# INFOMDW: Data Wrangling

# Heterogeneous Data Analysis

**Hakim Qahtan**

Department of Information and Computing Sciences

Utrecht University

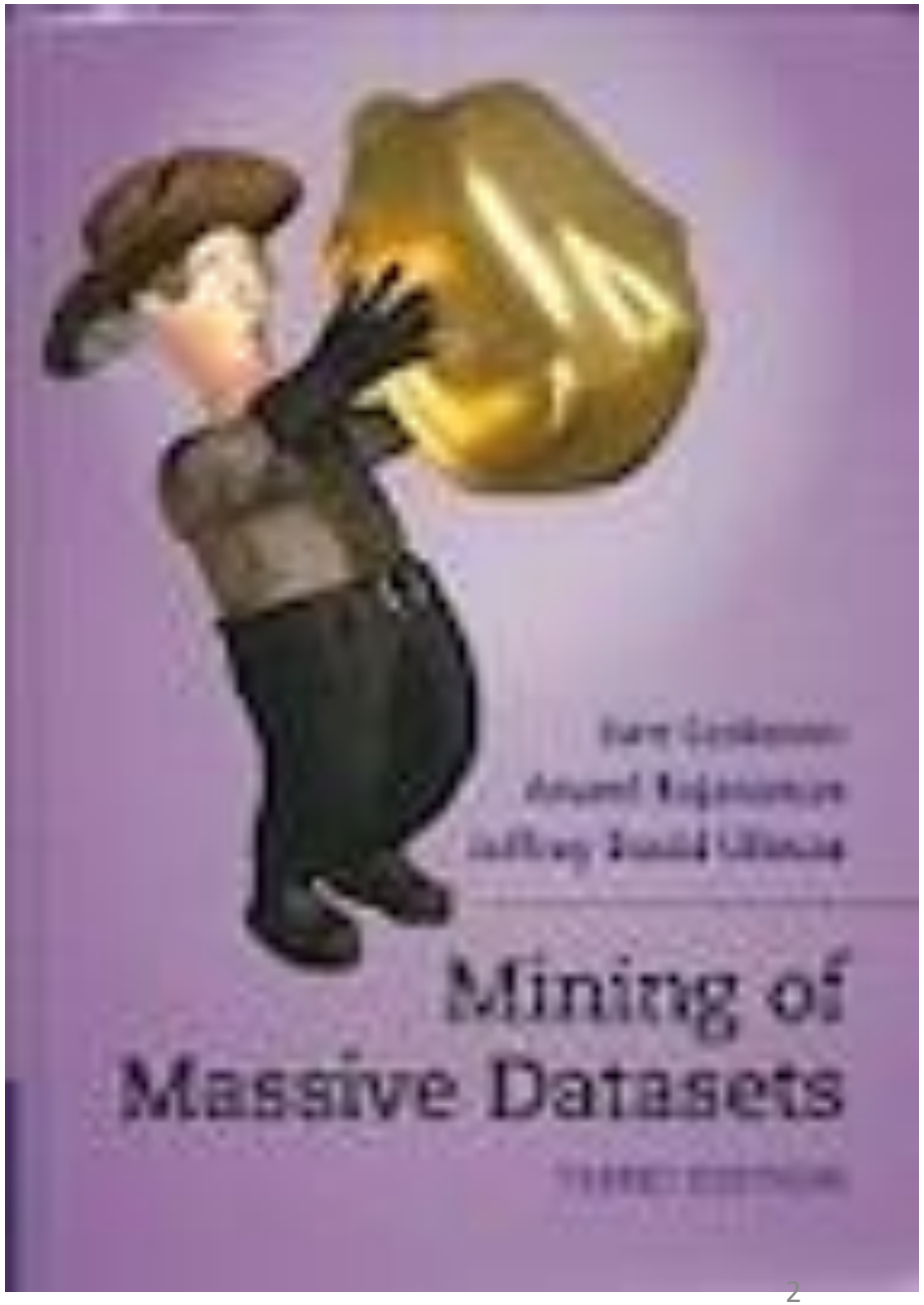
# Reading Material for Today

- Mining of Massive Datasets

by Jure Leskovec, Anand Rajaraman, Jeff Ullman

http://www.mmds.org

Chapter 3.1 – 3.5



# Entity Linkage

# The Wrong Paris










***How many names, descriptions are used for the same real-world "entity"?***



## *How many names, descriptions are used for the same real-world "entity"?*



London 런던 ᲚᲝᲜᲓᲝᲜᲘ लंडन लंदन ᲚᲔᲜᲓᲝᲜ ለንደን ロンドン
লন্ডন ลอนดอน இலண்டன் ლონდონი Llundain
Londain Londe Londen Londen Londen Londinium
London Londona Londonas Londoni Londono Londra
Londres Londrez Londyn Lontoo Loundres Luân Đôn
Lunden Lundúnir Lunnainn Lunnon لندن لندن لندن لوندون
לאנדאן לונדון Λονδίνο Лёндан Лондан Лондон Лондон
Лондон Լոնդոն 伦敦 …

## *How many names, descriptions are used for the same real-world "entity"?*



London 런던 ᲚᲝᲜᲓᲝᲜᲘ लंडन लंदन ᲚᲔᲜᲓᲝᲜ ለንደን ロンドン
লন্ডন ลอนดอน இலண்டன் ლონდონი Llundain
Londain Londe Londen Londen Londen Londinium
London Londona Londonas Londoni Londono Londra
Londres Londrez Londyn Lontoo Loundres Luân Đôn
Lunden Lundúnir Lunnainn Lunnon لندن لندن لندن لوندون
לאנדאן לונדון Λονδίνο Лёндан Лондан Лондон Лондон
Лондон Լոնդոն 伦敦 …

capital of UK, host city of the IV Olympic Games, host city of the XIV Olympic Games, future host of the XXX Olympic Games, city of the Westminster Abbey, city of the London Eye, the city described by Charles Dickens in his novels, …

## *How many names, descriptions are used for the same real-world "entity"?*



London 런던 ᲚᲝᲜᲓᲝᲜᲘ लंडन लंदन ሎንደን ለንደን ロンドン
লন্ডন ลอนดอน இலண்டன் ლონდონი Llundain
Londain Londe Londen Londen Londen Londinium
London Londona Londonas Londoni Londono Londra
Londres Londrez Londyn Lontoo Loundres Luân Đôn
Lunden Lundúnir Lunnainn Lunnon لندن لندن لندن لوندون
לאנדאן לונדון Λονδίνο Лёндан Лондан Лондон Лондон
Лондон Լոնդոն 伦敦 …

capital of UK, host city of the IV Olympic Games, host city of the XIV Olympic Games, future host of the XXX Olympic Games, city of the Westminster Abbey, city of the London Eye, the city described by Charles Dickens in his novels, …

http://sws.geonames.org/2643743/
http://en.wikipedia.org/wiki/London
http://dbpedia.org/resource/Category:London
…

# … or …

***How many "entities" have the same name?***

- London, KY
- London, Laurel, KY
- London, OH
- London, Madison, OH
- London, AR
- London, Pope, AR
- London, TX
- London, Kimble, TX
- London, MO
- London, MO
- London, London, MI
- London, London, Monroe, MI
- London, Uninc Conecuh County, AL
- London, Uninc Conecuh County, Conecuh, AL
- London, Uninc Shelby County, IN
- London, Uninc Shelby County, Shelby, IN
- London, Deerfield, WI
- London, Deerfield, Dane, WI
- London, Uninc Freeborn County, MN
- …

# … or …

***How many "entities" have the same name?***

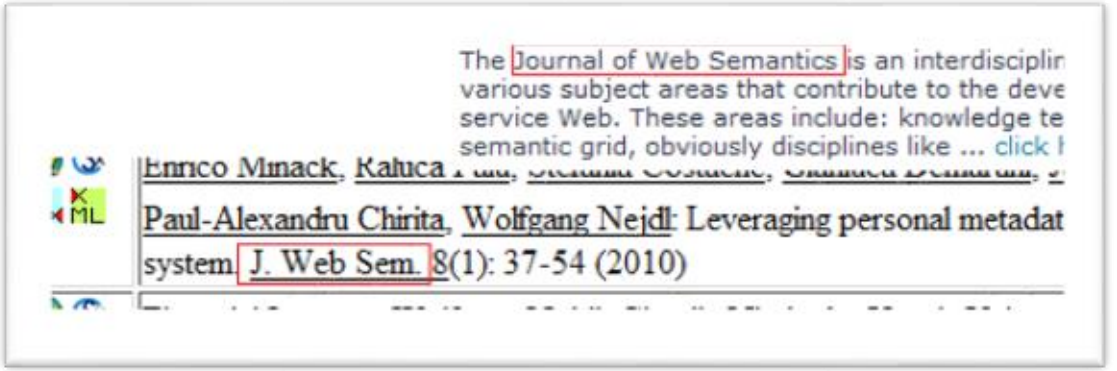
- London, KY
- London, Laurel, KY
- London, OH
- London, Madison, OH
- London, AR
- London, Pope, AR
- London, TX
- London, Kimble, TX
- London, MO
- London, MO
- London, London, MI
- London, London, Monroe, MI
- London, Uninc Conecuh County, AL
- London, Uninc Conecuh County, Conecuh, AL
- London, Uninc Shelby County, IN
- London, Uninc Shelby County, Shelby, IN
- London, Deerfield, WI
- London, Deerfield, Dane, WI
- London, Uninc Freeborn County, MN
- …

- London, Jack
  2612 Almes Dr
  Montgomery, AL
  (334) 272-7005
- London, Jack R
  2511 Winchester Rd
  Montgomery, AL 36106-3327
  (334) 272-7005
- London, Jack
  1222 Whitetail Trl
  Van Buren, AR 72956-7368
  (479) 474-4136
- London, Jack
  7400 Vista Del Mar Ave
  La Jolla, CA 92037-4954
  (858) 456-1850
- …

# Reasons of Different Descriptions

- Text variations:
  - Misspellings
  - Acronyms
  - Transformations
  - Abbreviations
  - etc.



# Reasons for Different Descriptions

- Text variations
- Local knowledge:
  - Each source uses different formats

    e.g., person from publication vs. person from email
  - Lack of global coordination for identifier assignment




**On-the-Fly Entity-Aware Query Processing in the Presence of Linkage**

Ekaterini Ioannou
L3S Research Center
Hannover, Germany
ioannou@L3S.de

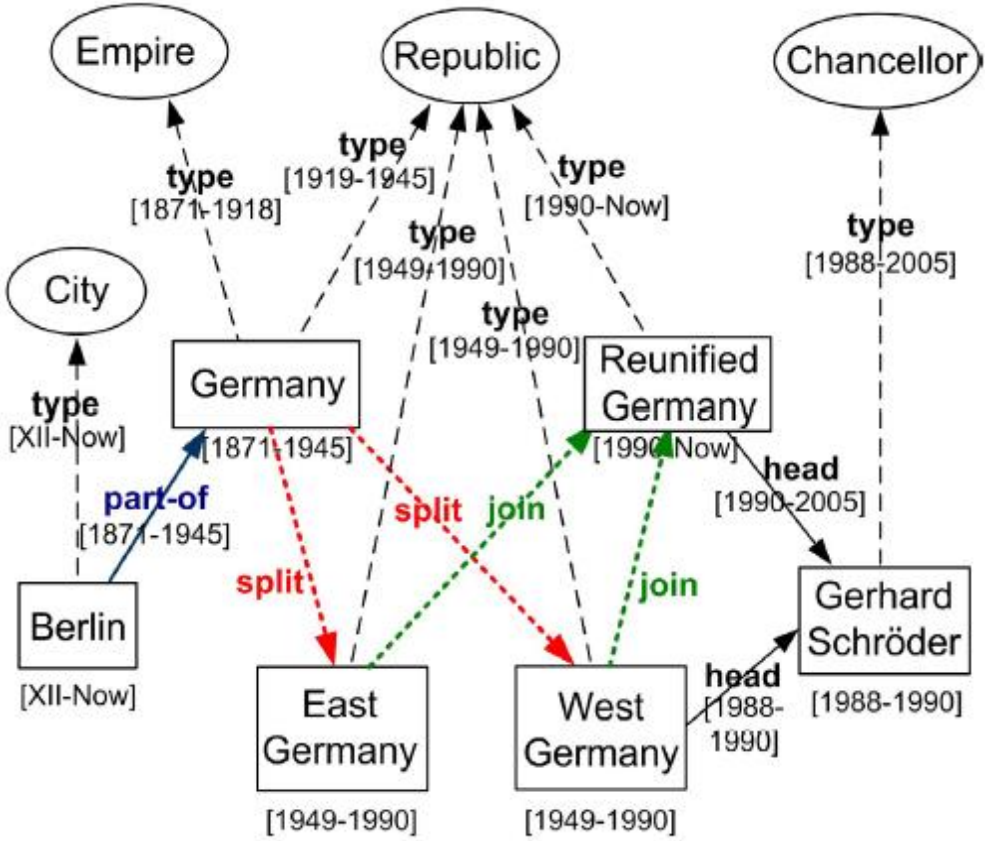
Wolfgang Nejdl
L3S Research Center
Hannover, Germany
nejdl@L3S.de

Claudia Niederée
L3S Research Center
Hannover, Germany
niederee@L3S.de

Yannis Velegrakis
University of Trento
Trento, Italy
velgias@disi.unitn.eu

# Reasons for Different Descriptions

- Text variations
- Local knowledge
- Evolving nature of data:
  - Entity alternative names
    - appearing in time
  - Updates in entity data




Jacqueline Lee Bouvier

[Vel09]

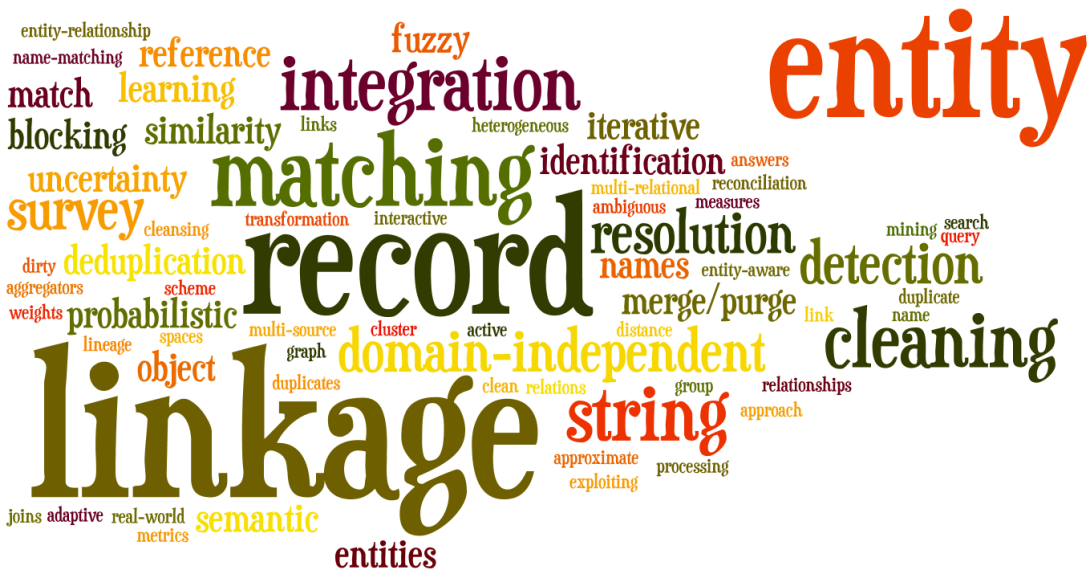
# Reasons for Different Descriptions

- Text variations
- Local knowledge
- Evolving nature of data
- New functionality:
  - Import data collections from various applications
    - e.g., Wikipedia data used in Freebase

# Entity Resolution

[Elm07] :

identify the different structures/records that model the same real-world object.



# Why it is useful

- Improves data quality and integrity
- Fosters re-use of existing data sources
- Optimize space

Application areas:

Linked Data, Social Networks, census data, price comparison portals

# Challenges for ER

- Variety – Semantic
  - Semi-structured data → unprecedented levels of heterogeneity
  - Numerous entity types & vocabularies
  - LOD (Linked Open Data) Cloud*: ~50,000 predicates, ~12,000 vocabularies

# Atomic similarity methods

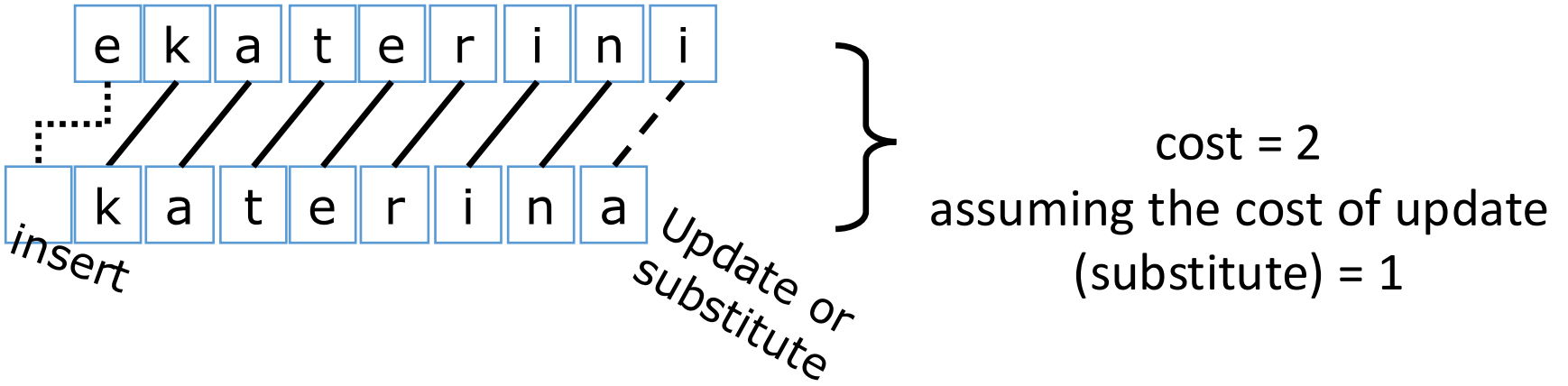
# Properties of the Distance Measures

- $d(x, y) \geq 0$ (distance is always positive)
- $d(x, y) = 0 \quad iff. \quad x = y$
- $d(x, y) = d(y, x)$ (Symmetry)
- $d(x, y) \leq d(x, z) + d(z, y)$ (the triangle in-equality)

# Atomic String Similarity – Edit Distance

- Number of operations to convert from 1st to 2nd string
- Operations in Levenstein distance [Lev66]

  → delete, insert, and update a character with cost 1





© E. Ioannou

# Computing Edit Distance – Another Example

- Example: compute the edit distance between intention and execution

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| I | N | T | E | * | N | T | I | O | N |
| \| | \| | \| | \| | \| | \| | \| | \| | \| | \| |
| * | E | X | E | C | U | T | I | O | N |
| d | s | s | | i | s | | | | |

- If each operation has cost of 1
  - Distance between these is 5
- If substitutions cost 2 (Levenshtein)
  - Distance between them is 8

# Computing Edit Distance (Cont.)

- Dynamic programming: A tabular computation of $D(n, m)$
- Solving problems by combining solutions to subproblems.
- Bottom up
  - We compute $D(i, j)$ for small $i, j$
  - And compute larger $D(i, j)$ based on previously computed smaller values
  - i.e., compute $D(i, j)$ for all $i$ $(0 < i < n)$ and $j$ $(0 < j < m)$

# Defining Minimum Edit Distance (Levenshtein)

- Initialization

$$D(i,0) = i$$

$$D(0,j) = j$$

- Recurrence Relation:

For each i = 1...M

For each j = 1...N

$$\text{D(i,j)} = \min \begin{cases} D(i-1,j) + 1 \\ \text{D(i,j}-1) + 1 \\ \text{D(i}-1,\text{j}-1) + \begin{cases} 2 & \text{if X(i)} \neq \text{Y(j)} \\ 0 & \text{if X(i)} = \text{Y(j)} \end{cases} \end{cases}$$

- Termination:

D(N,M) is distance

# Edit Distance Table – Example

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| N | 9 | | | | | | | | | |
| O | 8 | | | | | | | | | |
| I | 7 | | | | | | | | | |
| T | 6 | | | | | | | | | |
| N | 5 | | | | | | | | | |
| E | 4 | | | | | | | | | |
| T | 3 | | | | | | | | | |
| N | 2 | | | | | | | | | |
| I | 1 | | | | | | | | | |
| # | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | # | E | X | E | C | U | T | I | O | N |

# Edit Distance Table – Example (Cont.)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| N | 9 | | | | | | | | | |
| O | 8 | | | | | | | | | |
| I | 7 | | | | | | | | | |
| T | 6 | | | | | | | | | |
| N | 5 | | | | | | | | | |
| E | 4 | | | | | | | | | |
| T | 3 | | | | | | | | | |
| N | 2 | | | | | | | | | |
| I | 1 | | | | | | | | | |
| # | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | # | E | X | E | C | U | T | I | O | N |

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2 & \text{if } w1(i) \neq w2(j) \\ 0 & \text{if } w1(i) = w2(j) \end{cases} \end{cases}$$

# Edit Distance Table – Example (Cont.)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| N | 9 | 8 | 9 | 10 | 11 | 12 | 11 | 10 | 9 | **8** |
| O | 8 | 7 | 8 | 9 | 10 | 11 | 10 | 9 | **8** | 9 |
| I | 7 | 6 | 7 | 8 | 9 | 10 | 9 | **8** | 9 | 10 |
| T | 6 | 5 | 6 | 7 | 8 | 9 | **8** | 9 | 10 | 11 |
| N | 5 | 4 | 5 | 6 | 7 | **8** | 9 | 10 | 11 | 10 |
| E | 4 | 3 | 4 | **5** | **6** | 7 | 8 | 9 | 10 | 9 |
| T | 3 | 4 | **5** | 6 | 7 | 8 | 7 | 8 | 9 | 8 |
| N | 2 | **3** | 4 | 5 | 6 | 7 | 8 | 7 | 8 | 7 |
| I | **1** | 2 | 3 | 4 | 5 | 6 | 7 | 6 | 7 | 8 |
| # | **0** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | # | E | X | E | C | U | T | I | O | N |

# Another Way for Computing the Edit Distance

- $d(S_1, S_2) = |S_1| + |S_2| - (2 * |LCS(S_1, S_2)|$, where $LCS$ is the longest common substring
  - $LCS(S_1, S_2)$ can be obtained by deleting from $S_1$ and $S_2$
- Example:
  - $d(above, about) = 5 + 5 - 2 * 3 = 4, \quad LCS(above, about) = abo$
  - $d(execution, intention) = 9 + 9 - 5 = 8$
  - $LCS(execution, intention) = etion$

# Edit Distance is a Distance Measure

- $d(S_1, S_2) \geq 0$, there is no notion of negative edits

- $d(S_1, S_2) = 0 \iff S_1 = S_2$,
  - $S_1 \neq S_2 \Longrightarrow d(S_1, S_2) \neq 0$, since $S_1 \neq S_2$ there will be at least on edit
  - $d(S_1, S_1) = 0$, (obvious) no edits are needed

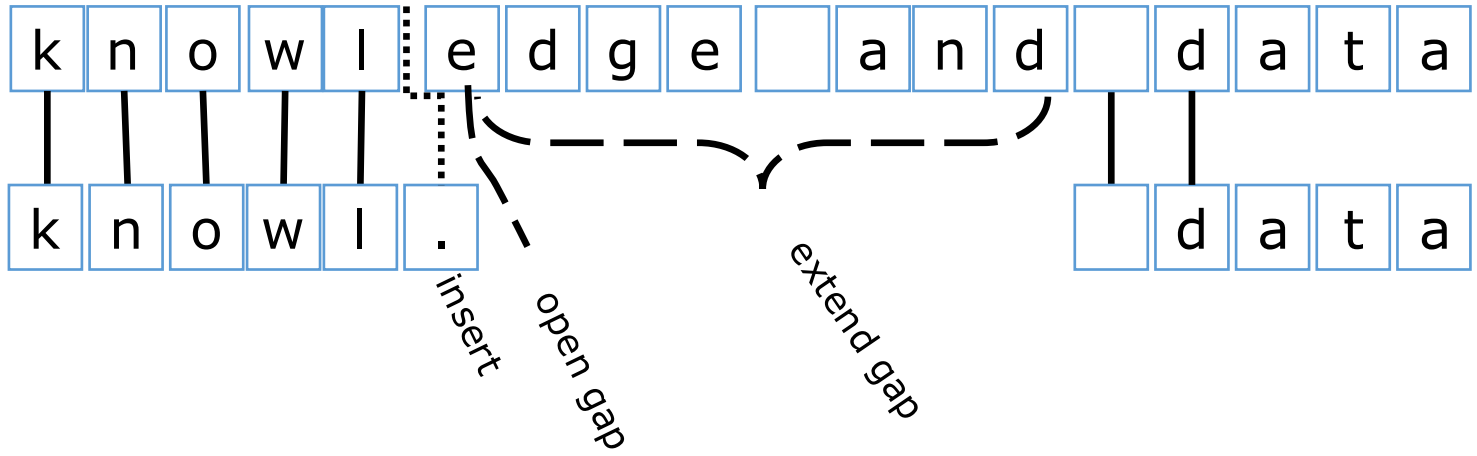
- $d(S_1, S_2) = d(S_2, S_1)$, insert/delete operations are inverses of each other

- $d(S_1, S_2) \leq d(S_1, S_3) + d(S_3, S_2)$, changing from $S_1$ to $S_3$ and from $S_3$ to $S_2$ is another way of changing $S_1$ to $S_2$

# Atomic String Similarity – Gap Distance

- Overcome limitation of edit distance with shortened strings
- Considers two extra operations

  → open gap, and extend gap (with small cost)

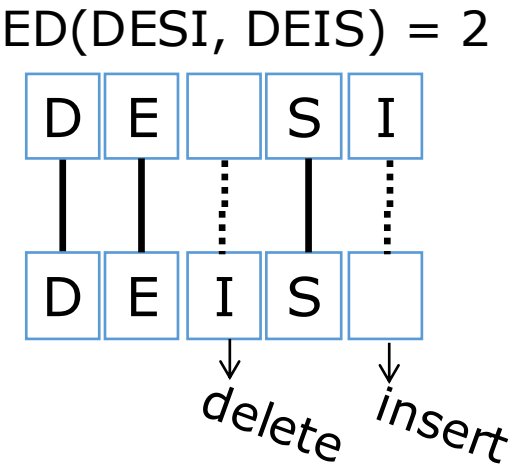



cost = 1 + o + 7e,

the open gap is not counted as part of the extended gap


© E. Ioannou

# Atomic String Similarity – Jaro Similarity




- Small strings, e.g., first and last names
- C is the set of common characters in $S_1$ and $S_2$
  - Two characters from $S_1$ and $S_2$ are considered *common* when they are the same and not farther than $\left\lfloor \frac{\max(|S_1|,|S_2|)}{2} \right\rfloor - 1$ characters apart.
- T is the number transpositions/2
  - $c_1$ and $c_2$ are a transposition if $c_1$ and $c_2$ are ***common*** but appear in different orders in $S_1$ and $S_2$

$$JaroSim(S_1, S_2) = \frac{1}{3}\left(\frac{C}{|S_1|} + \frac{C}{|S_2|} + \frac{C-T}{C}\right)$$ [Jar89]

- Example: "DEIS"vs. "DESI"

C=4, T=2/2, JaroSim= $\frac{1}{3}\left(\frac{4}{4} + \frac{4}{4} + \frac{4-1}{4}\right) = 0.9167$

# Atomic String Similarity – Jaro-Winkler Similarity

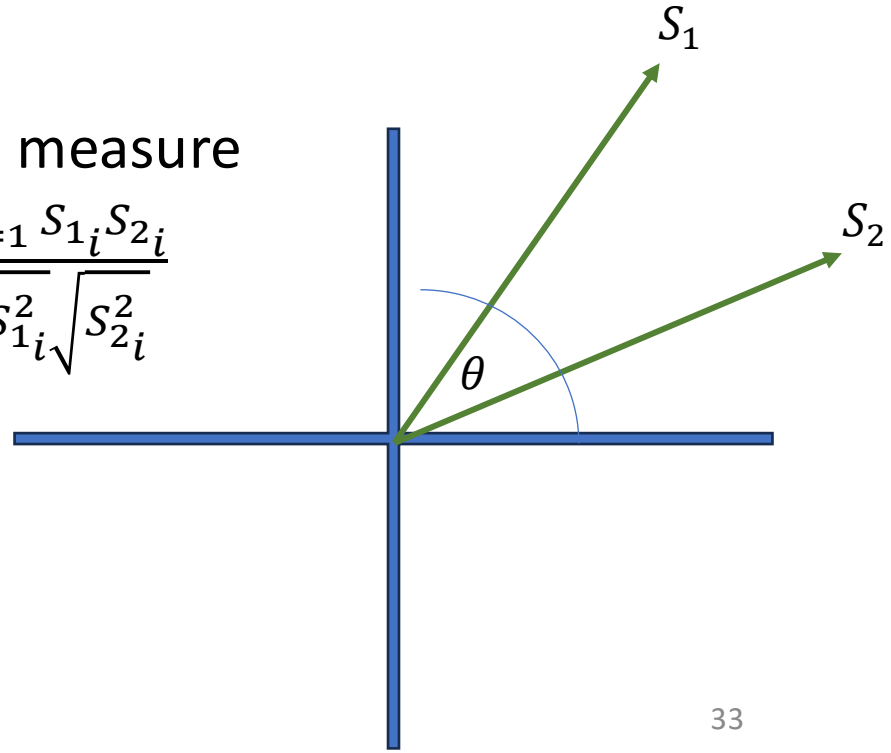
- Jaro-Winkler similarity [Win99]:
  - Extension that gives higher weight to matching prefix
  - Increasing its applicability to names
  - $J_w(S_1, S_2) = JaroSim + P * L * (1 - JaroSim)$
  - P is a scaling factor (0.1 by default)
  - L is the length of the common prefix up to maximum 4
  - Example: Compute $J_w(arnab, aranb)$
    - $JaroSim(arnab, aranb) = \frac{1}{3}\left(\frac{5}{5} + \frac{5}{5} + \frac{4}{5}\right) = 0.933$
    - $J_w(arnab, aranb) = 0.933 + 0.1 * 2 * (1 - 0.933) = 0.9466$

# Atomic String Similarity – Soundex

- Soundex: A phonetic algorithm that indexes names by their sounds when pronounced in English.
- Consists of the first letter of the name followed by three numbers. Numbers encode similar sounding consonants.
    - Remove all W, H
    - B, F, P, V encoded as 1, C,G,J,K,Q,S,X,Z as 2
    - D,T as 3, L as 4, M,N as 5, R as 6, Remove vowels
    - Concatenate first letter of string with first 3 numerals
- Ex: great and grate become G6EA3 and G6A3E and then G630
- More recent, metaphone, double metaphone etc.

# Atomic String Similarity – Cosine Similarity

- A string can be represented as a point in a high dimensional space
- The point can be represented as a vector from the origin $(0,0,\cdots,0)$ to its location
- Similar string have vectors pointing in the same direction
- Two vectors make an angle
  - The cosine of the angle can be used as a distance measure
  - $\cos\_sim(S_1, S_2) = cosine\ (\theta) = \frac{S_1 \cdot S_2}{\|S_1^2\|\|S_2^2\|} = \frac{\sum_{i=1}^{n} S_{1_i} S_{2_i}}{\sqrt{S_{1_i}^2}\sqrt{S_{2_i}^2}}$
  - $d(S_1, S_2) = arccos\left(\frac{\sum_{i=1}^{n} S_{1_i} S_{2_i}}{\sqrt{S_{1_i}^2}\sqrt{S_{2_i}^2}}\right)$



# Atomic String Similarity – Cosine Similarity

- Example:
  - $S_1 = (1,\ 2,\ -1),\ \ S_2 = (2,\ \ 1,\ \ 1)$
  - $\sum_{i=1}^{n} S_{1_i} S_{2_i} = 2 + 2 - 1 = 3$
  - $\sqrt{\sum_{i=1}^{n} S_{1_i}} = \sqrt{1 + 4 + 1} = \sqrt{6} = \sqrt{\sum_{i=1}^{n} S_{2_i}}$

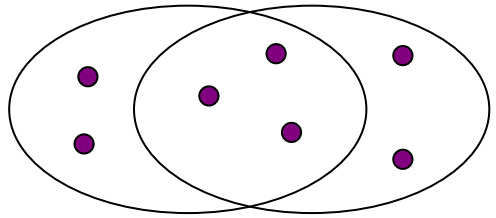
  - $\cos\_sim(S_1, S_2) = \frac{3}{\sqrt{6}\sqrt{6}} = 0.5$
  - $d(S_1, S_2) = \arccos(0.5) = 60°$ (distance in degrees)
  - $d(S_1, S_2) = 1 - \cos\_sim(S_1, S_2) = 0.5$

# Atomic String Similarity – Hamming Distance

- Hamming distance between two vectors is the number of positions in which the vectors' entries differ
- Example:
  - $v_1 = (1,\ 1,\ 2,\ 2,\ 3,\ 9,\ 6,\ 6,\ 8),\ \ v_2 = (1,\ 1,\ 2,\ 4,\ 3,\ 9,\ 5,\ 6,\ 8)$
  - $d(v_1, v_2) = 2$
  - $v_1, v_2$ are commonly used as bit-arrays

# Similarity methods for sets

# Similarity methods for sets



- **Jaccard similarity/distance**
  - The Jaccard similarity of two sets is:

$$J_{sim}(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

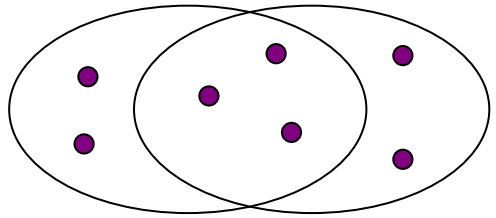
3 in intersection
7 in union
Jaccard similarity= 3/7
Jaccard distance = 4/7

Is $J_{dist}$ a distance measure?

  - Jaccard distance: $J_{dist} = 1 - J_{sim}(C_1, C_2) = 1 - \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$
- Similarity between {a, b, c, d} and {a, b, e, f} = 2/6 = 1/3

- Jaccard bag similarity counts the repetition of the elements
- The similarity between {a,a,a,b} and {a,a,b,b,c} = 3/9 = 1/3

# Similarity methods for sets



- **Sørensen Coefficient** (also called Coefficient of Community CC)
  - The Sørensen similarity of two sets is computed as:

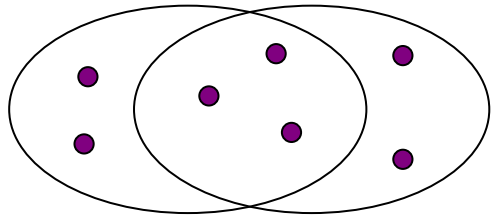
$$CC(C_1, C_2) = \frac{2*|C_1 \cap C_2|}{|C_1|+|C_2|}$$

3 in intersection
5 in each set
Sørensen similarity= 6/10

- Similarity between {a, b, c, d} and {a, b, e, f} = 4/8 = 1/2

- Gives more weight for the number of common elements

# Similarity methods for sets



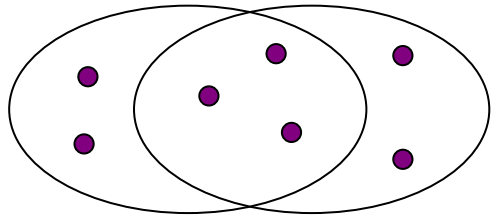
- **Tversky Index:** a generalized form of Jaccard and Sørensen
  - The Tversky Index of two sets is computed as:

$$S(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha|C_1 - C_2| + \beta|C_2 - C_1|}$$

  - $\alpha, \beta \geq 0$
  - $\alpha = \beta = 1 \implies$ Jaccard similarity
  - $\alpha = \beta = 0.5 \implies$ Sørensen similarity

3 in intersection
2 in the difference
$\alpha = 0.2$ & $\beta = 0.8$
Tversky similarity= 3/5

# Similarity methods for sets



- **Overlap Coefficient:** also called Szymkiewicz–Simpson coefficient
  - It is defined as:

3 in intersection
5 in each set
Overlap coefficient = 3/5

$$OC(C_1, C_2) = \frac{|C_1 \cap C_2|}{MIN(|C_1|,|C_2|)}$$

# Case of Documents

# A Common Metaphor

- Many problems can be expressed as finding "similar" sets
  - Find near-neighbors in high-dimensional space
- Examples:
  - Pages with similar words
    - For duplicate detection, classification by topic, plagiarism
  - Customers who purchased similar products (e.g. Movies)
  - Products with similar customer sets (e.g. fans)
  - Images with similar features
    - Users who visited similar websites


© J. Ullman et al.

# Shingles

- A *k*-shingle (or *k*-gram) for a document is a sequence of *k* tokens that appears in the doc
  - Tokens can be characters, words or something else, depending on the application
  - Assume tokens = characters for examples

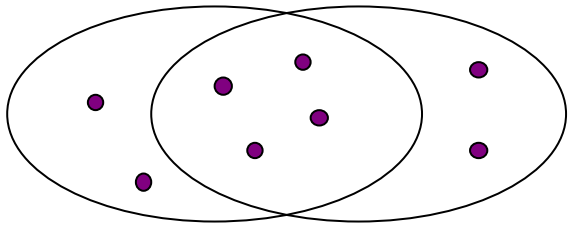
- **Example: k=2**; document $\mathbf{D_1}$ = abcab
  Set of 2-shingles: $\mathbf{S(D_1)}$ = {ab, bc, ca}
  - **Option:** Shingles as a bag (multiset), count ab twice: $\mathbf{S'(D_1)}$ = {ab, bc, ca, ab}


© J. Ullman et al.

# Similarity Metric for Shingles

- **Represent document $D_1$ as a set of its k-shingles $C_1=S(D_1)$**
- Equivalently, each document is a 0/1 vector in the space of *k*-shingles
  - Each unique shingle is a dimension
  - Vectors are very sparse
- **A natural similarity measure is the Jaccard similarity:**

$$J_{sim}(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$




© J. Ullman et al.

# Example

- **Rows** = elements (shingles)
- **Columns** = sets (documents)
  - 1 in row ***e*** and column ***s*** if and only if ***e*** is a member of ***s***
  - Column similarity is the Jaccard similarity of the corresponding sets (rows with value *1)*
  - **Typical matrix is sparse!**
- **Data**
  - **D1 = {jumped}**
  - **D2 = {bumped}**
  - **D3 = {dumped}**
  - **D4 = {pumped}**

Documents

Shingles

| | D1 | D2 | D3 | D4 |
|---|---|---|---|---|
| bum | 0 | 1 | 0 | 0 |
| dum | 0 | 0 | 1 | 0 |
| jum | 1 | 0 | 0 | 0 |
| mpe | 1 | 1 | 1 | 1 |
| ped | 1 | 1 | 1 | 1 |
| pum | 0 | 0 | 0 | 1 |
| ump | 1 | 1 | 1 | 1 |


© J. Ullman et al.

# Example

- **Each document is a column:**
  - $J_{sim}(D_1, D_2) = ?$
  - $|D_1 \cap D_2| = 3$
  - $|D_1 \cup D_2| = 5$
  - Jaccard similarity (not distance) = $J_{sim}(D_1, D_2) = \frac{3}{5}$
  - $d(D_1, D_2) = 1 - J_{sim}(D_1, D_2) = \frac{2}{5}$

Documents

Shingles

| | D1 | D2 | D3 | D4 |
|---|---|---|---|---|
| bum | 0 | 1 | 0 | 0 |
| dum | 0 | 0 | 1 | 0 |
| jum | 1 | 0 | 0 | 0 |
| mpe | 1 | 1 | 1 | 1 |
| ped | 1 | 1 | 1 | 1 |
| pum | 0 | 0 | 0 | 1 |
| ump | 1 | 1 | 1 | 1 |


© J. Ullman et al.

# Challenges for ER

- Variety – Semantic
  - Semi-structured data → unprecedented levels of heterogeneity
  - Numerous entity types & vocabularies
  - LOD Cloud*: ~50,000 predicates, ~12,000 vocabularies
- Volume - Performance
  - Millions of entities
  - Billions of name-value pairs describing them
  - LOD Cloud*: >$5{,}5 \cdot 10^7$ entities, ~$1{,}5 \cdot 10^{11}$ triples
  - **Too many documents, Too few memory**


© J. Ullman et al.

# Motivation

- **Suppose we need to find near-duplicate documents among $N = 1$ million documents**

- Naïvely, we would have to compute **pairwise Jaccard similarities** for **every pair of docs**
  - $N(N-1)/2 \approx 5*10^{11}$ comparisons
  - At $10^5$ secs/day and $10^6$ comparisons/sec, it would take **5 days**

- For $N = 10$ million, it takes more than a year…


© J. Ullman et al.

# Find Pairs of Similar Documents

- **Main idea: Candidates**
  - Instead of keeping a count of each pair, only keep a count of candidate pairs!
- **Pass 1:** Take documents and hash them to buckets such that documents that are similar hash to the same bucket
- **Pass 2:** Only compare documents that are **candidates** (i.e., they hashed to a same bucket)
- **Benefits: Instead of $O(N^2)$ comparisons, we need $O(N)$ comparisons to find similar documents**


© J. Ullman et al.

How could we use hashing to convert a document to a Sparse Boolean Vector (where each index represents a different word)?

# Finding similar documents requires more than simple hashing functions


© J. Ullman et al.

# 3 Essential Steps for Similar Docs

1. ***Shingling:*** Convert documents to sets

2. ***Min-Hashing:*** Convert large sets to short signatures, while preserving similarity

3. ***Locality-Sensitive Hashing:*** Focus on pairs of signatures likely to be from similar documents
   - **Candidate pairs!**


© J. Ullman et al.

# Additional References


- [Jar89] M. A. Jaro: Advances in record linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association 84: 414-420.
- [Win99] William E. Winkler: The state of record linkage and current research problems. IRS publication R99/04 (http://www.census.gov/srd/www/byname.html)
- Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. Journal of the American Statistical Association, 64(328):1183–1210.
- [Lev66] Levenshtein, Vladimir I. (February 1966). "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady. 10 (8): pp. 707–710.
- [Vel09] Rizzolo, F., Velegrakis, Y., Mylopoulos, J., Bykau, S. (2009). Modeling Concept Evolution: A Historical Perspective. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds) Conceptual Modeling - ER 2009. ER 2009. Lecture Notes in Computer Science, vol 5829. Springer, Berlin, Heidelberg.
- [Elm07] A. K. Elmagarmid, P. G. Ipeirotis and V. S. Verykios, "Duplicate Record Detection: A Survey," in IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 1, pp. 1-16, Jan. 2007.
- [Don15]

# Thank you for your attention!



# Questions?

Disclaimer: Much of the material presented originates from a number of different presentations and courses of the following people: Yannis Velegrakis (Utrecht University), Jeff Ullman (Stanford University), Bill Howe (U of Washington), Martin Fouler (Thought Works), Ekaterini Ioannou (Tilburg University), Themis Palpanas (U of Paris-Descartes). Copyright stays with the authors. No distribution is allowed without prior permission by the authors.




- Summarize what you learned today in 2-minutes

# DISCLAIMER

The information in this presentation has been compiled with the utmost care,
but no rights can be derived from its contents.

© Utrecht University