An approach for inventor name disambiguation in patent data

Luciano Kay, Ph.D.
Carlos Mozzati

InnovationPulse

PatentsView Inventor Disambiguation Workshop
USPTO Madison Auditorium
600 Dulany Street
Alexandria, VA
September 24, 2015
Introduction

• InnovationPulse
  – New technology & competitive intelligence company
  – Founded in 2014
  – Santa Barbara, CA

• PatentsView Inventor Disambiguation Workshop
  – Concrete problem to be solved
  – Competition & collaboration
  – An opportunity to further investigate and test our real time applications
Iterative development process

1. Understand the data
2. Understand common name cases
   – Own analysis of sample patent data
   – Other works
     • E.g. Kopcke & Rahm (2010); Chin et al. (2014)
3. Develop algorithm based on common cases
   – Cloud based, in-memory approach
4. Test & improve
## Selected examples and disambiguation requirements

<table>
<thead>
<tr>
<th>Case</th>
<th>Example 1 (First name; Last name)</th>
<th>Example 2 (First name; Last name)</th>
<th>Percent of sample affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyphenated (or double) last names</td>
<td>Guy; Cases-Langhoff</td>
<td>Guy; Cases Langhoff</td>
<td>2.64%</td>
</tr>
<tr>
<td>Use of special characters</td>
<td>Mathieu André; De Bas</td>
<td>Mathieu Andre; De Bas</td>
<td>0.88%</td>
</tr>
<tr>
<td>Missing or misplaced titles, prefixes and suffixes</td>
<td>Jr Yuan; Huang John T.; Carroll, III</td>
<td>Yuan; Huang Jr John T.; Carroll</td>
<td>&lt;0.2%</td>
</tr>
<tr>
<td>Shortened names</td>
<td>John Nicholas; Gross</td>
<td>John N.; Gross</td>
<td>&lt;0.2%</td>
</tr>
<tr>
<td>Incomplete names(^a)</td>
<td>Richard B.; Robbins</td>
<td>Richard; Robbins</td>
<td>&lt;0.2%</td>
</tr>
<tr>
<td>Subset names(^b)</td>
<td>Xudong; Xi Chen</td>
<td>Xudong Tao; Xi Chen</td>
<td>&lt;0.2%</td>
</tr>
<tr>
<td>Romanized and short form of names</td>
<td>Tatjana (Tanja); Barth</td>
<td>Tatjana; Barth</td>
<td>&lt;0.2%</td>
</tr>
</tbody>
</table>

Notes: a. this rule does not apply to Chinese names, as they rarely omit part of the name.
Source: own analysis based on data extracted from sample of USPTO patent application and grant XML raw records
Algorithm requirements

1. Load data into memory
2. Clean up and pre-process data
3. Create comparison groups
4. Compare inventor names
5. Produce output
Comparison groups

• Name 1: “Kevin Edward Poole”
  – Kevin Edward
  – Edward Poole
  – Kevin Poole

• Name 2: “Kevin E. Poole”
  – Kevin E.
  – E. Poole
  – Kevin Poole

Examples of comparison groups based on 2 terms extracted from each inventor full name

Both Name 1 and Name 2 are members of group “Kevin Poole”
Comparison rules

- **Exact match** (e.g. “Kevin Poole” and “Kevin Poole”)

- **Same sets of words** (regardless of their order) (e.g. “Kevin Edward Poole” and “Edward Kevin Poole”)

- **Shortened names** (e.g. “Kevin E. Poole” and “Kevin Edward Poole”) (applied to non-Chinese names only)

- **Subset names** (e.g. “Shi Chin Wenfeng” and “Shi Chin Zsu Wenfeng”) (applied to Chinese names only)

- **Almost identical names with same assignee organization** (e.g. “Kevin E. Poole” and “Kevin Poole”, both with “Apple Inc”)

- **Almost identical names with same technology category** (e.g. “Kevin E. Poole” and “Kevin Poole”, both working on 3-digit “A61” CPC class)
Names comparison

```plaintext
finished_group = true
while (k < group_members_count AND finished_group)
    for j = k+1 to group_members_count
        if name_j is not in output.csv then finished_group = false
            compare name_k = name_j using comparison rules
            if there is a match then
                ID_j = ID_k
                end comparisons (don't apply other rules)
            end if
        end if
    next j
    if name_k is not in output.csv then name_k to output.csv
    k++
```
Computing setup

- AWS EC-2 R3 instance
  - “r3.8xlarge” instance with 32 vCPU (virtual CPUs)
  - 244 GiB of RAM memory
  - 2 x 320 SSD storage
  - Amazon Linux AMI 2015.03.1 (HVM) 64-bit, SSD Volume Type (ami-d5c5d1e5) machine image
- Redis
- C, libraries

Setup matches specific workshop requirements and application
Concluding remarks

• Our goal here: Concrete problem-solving (e.g. disambiguation) as an opportunity to investigate real time applications

• Preliminary tests
  – Only tested with Trajtenberg et al., 2008
  – High recall scores, unsatisfactory precision affects F1
  – Runtime can be improved significantly

• Next steps
  – More work on comparison groups (new rules, weights)
  – Use pattern matching to identify inventor country of origin
  – Use technology categories instead of IPC classes (e.g. Kay et al., 2014)
  – Use disambiguated organization names
References


InnovationPulse®
Santa Barbara, CA
info@innovationpulse.com
www.innovationpulse.com