USPTO INVENTOR DISAMBIGUATION

Institute of Scientific and Technical Information of China

SEP 20, 2015

TEAM MEMBER:
Yang GuanCan
Zhang Jing
Cheng Liang
Zhang HaiChao
Lv LuCheng
Wang DaoRen
Content

1. Data Preparation
2. Features Selection
3. Computational Scale Reduction
4. Pair Comparison
5. Mixed Method
Why is USPTO Inventor Disambiguation?

- **USPTO inventor data is available**
  Patent data made available by the USTPO enables further research into technology and innovation.

- **Inventor disambiguation is non-trivial**
  Patents filed at USPTO have no consistent and unique identifiers for inventors.

- **Inventor disambiguation open new avenues of study**
  Inventors performance; International flow of talents; Structure and Policy of innovation.
Data Preparation

- Extract inventor, assignee, location, NBER subcategory and family datasets from primary and secondary data sources
- Standardize and cleansing
- Integrating the multiple datasets into the inventor-patent database
In our algorithm, the base unit of analysis in inventor disambiguation is an inventor-patent instance. Each record contains attributes used for disambiguation.

An inventor-patent instance consists of 14 fields, which be divided into:

- Inventor related features
- Patent related features
(1) Inventor related features

- 1. name_first: the first name field of the inventor record;
- 2. name_middle: the middle name field of the inventor record;
- 3. name_last: the last name field of the inventor record;
- 4. name_last_f: the inventor's last name plus the initial of first name;
- 5. name_first_m: the inventor's first name plus the initial of middle name;
- 6. name_full: combine the first name, middle name, last name and suffix;
- 7. name_first_sdx: soundex code of the first name of the inventor record;
- 8. name_last_sdx: soundex code of the last name of the inventor record.
(2) Patent related features

- **city**: the city name field of the inventor record
- **country**: the county abbreviation field of the inventor record
- **assignee**: the closed assignee corresponding to the patent of inventor record
- **subcategory_id**: NBER technology subcategories corresponding to the patent of inventor record
- **inpadoc_family_id**: INPADOC family id corresponding to the patent of inventor record (from PATSTAT)
- **docdb_family_id**: Docdb family id corresponding to the patent of inventor record (from PATSTAT)
Each record represents an inventor-patent instance

Number of distinct attribute values of inventor-patent instances
Implement of reducing computational scale

- To avoid a prohibitively expensive comparison of all pairs of records, three techniques we implemented to partition the records into smaller subsets.
  
  1. Initial deduplication directly according to exact matching rule
  2. Blocking, makes sure that whenever we do a comparison in our algorithm it is done among only potential matches
  3. To further control the memory consumption, we split all the inventor data into 26 parts according the alphabet.
(1) Initial Deduplication

- In fact, most of inventor’s name at his career have no change, so it is no nessessary to spend prohibitively expensive computational cost to this part of data.

- we adopt a rule of thumb; with a strict matching rule to initial identify some duplicated inventor’s id.

- “name_first+name_middle+name_last+assignee+city”

- After this initial deduplication, the whole amount of inventor-patent instances reduced from 12,392,012 to 6,251,305, at the same time, this simple way is accompanied by the low error rate.
There are two strategies implemented during our blocking processes. As the common cases, this blocking strategy requires that two records must agree in:

- (1) first name and last name, or
- (2) name_last_f and city, or
- (3) name_last_f and assignee name to appear in the resulting set of comparison patterns.

In terms of some special parts, such as "S","L", we take a more strict blocking strategy.
At the same time, to further control the memory consumption, we split all the inventor data into 26 parts according the alphabet.

- A Blocking with the first character of last name is “A”
- “B” block
- ……
- “Z” block
<table>
<thead>
<tr>
<th>patent_id</th>
<th>sequence</th>
<th>ind</th>
<th>name_full</th>
<th>assignee</th>
<th>city</th>
<th>subcategory_id</th>
<th>impadoc_family_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>50992500</td>
<td>4</td>
<td>1915</td>
<td>Tousan Chaudan</td>
<td>Motorola Inc</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5390142</td>
<td>0</td>
<td>1916</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5602791</td>
<td>0</td>
<td>1916</td>
<td>Shimon Gendlin</td>
<td>Kappa Numerics Inc</td>
<td>Jerusalem</td>
<td>24</td>
<td>914203</td>
</tr>
<tr>
<td>5673220</td>
<td>0</td>
<td>1916</td>
<td>Shimon Gendlin</td>
<td>Kappa Numerics Inc</td>
<td>New York</td>
<td>24</td>
<td>220511</td>
</tr>
<tr>
<td>5707887</td>
<td>0</td>
<td>1916</td>
<td>Shimon Gendlin</td>
<td>Kappa Numerics Inc</td>
<td>Jerusalem</td>
<td>46</td>
<td>914203</td>
</tr>
<tr>
<td>5717235</td>
<td>0</td>
<td>1916</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5718983</td>
<td>0</td>
<td>1916</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5841699</td>
<td>0</td>
<td>1916</td>
<td>Shimon Gendlin</td>
<td></td>
<td>Westbury</td>
<td>24</td>
<td>46940055</td>
</tr>
</tbody>
</table>

**Blocking**
Pair Comparison

- We compare pairs of inventor records and determine if each pair is a match (the same unique inventor) or a non-match (two non-unique inventors). We implement three kinds of methods to compute the similarity of pairs:
  - (1) Jaro-Winkler string similarity
  - (2) Exact matching
  - (3) Exact matching performed on the SoundEx abbreviations of the pairs
<table>
<thead>
<tr>
<th>id</th>
<th>patent_id</th>
<th>sequence</th>
<th>name_first</th>
<th>name_middle</th>
<th>name_last</th>
<th>name_last_first</th>
<th>name_last_sdx</th>
<th>name_last_sdx</th>
<th>name_full</th>
<th>assignee</th>
<th>city</th>
<th>country</th>
<th>is_match</th>
</tr>
</thead>
<tbody>
<tr>
<td>3313</td>
<td>5692791</td>
<td>0</td>
<td>Shimyon</td>
<td>GENDLIN</td>
<td>Gendlin S</td>
<td>S550</td>
<td>GS34</td>
<td>Gendlin</td>
<td>Shimyon Gendlin</td>
<td>RAFA NUMERICS INC</td>
<td>Jerusalem</td>
<td>IL 24</td>
<td>1</td>
</tr>
<tr>
<td>3313</td>
<td>5692791</td>
<td>0</td>
<td>Shimyon</td>
<td>GENDLIN</td>
<td>Gendlin S</td>
<td>S550</td>
<td>GS34</td>
<td>Gendlin</td>
<td>Shimyon Gendlin</td>
<td>RAFA NUMERICS INC</td>
<td>New York</td>
<td>US 24</td>
<td>1</td>
</tr>
<tr>
<td>3313</td>
<td>5692791</td>
<td>0</td>
<td>Shimyon</td>
<td>GENDLIN</td>
<td>Gendlin S</td>
<td>S550</td>
<td>GS34</td>
<td>Gendlin</td>
<td>Shimyon Gendlin</td>
<td>RAFA NUMERICS INC</td>
<td>Jerusalem</td>
<td>IL 24</td>
<td>1</td>
</tr>
<tr>
<td>3408</td>
<td>5797807</td>
<td>0</td>
<td>Shimyon</td>
<td>GENDLIN</td>
<td>Gendlin S</td>
<td>S550</td>
<td>GS34</td>
<td>Gendlin</td>
<td>Shimyon Gendlin</td>
<td>RAFA NUMERICS INC</td>
<td>Jerusalem</td>
<td>IL 46</td>
<td>1</td>
</tr>
<tr>
<td>4032</td>
<td>5041605</td>
<td>0</td>
<td>Shimyon</td>
<td>GENDLIN</td>
<td>Gendlin S</td>
<td>S550</td>
<td>GS34</td>
<td>Gendlin</td>
<td>Shimyon Gendlin</td>
<td>RAFA NUMERICS INC</td>
<td>Westbury</td>
<td>US 24</td>
<td>1</td>
</tr>
</tbody>
</table>

**Pair Comparison**
the inventor disambiguation is very complex problem, sometimes, a single supervised machining learning will not gain the good performance under special cases:

Training data is limited and cannot cover the main variation patterns, or some special name distribution.

Sometimes, a classification error may cause a very expensive mistake. Such as there are two inventors group existed, at this time, one wrong inventor pair cause the two subgroup be connected.
A Mixed Method

- Method 1: AdaBoost supervised learning method
- Method 2: Stochastic record linkage
- Method 3: Rule-based method
- Method 4: Graph-based clustering
we choose a AdaBoost model as the method of pair similarity classification is to find a boundary contour of the whole inventor network.

All links are based on the prediction results of pair comparison model. If the result show matching, then the ends of pair will be linked.

Sometimes, the boundary area is hardy to distinguish, and some pendants need use transitive closure.
Basic assumptions: a single supervised machining learning will not gain the good performance.

So we choose a Stochastic record linkage to calculate the weight of pair similiary.

Then we select some pairs with highly weight by a threshold, and trusted them would be correct pairs.
Method3: Rule-based Method

- Rule 1: The pairs with similarity weight greater than or equal 0.78 will be saved as candidate match pairs.
- Rule 2: The pairs with similarity weight greater than or equal 0.70 and have the exact matching first character of the middle name will be saved as candidate match pairs.
- Rule 3: The pairs with similarity weight greater than or equal 0.70 and have the same subcategory_id will be saved as candidate match pairs.
- Rule 4: The pairs with the same inpadoc_family_id will be saved as candidate match pairs.
- Rule 5: The pairs with the docdb family citation relationship will be saved as candidate match pairs.

In final, we consolidate all these candidate match pairs and using red links to present.
At final, we just choose some pairs links with high confidents to construct the inventor network.

All linked inventor pairs group is a complete graph.
Assigning an unique identifier

- Assigning an unique identifier for each inventor

- Grouping the inventors by graph clustering.

- Assigning each inventor-patent instance in inventor_dedup dataset a clustered group identifier

- Assigning each inventor-patent instance in the whole inventor dataset a clustered group identifier

- In rare cases, we enforce to extract the first group identifier among Results:

- Through the above inventor disambiguation process, 3,631,364 unique inventor identifiers were achieved.
Thanks for your attention