Apollo: Near-Duplicate Detection for Job Ads

Authors: Hunter Burk, Faizan Javed and Janani Balaji
Presenter: Ozgur Ozturk   @DrOzturk in/DrOzturk
Near-Duplicate Detection (NDD)

• Applications: Web Search, plagiarism detection, digital libraries, spam email detection

• Duplicates
  • hinder user experience
  • Increase costs
    • storage
    • technical operations
    • maintenance
Supply & Demand Analytics

• Employers want to know:
  • who are their biggest competitors
  • what are emerging job titles and skills
  • which platform would give their job ad more visibility

• Job ad duplication/variation
  • change over time between postings at different feed sources
  • A/B testing on varied titles and descriptions
  • site formatting disparities
  • geographical differences, same job at two branches in same city
  • copy-paste-edit job ads
Related Work

• Related Work:
  • Syntactic (character sequence similarity)
    • Hash signatures of word or character sequences
  • Semantic (meaning)
    • Semantic graphs with syntactic NDD methods
  • URL based
    • Rule mining to find URLs pointing to same document
Related Work: Syntactic NDD methods

• Shingling
  • Shingles: overlapping, fixed-size, sliding windows of subsequences
  • Shingles encoded using a fingerprinting scheme for efficient matching
  • Super shingles to improve storage and run time
  • similarity $(d, d') = |S(d) \cap S(d')| / |S(d) \cup S(d')|$
    i.e. Percentage of unique shingles two pages $d$ and $d'$ share

• Random Projection (SimHash)
  • Locality sensitive hashing
    maps high-D feature space of documents to a fixed-sized fingerprint
  • Document similarity is hamming similarity between two fingerprint ratio of matching bits in their projections
Related Work: hybrid algorithm

- Shingling
  - ignores repetition of terms
  - higher recall

- SimHash
  - ignores the order of terms
  - more efficient document representation
  - higher precision
Fingerprint construction methods and algorithms (right)

Other Domain Specific NDD Applications

- SpotSigs:
  - news
  - Titles mostly stop-word free
- ANDD
  - news
  - email spam
Apollo: Near-Duplicate Detection for Job Ads
270 million job ads from 5 sources

Monthly Apollo Run (16 hours)

63 million deduplicated job ads
Defining Near Duplicate Job Ads

• Ads for different geographical locations => not same

• Title alone not enough
  • “Software Engineer” needed for 2 teams
    • Experience level, skills needed will differ
    • Incorporate description in similarity measure

• Preprocess to remove boilerplate text
  • Usually same in all ads from same employer
    • company’s mission, culture, history, and benefits etc.
Apollo

Parse Feeds

Normalize Geography

Normalize Company

Classify Skills

Normalize Job Title

Deduplicate

Analytics Products

Internal Client Lead Generation

Custom Client Feeds
Steps of Apollo

• **Step 1**: Find exact match between two job ads
  - match on city, state, country, company name, job title, and job description
  - 20% reduction

• **Step 2**: Blocking (for map)
  - We did not base on having a common shingle, since large blocks form
    - Comparisons within a block is quadratic
  - Mapping key concatenation of SOC Major, company name, geolocation (city, state, and country)
    - all normalized via existing services
Steps of Apollo

• **Step 3**: Shingling (via Lucene, n=5), ex:

  Hiring a Sr Data Scientist in New Orleans LA USA read on

  The resulting 5-shingle set would be:

  Hiring a Sr Data Scientist
  a Sr Data Scientist in
  Sr Data Scientist in New
  Data Scientist in New Orleans
  Scientist in New Orleans LA
  in New Orleans LA USA
  New Orleans LA USA read
  Orleans LA USA read on
Steps of Apollo

• **Step 4**: Boilerplate removal
  • weed out shingles common in all descriptions in a block
  • Only if
    - more than 3 jobs are in block
    - at least 10 shingles remain.
Steps of Apollo

• **Step 5:** Document Similarity
  • a) Jaccard on shingles of document 1 and 2

\[
JS (D_1, D_2) = \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|}
\]

• b) filter by empirically set minimum Jaccard Similarity thresholds
no boilerplate text removed, and no job title match \( \Rightarrow 90\% 
boilerplate text removed or a job title match \( \Rightarrow 50\% 
boilerplate text removed and job title match \( \Rightarrow 25\% 

Evaluation

• Compared Apollo against two baseline
  • SimHash and a basic shingle comparison based on Jaccard Similarity with one threshold
    • also implemented as MapReduce jobs
  • F-score, precision, recall, and run time

• Data: 7.5 million US Jobs Ads from 4 sources posted in March 2017
Evaluation

• For each method
  • Exact job ad matches were removed from evaluation
  • 500 detected as match and 500 as non-match human evaluated

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
<th>Runtime (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimHash</td>
<td>0.2715</td>
<td>0.3299</td>
<td>0.2979</td>
<td>20</td>
</tr>
<tr>
<td>Shingling</td>
<td>0.3533</td>
<td>0.5694</td>
<td>0.4360</td>
<td>57</td>
</tr>
<tr>
<td>Apollo</td>
<td>0.3666</td>
<td>0.5975</td>
<td>0.4544</td>
<td>56</td>
</tr>
</tbody>
</table>
Future Work

• Build and utilize weighted word dictionaries for jobs domain
• Hybrid algorithm