Employer Industry Classification using Job Postings

Mahak Goindani (Intern from Purdue University)
Qiaoling Liu (CareerBuilder)
Josh Chao (CareerBuilder)
Valentin Jijkoun (Textkernel)
Employer Industry Classification

- Employer industry of jobs => insight about demand per industry
- Current system at CareerBuilder
  - CompanyDepot: Employer name normalization system
  - Employer knowledge base (KB): ~20M entities
- Errors occur but manual detection is impossible
- Observation: Job postings can be helpful
Clues from Job Postings

1. “Truck Driver” jobs indicate employer industry is Transportation

2. Company Name contains keyword “Transportation”
Automatic Error Detection based on Job Postings

```
[{
  "organization_name",
  "organization_industry",
  "profession",
  "count"
},
{
  "Hornady Transportation LLC",
  {
    "value": "56",
    "label": "Administrative and Support and Waste Management and Remediation Services"
  },
  {
    "value": 530,
    "label": "Truck Driver"
  },
  112
},
{
  "Hornady Transportation LLC",
  {
    "value": "56",
    "label": "Administrative and Support and Waste Management and Remediation Services"
  },
  {
    "value": 532,
    "label": "Delivery Driver"
  },
  2
}
]
```
Dataset and Method

- **Jobfeed API:** [https://us.jobfeed.com/api/v3/help](https://us.jobfeed.com/api/v3/help)
  - Get top 10K employers with most jobs in Apr-Jun 2017
  - Consider top two industries

<table>
<thead>
<tr>
<th>Industry</th>
<th>Positive Class</th>
<th>Negative Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transportation</td>
<td>516 (5.16%)</td>
<td>9484 (94.84%)</td>
</tr>
<tr>
<td>Health Care</td>
<td>1318 (13.18%)</td>
<td>8682 (86.82%)</td>
</tr>
</tbody>
</table>

- **Build a binary classifier per industry**
  - 70% for training, 10% for validation, 20% for testing
  - Use “legacy labels” for training
Feature Generation

For each employer e, we combine

- **Title feature vector**
  - Job titles posted by the employer => 5426 normalized titles
  - $V_{ti}$: Significant titles for industry i
  - $f_{e,t} = \begin{cases} 
    R_{e,t}, & R_{e,t} \geq 0.01 \\
    0, & \text{otherwise} 
  \end{cases}$ ($R_{e,t}$ is percentage of t in all jobs posted by e)

- **Keyword feature vector**
  - Keywords in the employer name (including raw names and normalized name)
  - $V_{wi}$: Significant keywords for industry i
  - $f_{e,w} = \begin{cases} 
    1, & w \in N_e \\
    0, & \text{otherwise} 
  \end{cases}$ ($N_e$ is list of unigrams in the employer name)

Assumption: main jobs posted by an employer are related to the industry
Top Keywords and Titles

(1) Transportation Industry

<table>
<thead>
<tr>
<th>Vocabulary Size</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titles</td>
<td>140</td>
</tr>
<tr>
<td>Keywords</td>
<td>45</td>
</tr>
<tr>
<td>Titles</td>
<td>Truck Driver, Hauler, Deck Officer, Independent Contractor ((Transportation and Material Moving)), Flatbed Driver</td>
</tr>
<tr>
<td>Keywords</td>
<td>carriers, airlines, trucking, transport, freight, transfer</td>
</tr>
</tbody>
</table>

(2) Health Care Industry

<table>
<thead>
<tr>
<th>Vocabulary Size</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titles</td>
<td>383</td>
</tr>
<tr>
<td>Keywords</td>
<td>50</td>
</tr>
<tr>
<td>Titles</td>
<td>Dialysis Technician, Adult Case Manager, Inpatient Services Nurse, Rehabilitation Aide, Triage Nurse</td>
</tr>
<tr>
<td>Keywords</td>
<td>hospice, clinic, rehabilitation, hospital, medical, health</td>
</tr>
</tbody>
</table>
Learning Algorithms

• Support Vector Machine (SVM)
  • LibSVM package
  • Cost-sensitive learning: different costs for misclassifications in different classes

• Gradient Boosted Decision Trees (GBDT)
  • MLlib package
Experiments

• Label partial test set to save effort => Disagreement Set
  
  • Ignoring cases when predictions agree with legacy label

<table>
<thead>
<tr>
<th></th>
<th>Transportation</th>
<th>Health Care</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test set</td>
<td>1989</td>
<td>1991</td>
</tr>
<tr>
<td>Disagreement set</td>
<td>58</td>
<td>217</td>
</tr>
</tbody>
</table>

• An employer in disagreement set is “defined” if correctly normalized, otherwise “undefined”.
Results for Classifying Defined Employers

- Legacy system performs best
  - Higher recall
  - Used manually edited KB
- SVM and GBDT perform similar
  - Much better than baselines

Can the ML models detect errors in the legacy system?

<table>
<thead>
<tr>
<th>(1) Transportation Industry</th>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy System</td>
<td>0.94</td>
<td>0.92</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>0.97</td>
<td>0.70</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>GBDT</td>
<td>1.00</td>
<td>0.65</td>
<td>0.79</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(2) Health Care Industry</th>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy System</td>
<td>0.96</td>
<td>0.82</td>
<td>0.89</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>0.97</td>
<td>0.71</td>
<td>0.82</td>
<td></td>
</tr>
<tr>
<td>GBDT</td>
<td>0.98</td>
<td>0.76</td>
<td>0.86</td>
<td></td>
</tr>
</tbody>
</table>

Note that we calculate precision, recall and f-score for all the manually identified "undefined" employers.
Utility Metrics for Detecting Undefined Employers

• Given a disagreement set $D$ computed based on multiple models:
  - $E_{U|D}$: # of “undefined” employers annotated in the disagreement set $D$
  - $E_{M|D}$: # of employers that occur in the disagreement set of a specific model

• Utility Precision ($U_{P|D}$):
  - % of employers annotated as “undefined” in the disagreement set of the model

• Utility Recall ($U_{R|D}$):
  - % of “undefined” employers in the disagreement set of the model among all the annotated “undefined” employers

• Utility F-score ($U_{F|D}$):
  - Harmonic mean of Utility Precision and Utility Recall

$$U_{P|D} = \frac{E_{M|D} \cap E_{U|D}}{E_{M|D}}$$

$$U_{R|D} = \frac{E_{M|D} \cap E_{U|D}}{E_{U|D}}$$

$$U_{F|D} = \frac{2 \times U_{P|D} \times U_{R|D}}{U_{P|D} + U_{R|D}}$$
Utility for Detecting Undefined Employers

GBDT performs better, capturing the complex relationships between features
• E.g., related job titles: “Truck Driver” and “Commercial Driver’s License Driver”
Error Analysis

• Automatically detected important errors in legacy system

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Transportation</th>
<th>Health Care</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect Name Extraction</td>
<td>7 (31.82%)</td>
<td>18 (15.38%)</td>
</tr>
<tr>
<td>Incorrect Name Normalization</td>
<td>2 (9.09%)</td>
<td>40 (34.19%)</td>
</tr>
<tr>
<td>Incorrect KB Attribute</td>
<td>13 (59.09%)</td>
<td>59 (50.43%)</td>
</tr>
</tbody>
</table>

• “Truck Driver Cdl-a” => not valid employer name
• “Omni Specialized LLC” => wrong normalization: “OWI Specialized, Inc”
• “Hornady Transportation LLC” => wrong industry: “Temporary Help Services”

• Model errors <= Our assumption does not hold
  • “CGB Enterprises, Inc.” => jobs: “Landscape Laborer” and “Operations Manager”
Conclusion

• Inferring employer industry of jobs is important
  • CareerBuilder uses an employer normalization system and an employer KB
  • Manual detection of errors is hard

• We used job postings and machine learning methods to infer employer industry and automatically detect errors in legacy system
  • Titles from job postings
  • Keywords from employer name
Future Work

• Add more features, e.g.,
  • Employer description
  • Text of job description

• Investigate more industries and try multiclass classification
  • Staffing industry: our assumption does not hold

• Handle employers spanning across multiple industries
  • Multi-label classification or fuzzy classification
Thank you!

Any Questions?
Backup Slides
How to Compute Significant Titles per Industry?

• Main idea: t is more significant if it appears more often in positive class

• Compute significance score $S_{ti} = \frac{f_{tpi}}{f_t}$ for industry i
  
  • $f_t$ is frequency of t across all employers, and $f_{tpi}$ focuses on positive class of i

• Compute significant titles $V_{ti} = \{ t \mid S_{ti} \geq \theta_{si}, f_t \geq \theta_{fi} \}$
  
  • $\theta_{si}$ and $\theta_{fi}$ are chosen based on quartiles
    1. Filter out titles that have $f_{tpi} \leq 1$
    2. Compute median of $S_{ti}$ as $\theta_{si}$ and median of $f_t$ as $\theta_{fi}$
    3. If $|V_{ti}| < 50$, reduce $\theta_{fi}$ to 1st quartile of $f_t$ and keep $\theta_{si}$ unchanged

Vocabulary of significant keywords per industry $V_{wi}$ is computed similarly.
Feature Ablation

RESULTS FOR DIFFERENT FEATURE SETS (TRANSPORTATION INDUSTRY)

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>SVM</td>
<td>0.95</td>
<td>0.66</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>GBDT</td>
<td>1.00</td>
<td>0.58</td>
<td>0.73</td>
</tr>
<tr>
<td>Titles</td>
<td>SVM</td>
<td>0.97</td>
<td>0.70</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>GBDT</td>
<td>0.97</td>
<td>0.67</td>
<td>0.79</td>
</tr>
<tr>
<td>Titles &amp; Keywords</td>
<td>SVM</td>
<td>0.97</td>
<td>0.70</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>GBDT</td>
<td>1.00</td>
<td>0.65</td>
<td>0.79</td>
</tr>
</tbody>
</table>

• Titles capture more information than employer name alone
• Adding keywords from employer names does not help much