</td>
<td>annotation</td>
</tr>
<tr>
<td>column descriptions</td>
<td></td>
</tr>
<tr>
<td>dataset names</td>
<td></td>
</tr>
<tr>
<td>owner teams</td>
<td>Username</td>
</tr>
<tr>
<td>profile name, handle of the user,</td>
<td></td>
</tr>
<tr>
<td>public name</td>
<td>Profile, Feed, Ads</td>
</tr>
<tr>
<td>ads_prediction</td>
<td></td>
</tr>
</tbody>
</table>
Step 2: The calibration model

- Held out 20% of the training corpus
- Elasticsearch returns scores based on Lucene's practical scoring function, TF-IDF similarity for each class
- These Elasticsearch result scores become a numeric feature vector of 504 dimensions for the calibration model
  - ⇒ Need a multi-class classification model
- Also fuse together results from different variations of ES queries
- Experimented with different multi-classification models and decided to use an artificial neural net model
The setup
Performance results

Top-10 accuracy percentage vs. Models

Negative Log Likelihood (nats) vs. Models
73.8% of annotated datasets used one or more PDT recommendations
Conclusions

- Using ML models and dataset metadata/schemas we developed a recommendation engine to annotate legacy datasets at Twitter to a new standardized taxonomy
- New annotations: 73.8% use the recommendations directly from the service.
- Facilitates the development of tools for data discovery and data auditing and handling
- Auditing and handling tools help to understand the sensitivity of the accessed data, which allows teams to align data permissions based on sensitivity