

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN

KHAI PHÁ DỮ LIỆU - CO3029 - HK222

CHỦ ĐỀ 2: DATA MATCHING

Giảng viên hướng dẫn: PGS.TS Lê Hồng Trang
Lớp: L01
Nhóm: 1
Sinh viên thực hiện: Nguyễn Đức An – 2010102
Luu Hoàng Thu Hà – 2010236
Tô Thanh Phong – 1914637
Trần Thị Thu Thảo – 2010629
Bùi Khánh Vĩnh – 2010091

Mục lục

1	Giới thiệu vấn đề	2
1.1	Giới thiệu data integration	2
1.2	Khái niệm data matching	4
2	Phương pháp tiếp cận	4
2.1	Rule-based matching	4
2.2	Learning-based matching	7
2.3	Matching by clustering	7
2.4	Probabilistic Approaches	8
3	Giới thiệu về tập dữ liệu	9
4	JedAI Toolkit	10
4.1	Giới thiệu JedAI Toolkit	10
4.2	Workflow	11
4.3	Data Reading	11
4.4	Schema Clustering	12
4.5	Block Building	12
4.6	Block Cleaning	13
4.7	Comparison Cleaning	13
4.8	Entity Matching	15
4.9	Entity Clustering	16
4.10	Similarity Join	17
4.11	Comparison Prioritization	19
5	Kết quả và đánh giá	21
5.1	Hướng dẫn chạy source code	21
5.2	Hướng dẫn load dữ liệu	22
5.3	Kết quả	25
5.3.1	Block-based workflow	25
5.3.2	Join-based workflow	27
5.3.3	Progressive workflow	28
5.4	Đánh giá	29
6	Kết luận	32
7	Tài liệu tham khảo	33

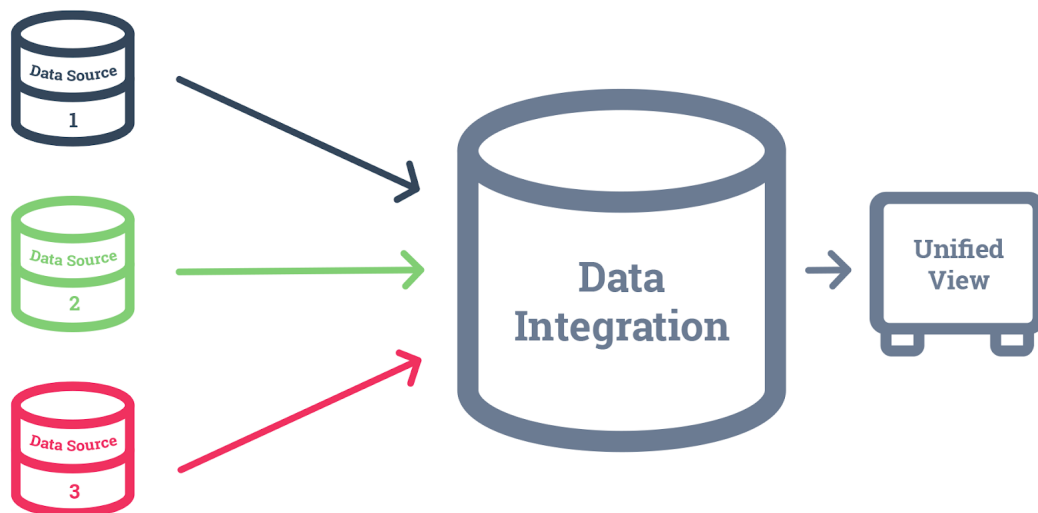
1 Giới thiệu vấn đề

Theo sự phát triển của thời gian, lượng dữ liệu sinh ra ngày một nhiều và nằm rải rác trên các nguồn dữ liệu khác nhau và phân tán ở nhiều nơi. Tương ứng với mỗi nguồn thì dữ liệu sẽ tồn tại dưới nhiều định dạng, mô hình khác nhau và rất đa dạng về ngữ nghĩa. Điều đó gây khó khăn khi chúng ta cần truy vấn các thông tin cần thiết. Bên cạnh đó, nhu cầu chia sẻ dữ liệu để thực hiện các công việc như thống kê, phân tích, dự đoán,... thường xuyên diễn ra. Để đảm bảo quá trình trao đổi dữ liệu diễn ra hiệu quả và khai thác tốt được giá trị hữu ích từ dữ liệu thu thập được thì tích hợp dữ liệu (data integration) là một bước không thể thiếu trong quá trình xử lý dữ liệu.

Như đã đề cập thì do dữ liệu sẽ được phân tán ở nhiều nơi nên quá trình tích hợp cũng sẽ gặp một số khó khăn nhất định như là sự không đồng nhất về định dạng dữ liệu ở các nguồn, ở mỗi nguồn dữ liệu có cùng 1 thuộc tính nhưng lại tồn tại nhiều tên khác nhau, các thuộc tính khác nhau đôi khi lại trùng tên, mô hình ở các nguồn rất đa dạng, chất lượng dữ liệu kém, quá trình thu thập và xử lý dữ liệu thời gian thực phải kịp thời,... Bên cạnh đó, thì dữ liệu ở các nơi khác nhau hoặc thậm chí ở cùng một nguồn cũng có thể bị trùng lặp gây nên sự tổn kém về chi phí lưu trữ, xử lý,.. Kết hợp dữ liệu (data matching) sẽ giúp tìm kiếm các dữ liệu trùng lặp này và giải quyết được một vấn đề mà quá trình tích hợp dữ liệu đang gặp phải.

1.1 Giới thiệu data integration

Tích hợp dữ liệu (data integration): là quá trình kết hợp dữ liệu lại với nhau từ nhiều nguồn khác nhau thành dữ liệu nhất quán để dễ sử dụng, xử lý và quản lý hiệu quả.



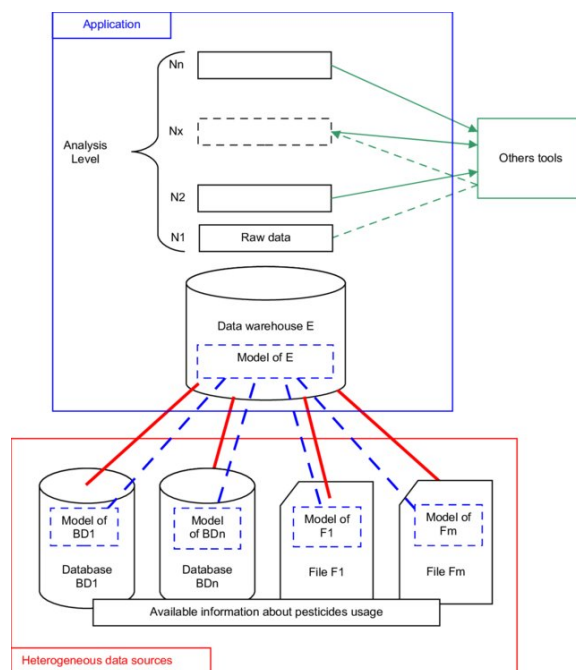
Một số ứng dụng phổ biến của tích hợp dữ liệu:

- Trong kinh doanh, thông thường, các doanh nghiệp lớn nhỏ sẽ sử dụng nhiều hệ thống khác nhau để điều hành hoạt động của mình. Vì vậy nên tích hợp dữ liệu có thể bao gồm tích hợp thông tin người dùng, dữ liệu về doanh số bán hàng, tiếp thị, kế toán,... Từ kết quả của tích hợp dữ liệu, các doanh nghiệp có thể dễ dàng phân tích và dự đoán thị hiếu của khách hàng, đạt được lợi thế cạnh tranh. Ngoài ra còn giúp doanh nghiệp tiết kiệm chi phí nhân lực và bảo trì do chỉ cần quản lý dữ liệu trên một nền tảng duy nhất.

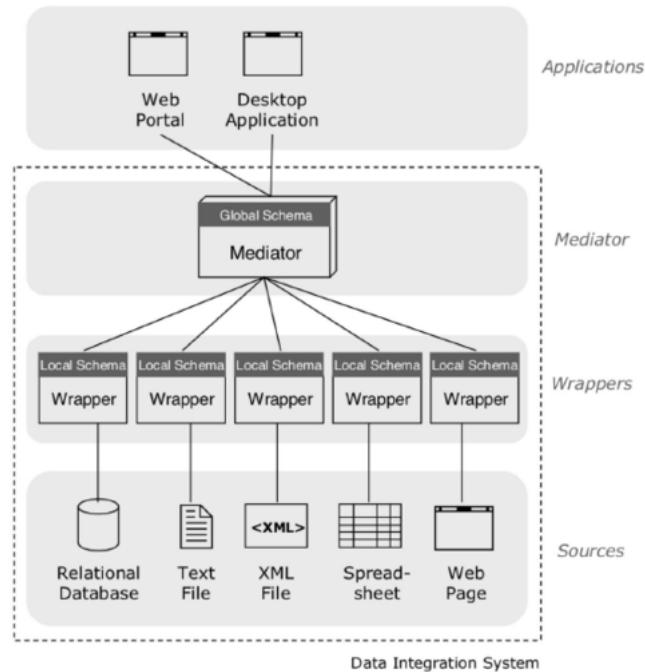
- Đặc biệt trong lĩnh vực khoa học y tế, hàng trăm nguồn dữ liệu y sinh đang có sẵn và đang phát triển rất nhanh chóng. Trong ngành chăm sóc sức khỏe, dữ liệu tích hợp từ bệnh nhân và phòng khám khác nhau giúp bác sĩ có thể xác định được các bệnh hoặc rối loạn y tế thông qua tiền sử và các triệu chứng của bệnh nhân, từ đó tăng chất lượng chăm sóc và điều trị. Việc thu thập và tích hợp dữ liệu hiệu quả cũng cải thiện độ chính xác của quá trình xử lý yêu cầu bảo hiểm y tế và đảm bảo rằng tên bệnh nhân và thông tin liên hệ được ghi lại một cách nhất quán và chính xác.
- Tích hợp dữ liệu cũng được sử dụng trên các website thông qua các form điền dữ liệu để thu thập thông tin nhằm tăng trải nghiệm của khách hàng.

Có hai kỹ thuật tích hợp dữ liệu phổ biến:

- **Data warehousing:** cách này sẽ sử dụng một vùng lưu trữ chung (gọi là kho dữ liệu), để làm sạch, định dạng và lưu trữ. Quá trình này thường sử dụng những công cụ ETL để lấy dữ liệu. Dữ liệu từ tất cả các ứng dụng khác nhau trong toàn tổ chức được sao chép vào kho dữ liệu, nơi các nhà phân tích dữ liệu có thể truy vấn dữ liệu đó. Việc truy vấn dữ liệu sẽ được xảy ra ở trong kho dữ liệu nên không cần lo lắng sẽ ảnh hưởng đến hiệu suất làm việc của các ứng dụng trong tổ chức. Tuy nhiên kỹ thuật này sẽ gây tốn chi phí lưu trữ do phải lưu dữ liệu ở nhiều vị trí và chi phí tạo, bảo trì kho dữ liệu.



- **Virtual data integration:** vẫn lưu trữ dữ liệu tại các nguồn riêng biệt và truy cập chúng khi thực hiện truy vấn. Phương pháp này được cho là phù hợp với những xu hướng tích hợp dữ liệu trong tương lai.



1.2 Khái niệm data matching

Data matching là quá trình so sánh dữ liệu để tìm kiếm các dữ liệu cùng chỉ đến một thực thể trong thế giới thực. Các dữ liệu này có thể đến cùng một bảng, nhiều bảng khác nhau hoặc thậm chí chúng không phải là những dữ liệu quan hệ (XML hoặc RDF). Về lý thuyết thì data matching có thể xem như là string matching nếu như chúng ta nối thông tin của các trường dữ liệu lại với nhau thành một chuỗi, tuy nhiên cách này có thể gây khó khăn khi áp dụng một số kỹ thuật phức tạp và mang lại kết quả không mong đợi.

Như đã đề cập ở trên, việc trùng lặp dữ liệu là một thách thức đối với quá trình tích hợp dữ liệu có thể xảy ra do một số lý do không mong muốn như các lỗi chính tả, sự không đồng nhất trong cách biểu diễn dữ liệu của các bộ phận trong một tổ chức, các biến thể, ... Data Matching sẽ giúp loại bỏ được các dữ liệu trùng lặp, làm cho dữ liệu được sạch và chính xác hơn, tiết kiệm chi phí cho việc lưu trữ, phân tích, và chi phí thực thi một chiến lược nhiều lần cho cùng một thực thể, ...

Một số cách tiếp cận phổ biến của data matching là : dựa trên quy tắc(rule-based), dựa trên học tập (learning-based), phân cụm, dựa trên xác suất, ...

2 Phương pháp tiếp cận

2.1 Rule-based matching

Đối với phương pháp này, các nhà phát triển xây dựng các luật để kiểm tra xem các hai tuple có match với nhau hay không dựa trên các phương pháp chính sau:

- Linearly Weighted Combination Rules
- Logistic Regression Rules

1. Linearly Weighted Combination Rules:

a. Nội dung

- Ý tưởng chính của phương pháp này là chúng ta sẽ tính toán **sim score** giữa các tuple x và y dựa vào tất cả các thuộc tính tương đồng của các tuples theo mô hình tuyến tính bằng công thức sau:

$$sim(x, y) = \sum_{i=1}^N \alpha_i * sim_i(x, y)$$

- Trong đó:
 - N là số attributes của các tuple.
 - $sim_i(x, y)$ là sim score của thuộc tính thứ i giữa tuple x và y.
 - α_i là một trọng số (weight) được xác định trước trong khoảng [0,1] tương ứng với thuộc tính thứ i trong tuple, nhằm đánh giá tầm ảnh hưởng của thuộc tính đó đối với giá trị sim(x,y) giữa các tuple x và y.
- Chúng ta sẽ quy ước một luật sao cho với một hằng số β cho trước, hai tuple x và y được gọi là match với nhau nếu $sim(x, y) \geq \beta$ và not-match nếu $sim(x, y) \leq \gamma$ hoặc tùy vào thuộc tính mà người lập trình chọn để đưa vào mô hình.

b. Ví dụ

Table X					Table Y					Matches	
	Name	Phone	City	State		Name	Phone	City	State		
x_1	Dave Smith	(608) 395 9462	Madison	WI	y_1	David D. Smith	395 9426	Madison	WI		(x_1, y_1)
x_2	Joe Wilson	(408) 123 4265	San Jose	CA	y_2	Daniel W. Smith	256 1212	Madison	WI		(x_2, y_2)
x_3	Dan Smith	(608) 256 1212	Middleton	WI							

(a)

(b)

(c)

- $sim(x, y) = 0.3s_{name}(x, y) + 0.3s_{phone}(x, y) + 0.1s_{city}(x, y) + 0.3s_{state}(x, y)$
 - $s_{name}(x, y)$: based on Jaro-Winkler
 - $s_{phone}(x, y)$: based on edit distance between x's phone (after removing area code) and y's phone
 - $s_{city}(x, y)$: based on edit distance
 - $s_{state}(x, y)$: based on exact match; yes $\rightarrow 1$, no $\rightarrow 0$

c. Đánh giá

- Ưu điểm:**
 - Đơn giản, dễ hiện thực
 - Có thể cải thiện hằng số α trong quá trình training tập dữ liệu.
- Nhược điểm:**
 - Dễ xảy ra hiện tượng **diminishing return**.
 - Giả sử nếu $s_{name}(x, y)$ có trọng số 0.95 đối với công thức trên, thì quá trình tiếp tục tăng giá trị $s_{name}(x, y)$ sẽ được gọi là contribute minimally, bởi vì mặc dù trường name có tỉ lệ thuận ảnh hưởng cao đến $s(x, y)$ nhưng khi tiếp tục tăng đến một mức nào đó thì $s(x, y)$ sẽ bắt đầu giảm, hiện tượng này được gọi là diminishing return.

2. Logistic Regression Rules

a. Nội dung

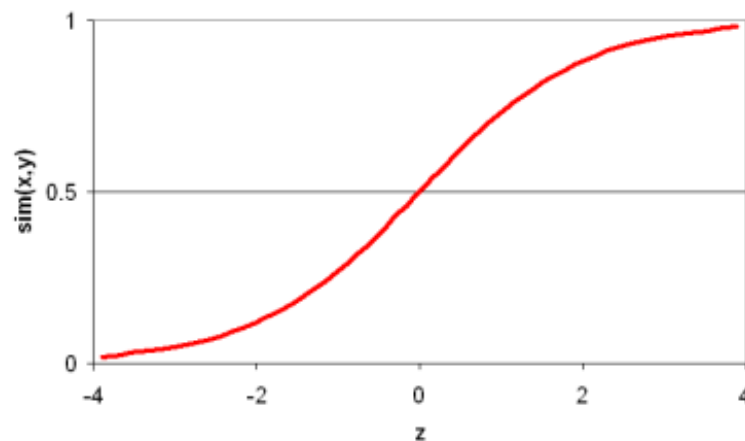
- Được sử dụng để giải quyết vấn đề về diminishing return mà phương pháp linearly weighted combination rules chưa làm được.
- Lúc này sim score được tính toán theo công thức sau:

$$\text{sim}(x, y) = \frac{1}{1 + e^{-z}}$$

$$z = \sum_{i=1}^N \alpha_i * \text{sim}_i(x, y)$$

- Lưu ý:

- α_i không còn bị ràng buộc trong khoảng $[0, 1]$ và tổng hệ số bằng 1.
- z có giá trị trong khoảng từ $(-\infty, +\infty)$ và sự gia tăng của bất cứ giá trị $\text{sim}_i(x, y)$ sẽ không gây ảnh hưởng đột biến đến giá trị $\text{sim}(x, y)$.



b. Đánh giá

- Phù hợp để đánh giá tính similarity giữa các tuple dựa trên nhiều trường thuộc tính khác nhau.
- Không cần phải đánh giá mô hình bằng tất cả các trường như mô hình tuyến tính mà có thể chọn ra những thuộc tính quan trọng để tăng độ chính xác của mô hình.

3. Đánh giá chung

• Ưu điểm

- Đơn giản, dễ hiện thực và debug.
- Tốc độ chạy nhanh.
- Có thể encode được các complex matching knowledge.

• Nhược điểm

- Tốn nhiều thời gian để xây dựng các rules đủ tốt cho mô hình.
- Khó khăn trong việc xác định đúng trọng số cho các trường thuộc tính.

2.2 Learning-based matching

Ở đây, chúng ta xét đến việc học tập có giám sát (supervised learning). Quá trình này bao gồm việc huấn luyện một mô hình M (matching model) từ tập dữ liệu huấn luyện, sau đó áp dụng mô hình này để so trùng trên các cặp tuples mới. Cụ thể từng bước như sau.

1. Huấn luyện một mô hình M (matching model)

- Giả sử dữ liệu dùng để huấn luyện là tập $T = \{(x_1, y_1, l_1), \dots, (x_n, y_n, l_n)\}$, trong đó, mỗi cặp (x_i, y_i) là một cặp tuple và l_i là nhãn: "có" nếu x_i trùng với y_i và "không" nếu ngược lại.
- Xác định một tập hợp các đặc trưng f_1, \dots, f_m , trong đó, mỗi đặc trưng dùng để định lượng một khía cạnh của miền được đánh giá là có thể liên quan đến việc so trùng các tuples.
- Chuyển đổi từng mẫu dữ liệu (x_i, y_i, l_i) trong tập dữ liệu huấn luyện T thành một cặp $(\langle f_1(x_i, y_i), \dots, f_m(x_i, y_i) \rangle, c_i)$, trong đó:
 - $v_i = \langle f_1(x_i, y_i), \dots, f_m(x_i, y_i) \rangle$ là một vectơ đặc trưng mã hóa (x_i, y_i) dưới dạng các đặc trưng.
 - c_i là một cách biểu diễn của nhãn l_i . Ví dụ: nhãn "có"/"không" được biểu diễn thành 1/0. Cách biểu diễn phụ thuộc vào phương thức mã hoá nhãn (label encoding).
- Từ đó, tập dữ liệu huấn luyện T ban đầu được biến đổi thành tập $T' = \{(v_1, c_1), \dots, (v_n, c_n)\}$
- Sau đó, sử dụng các thuật toán học máy (ví dụ: decision trees, SVMs) trên tập dữ liệu huấn luyện T' để huấn luyện mô hình matching M .

2. Áp dụng mô hình M để so trùng trên các cặp tuples mới

- Với một điểm dữ liệu mới (chưa biết nhãn) (x, y) thì ta sẽ biến đổi thành vector đặc trưng $v = \langle f_1(x, y), \dots, f_m(x, y) \rangle$
- Sau đó, áp dụng mô hình M để dự đoán x có trùng khớp với y hay không.

Ưu điểm và nhược điểm của hướng tiếp cận Learning-based:

• Ưu điểm

- Hướng tiếp cận Learning-based giúp tự động tổng hợp thông tin từ một số lượng lớn các đặc trưng.
- Hướng tiếp cận Learning-based có thể học được những quy tắc (rules) rất phức tạp.

• Nhược điểm

- Yêu cầu các mẫu dữ liệu huấn luyện, trong nhiều trường hợp là một số lượng lớn, điều này đôi khi khó có thể đạt được.

2.3 Matching by clustering

Clustering là một kỹ thuật được sử dụng trong học máy để nhóm các điểm dữ liệu tương tự lại với nhau. Đó là một kiểu học tập không giám sát trong đó thuật toán cố gắng tìm các mẫu trong dữ liệu mà không có bất kỳ kiến thức nào trước đó về những gì nó đang tìm kiếm. Các thuật toán phân cụm hoạt động bằng cách tính toán sự giống nhau giữa tất cả các cặp ví dụ. Điều này có nghĩa là thời gian chạy của chúng tăng theo bình phương của số ví dụ n , được ký hiệu là $O(n^2)$ trong ký hiệu độ phức tạp.

Nhưng $O(n^2)$ lại là thuật toán không có ứng dụng thực tiễn cao vì sẽ tốn rất nhiều thời gian nếu dữ liệu đạt tới mức hàng triệu. Vì thế các nhà nghiên cứu luôn cố gắng để sáng tạo và tìm ra các giải thuật có độ phức tạp nhỏ hơn nhưng vẫn cho kết quả đủ tốt. Một số ví dụ về các thuật toán phân cụm nổi tiếng là

- K-means clustering
- Hierarchical clustering
- Density-based clustering
- Distribution-based clustering
- Subspace clustering

Nói chung, việc lựa chọn một phương pháp phân cụm tương tự tốt (cluster similarity method) phụ thuộc vào ứng dụng và yêu cầu xem xét cẩn thận từ người xây dựng ứng dụng. Trước khi tiếp tục các cách tiếp cận khác, điều quan trọng là phải nhấn mạnh các quan điểm mới mà cách tiếp cận phân cụm đưa ra cho vấn đề đối sánh dữ liệu:

1. Xem các bộ phù hợp là vấn đề xây dựng các thực thể (nghĩa là các cụm), với sự hiểu biết rằng chỉ các bộ dữ liệu trong một cụm khớp.
2. Quá trình này lặp lại: trong mỗi lần lặp lại, chúng tôi tận dụng những gì chúng tôi đã biết cho đến nay (trong các lần lặp lại trước đó) để xây dựng các thực thể “tốt hơn”.
3. Trong mỗi lần lặp, chúng ta cố gắng “merge” tất cả các bộ phù hợp trong mỗi cụm để xây dựng một “entity profile”, sau đó sử dụng entity profile này để so khớp các bộ dữ liệu khác. Điều này được thấy rõ nhất trong trường hợp tạo ra một bộ dữ liệu chính tắc, có thể được xem như một entity profile. Bằng cách như vậy, như vậy, phân cụm giới thiệu khía cạnh mới của việc hợp nhất sau đó khai thác thông tin để giúp kết hợp. Chúng ta sẽ thấy các nguyên tắc tương tự cũng xuất hiện trong các cách tiếp cận khác.

2.4 Probabilistic Approaches

Probabilistic Approaches là hướng tiếp cận theo cách thống kê dùng để giải quyết bài toán nhận dạng dựa trên xác suất và hoán vị để xác định được độ khớp của các tập dữ liệu với nhau. Phương pháp này có độ chính xác cao, thường được dùng cho các tập dữ liệu lớn và phức tạp.

Một trong những lợi ích của hướng tiếp cận này là có thể xử lý cho nhiều loại dữ liệu (có thể có cấu trúc hoặc không), phù hợp trong việc giải quyết các bài toán với những tập dữ liệu có các phần không đồng nhất với nhau. Càng nhiều dữ liệu được xử lý thì phương pháp này càng có thể nâng cao độ chính xác cho kết quả.

Trong lý thuyết xác suất, hai sự kiện được gọi là độc lập nếu việc A xảy ra không ảnh hưởng đến xác suất của B xảy ra. Xác suất của hai sự kiện độc lập A và B bằng tích của xác suất A và xác suất B. Các phương pháp theo hướng tiếp cận bằng xác suất ta giả sử khi matching 2 record, thì việc matching này hoạt động dựa trên việc so sánh các trường hoặc biến số của mỗi bản ghi và tính toán xác suất (bằng tích xác suất của các trường tương ứng phù hợp) và so sánh nó với ngưỡng (threshold) được đặt ra để đưa đến nhận định là khớp (match), không khớp (unmatch) hoặc là có khả năng khớp. Điều này bao

gồm việc tính toán xác suất cho mỗi trường hoặc biến số tham gia vào việc kiểm tra, cũng như xác suất của bất kỳ sai khác hoặc lỗi nào trong dữ liệu.

Trong Probabilistic Matching, điểm đánh giá của một cặp bản ghi dựa trên xác suất ước tính rằng một cặp bản ghi đại diện cho cùng một thực thể, phương pháp này dựa trên việc tính toán các trọng số về việc match và unmatched của các trường hay các biến:

- $P(\text{agree} | \text{true match})$ hay m-probability (M): xác suất rằng hai đối tượng là một khi mà 2 trường của nó giống nhau, xác suất này giống nhau khi áp dụng với tất cả các trường cho tất cả các record
- $P(\text{agree} | \text{no match})$ hay u-probability (U): xác suất rằng hai đối tượng là một khi mà 2 trường của nó không giống nhau, xác suất này khác nhau giữa các trường.

Để đơn giản hóa tính toán, chúng ta sử dụng logarith để tính thay vì dùng xác suất trực tiếp. Với mỗi trường hay biến với các trọng số match và unmatched ta tính được các trọng số về việc ước lượng (estimating weight):

- Agreement weight:

$$\log \left(\frac{M}{U} \right)$$

- Disagreement weight:

$$\log \left(\frac{1 - M}{1 - U} \right)$$

Để đánh giá về 2 record thì ta sẽ cộng các trọng số này lại theo tính chất match hay unmatched của các trường tương ứng.

Một trong những phương pháp thông dụng trong Data Matching theo hướng tiếp cận bằng xác suất là mô hình Fellegi-Sunter, khi các trọng số hoặc xác suất được gán cho các thuộc tính khác nhau dựa trên sự quan trọng và đáng tin cậy của chúng trong việc xác định kết quả phù hợp. Sau đó mô hình tính toán tỷ lệ xác suất cho mỗi cặp bản ghi, được sử dụng để phân loại chúng là phù hợp hay không phù hợp.

Tuy nhiên hướng tiếp cận theo xác suất thì đòi hỏi nhiều trong việc tính toán và cũng có thể khó khăn trong việc đưa ra kết quả. Phương pháp này cũng không phù hợp với việc giải quyết vấn đề trong thời gian thực do cần phải có bộ dữ liệu được lưu trữ trước đó.

3 Giới thiệu về tập dữ liệu

Truy cập [tại đây](#) để có thể lấy được dữ liệu

Gồm nhiều dataset, mỗi dataset chứa các entity profile (entity files- đại diện cho một thực thể thực tế) và golden standard (groundtruth files). Dựa vào dữ liệu đầu vào ER được chia làm 2 loại :

- Clean-Clean ER: nhận hai bộ hồ sơ (không có các profile nào duplicate nhưng có thể overlapping nhau), P_1 và P_2 , và trả ra kết quả là các cặp profile là trùng nhau ($P_1 \cap P_2$)
- Dirty ER: nhận các profile và profile trùng với nó và tạo ra một tập hợp các nhóm tương đương, với mỗi nhóm tương ứng với một profile khác nhau.

Dataset Name	D1 Entities	D2 Entities	D1 Pairs	D2 Pairs	Duplicates	Average NVP	Brute-force Comparisons
Restaurants	339	2,256	1,130	7,519	89	3.3	7.64E+05
Abt-Buy	1,076	1,076	2,568	2,308	1,076	2.4	1.16E+06
Amazon-Google Products	1,354	3,039	5,302	9,110	1,104	3.9	4.11E+06
DBLP-ACM	2,616	2,294	10,464	9,162	2,224	4.0	6.00E+06
IMDB-TMDB	5,118	6,056	21,294	23,761	1,968	4.0	3.10E+07
IMDB-TVDB	5,118	7,810	21,294	20,902	1,072	3.2	4.00E+07
TMDB-TVDB	6,056	7,810	23,761	20,902	1,095	2.2	4.73E+07
Ama-Wal	2,554	22,074	14,143	114,315	853	5.2	5.64E+07
DBLP-Scholar	2,516	61,353	10,064	198,001	2,308	4.0	1.54E+08
Movies	27,615	23,182	155,436	816,009	22,863	5.6	6.40E+08
DBPedia	1,190,733	2,164,040	1.69E+07	3.50E+07	892,586	14.2	2.58E+12

4 JedAI Toolkit

4.1 Giới thiệu JedAI Toolkit

JedAI Toolkit là một bộ công cụ mã nguồn mở để giải quyết vấn đề liên quan đến quá trình data integration. Nó là một bộ công cụ hoàn chỉnh bao gồm nhiều thuật toán và phương pháp khai thác tri thức từ dữ liệu, bao gồm khai thác thực thể, liên kết và trích xuất thuộc tính.

JedAI hỗ trợ nhiều định dạng dữ liệu, bao gồm RDF, CSV và JSON, giúp người dùng có thể sử dụng dữ liệu của mình một cách linh hoạt.

JedAI có thể được sử dụng với các hình thức sau:

- Là một thư viện mã nguồn mở bao gồm các bước cho quá trình end-to-end ER workflow.
- Hỗ trợ phiên bản desktop application có hỗ trợ GUI cho phép người dùng hoặc những chuyên gia dễ dàng sử dụng.
- Có vai trò như một workbench cho phép so sánh hiệu năng giữa các end-to-end ER workflow.

Bảng so sánh JedAI Toolkit và Magellan Tool:



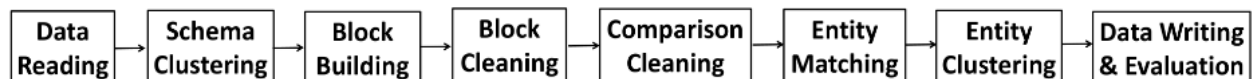
Magellan



- × **limited variety** of (blocking) methods
- × restricted to **relational data only**
- × targeted to **expert users**, focusing on development of tailor-made methods
- × offers command-line interface, **no GUI**
- ✓ **rich variety** available methods for every step in the end-to-end workflow
- ✓ applies to both **structured** and **non-structured** data
- ✓ **hands-off functionality** through default configuration of every method, but also **extensible**
- ✓ intuitive **GUI** with guidelines even for novice users
- ✓ **multi-core execution** (coming soon)

4.2 Workflow

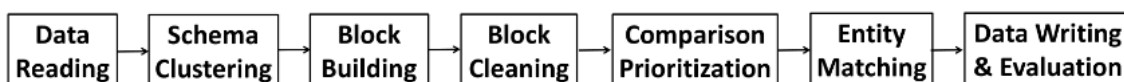
Workflow 1



Workflow 2



Workflow 3



4.3 Data Reading

Bước đầu tiên trong tất cả các workflow của JedAI là đọc dữ liệu từ đĩa vào bộ nhớ chính. Dữ liệu có thể từ một nguồn (**Dirty ER**) hoặc hai nguồn (**Clean-Clean ER**) và sau đó tiến hành chuyển đổi những dữ liệu này thành các entity profile. Mỗi profile của một thực thể (entity) bao gồm một mã định danh duy nhất và tập hợp các cặp name-value mô tả thông tin tương ứng cho thực thể đó. Một số định dạng của tệp dữ liệu được JedAI hỗ trợ:

- Dữ liệu có cấu trúc như CSV, cơ sở dữ liệu quan hệ (MySQL, PostgreSQL)
- Dữ liệu bán cấu trúc như SPARQL endpoints, các loại tệp dữ liệu RDF: XML, OWL, HDT, JSON,..
- Java serialized objects

Sự kết hợp của các định dạng trên cũng được JedAI hỗ trợ trong trường hợp Clean-Clean ER.

4.4 Schema Clustering

Schema Clustering là một bước tùy chọn nhằm mục đích nhóm các thuộc tính có cú pháp giống nhau lại với nhau mà không nhất thiết phải giống nhau về mặt ngữ nghĩa. Đây là điểm khác biệt giữa Schema Matching và Schema Clustering. Ba phương pháp hiện đang được JedAI hỗ trợ:

- Attribute Name Clustering: gom cụm các thuộc tính có tên tương tự nhau.
- Attribute Value Clustering: gom cụm các thuộc tính có giá trị thuộc tính tương tự nhau.
- Holistic Attribute Clustering: gom cụm dựa trên sự tương đồng về cả tên thuộc tính lẫn giá trị của thuộc tính.

Chỉ một trong ba phương pháp trên được hiện thực trong một workflow. Các phương pháp này sẽ được kết hợp với mô hình xử lý văn bản được đề ra bởi Text Processing component trong kiến trúc của JedAI để tiến hành so sánh độ tương đồng. Output của bước Schema Clustering là tập hợp các cụm thuộc tính tương tự nhau.

Text Processing component: cung cấp các kỹ thuật để xác định sự tương đồng giữa các bộ giá trị văn bản đại diện cho các thuộc tính riêng lẻ (trong Schema Clustering) hoặc các thực thể riêng lẻ (trong Entity Matching). Nền tảng bên dưới là mô hình vector, JedAI sẽ chuyển đổi văn bản đầu vào thành tập hợp các từ hoặc token n-grams. Trong đó: các từ n-grams với $n \in 2, 3, 4$, và token n-grams với $n \in 1, 2, 3$. Có 2 chỉ số dùng để biểu diễn các vector: TF và RF-IDF.

Để đánh giá sự tương đồng của các vector, một số độ đo được thiết lập: cosine, Jaccard, SIGMA, ARCS.... Tuy nhiên, mô hình vector này không quan tâm đến thứ tự xuất hiện của các token, do đó JedAI sử dụng đồ thị n-grams để giải quyết vấn đề trên. Tất cả các độ đo lường về sự tương đồng sẽ được chuẩn hóa về khoảng giá trị $[0, 1]$, với giá trị càng cao thì độ tương đồng càng cao.

4.5 Block Building

Tại bước Block Building, ý tưởng chính là chúng ta sẽ gom cụm các thực thể tương đồng với nhau vào chung một block nhằm mục tiêu làm giảm thời gian chạy của thuật toán và việc gom cụm hiệu quả hơn.

1. Token Blocking

- Phương pháp này gom cụm các entity dựa vào giá trị của các trường thuộc tính dữ liệu. Tất cả các thuộc tính sẽ được tập hợp lại với nhau và được sử dụng để tạo thành **token key** cho mỗi block.
- Phương pháp này được gọi là "**parameter-free**" bởi vì nó không yêu cầu bất cứ input đầu vào hay tuning parameters nào để điều chỉnh độ chính xác của mô hình. Tuy nhiên phương pháp này lại không hiệu quả đối với những thực thể có quá ít thông tin hoặc các thực thể lỗi (character-level errors).

- Để khắc phục vấn đề này, người ta phát triển thêm một số thuật toán như **Suffix Arrays**, **Extended Suffix Arrays** và **Q-Grams Blocking**. Các phương pháp này được hiện thực bằng phương pháp **hash-based** sẽ tạo ra một block phân biệt sao cho đảm bảo các thực thể match nhau khi chúng share với nhau ít nhất một key. Tuy nhiên, các thực thể có thể bị lặp lại trong cùng một block. Để giải quyết được vấn đề này, chúng ta có thể sử dụng các phương pháp dựa vào **similarity of keys** được trình bày ở các phần dưới đây.

2. Sorted Neighborhood

- Ý tưởng chính của giải thuật này là sẽ sắp xếp các TB trong block theo thứ tự alphabet và sắp xếp các record dựa trên thứ tự đó. Sau đó, một window có kích thước cố định w sẽ được trượt trên danh sách các thực thể đã được sắp xếp. Trong mỗi lần lặp, thực thể cuối cùng trong window hiện tại được so sánh với tất cả các thực thể khác trong cùng window. Điều này cho phép xác định các record có match với nhau hay không dựa vào similarity of keys.
- **Extended Sorted Neighborhood** là một bước cải thiện của SN bằng cách sắp xếp danh sách theo các blocking keys thay vì danh sách các entities. Điều này có nghĩa là mỗi block có thể bao gồm thêm TB của block đó. Điều này nhằm hỗ trợ tiết kiệm thời gian trong quá trình tính toán và cải thiện độ chính xác.

3. Locality Sensitive Hashing (LSH)

- Cuối cùng đó là phương pháp tiếp cận bằng thuật toán LSH, ví dụ như **LSH MinHash** và **LSH Superbit Blocking**.
- Ý tưởng chính của phương pháp này vẫn sẽ là tạo thành các block các thực thể trong điều kiện các key của block vượt qua được một ngưỡng cho trước dựa trên giá trị Jaccard hoặc Consine Similarity.

4.6 Block Cleaning

Đây là một **optional step** nhằm làm sạch các overlapping blocks nhằm loại bỏ các phép so sánh không cần thiết ví dụ như bao gồm các **redundant comparison** (khi phép so sánh đó đã được thực thi ở block trước đó) và **superfluous comparison** (khi phép so sánh đó bao gồm non-matching entities). Việc loại bỏ các phép so sánh không cần thiết sẽ cải thiện precision và chi phí trong recall.

Có một giả thuyết rằng số lượng một block càng lớn thì nó càng có xác suất cao xuất hiện duplicates. Trong trường hợp này, thuật toán **Size-based Block Purging** sẽ được sử dụng để loại bỏ các block vượt quá số lượng tối đa các entities cho trước. **Cardinality-based Block Purging** loại bỏ các block vượt quá số lượng comparisons cho trước. **Block Filtering** trả về một tập hợp các entities sao cho thỏa mãn $r\%$ số lượng entities của block nhỏ nhất. **Block Clustering** đảm bảo tất cả block đều thỏa mãn kích thước do người dùng tự định nghĩa.

4.7 Comparison Cleaning

Tương tự như **Block Cleaning**, bước này nhằm mục đích làm sạch một tập hợp các khối nhằm loại bỏ các phép so sánh dư thừa và không cần thiết (**redundant comparison** và **superfluous comparisons**). Tuy nhiên, khác với **Block Cleaning**, các phương pháp của nó hoạt động tốt hơn trên mức độ chi tiết của các so sánh riêng lẻ.

Các phương pháp sau đây hiện đang được hỗ trợ.

1. Comparison Propagation

Comparison Propagation loại bỏ tất cả các so sánh dư thừa bằng cách liệt kê các khối đầu vào và xây dựng một chỉ mục đảo ngược từ id của thực thể đến id của khối. Sau đó, nó đánh dấu phép so sánh $c_{i,j}$ là dư thừa nếu id của khối hiện tại lớn hơn id khối chung nhỏ nhất của các thực thể p_i và p_j .

2. Cardinality Edge Pruning (CEP)

Cardinality Edge Pruning (CEP) bao gồm một thuật toán lấy cạnh làm trung tâm (edge-centric algorithm) được kết hợp với ngưỡng cardinality toàn cục. Nó giữ lại K cạnh trên cùng (top- K edges) của toàn bộ đồ thị khối (blocking graph), với K được định nghĩa như sau:

$$K = \lfloor \frac{BC(B) \times |E|}{2} \rfloor$$

Với $BC(B)$ là viết tắt của *Blocking Cardinality* của B , nghĩa là trung bình của số lượng khối được liên kết với mỗi thực thể trong B : $BC(B) = \sum_{b_i \in B} |b_i| / |E|$

3. Cardinality Node Pruning (CNP)

Cardinality Node Pruning (CNP) bao gồm một thuật toán cắt tỉa lấy nút làm trung tâm (node-centric pruning algorithm) sử dụng ngưỡng cardinality toàn cục. Đối với mỗi nút, nó giữ lại k cạnh trên cùng (top- k edges) của vùng lân cận, trong đó k được định nghĩa như sau:

$$K = \lfloor BC(B) - 1 \rfloor$$

4. Weighed Edge Pruning (WEP)

Weighed Edge Pruning (WEP) bao gồm một thuật toán cắt tỉa lấy cạnh làm trung tâm (edge-centric pruning algorithm) sử dụng ngưỡng trọng số toàn cục. Về bản chất, nó loại bỏ tất cả các cạnh không vượt quá trọng số cạnh trung bình của toàn bộ đồ thị khối (blocking graph).

5. Weighed Node Pruning (WNP)

Weighed Node Pruning (WNP) bao gồm một thuật toán cắt tỉa lấy nút làm trung tâm (node-centric pruning algorithm) sử dụng ngưỡng trọng số cục bộ. Đối với mỗi nút, nó giữ lại những thực thể (entities) lân cận vượt quá trọng số cạnh trung bình của vùng lân cận.

Ngoài ra, còn một số phương pháp sau cũng được hỗ trợ:

- Reciprocal Cardinality Node Pruning (ReCNP)
- Reciprocal Weighed Node Pruning (ReWNP)
- BLASH
- Canopy Clusetring
- Extended Canopy Clustering

Hầu hết các phương pháp này là những kỹ thuật Meta-blocking. Tất cả các phương pháp là tùy chọn, nhưng có tính cạnh tranh, theo nghĩa là chỉ một trong số chúng có thể là một phần của quy trình Entity Resolution. Chúng có thể được kết hợp với một trong các sơ đồ trọng số sau:

1. Aggregate Reciprocal Comparisons Scheme (ARCS)

Aggregate Reciprocal Comparisons Scheme (ARCS) nắm bắt trực giác rằng các khối mà hai thực thể chia sẻ càng nhỏ thì càng có nhiều khả năng chúng khớp với nhau. Do đó, trọng số của nó được suy ra từ công thức sau:

$$ARCS(e_{i,j}) = \sum_{b_k \in B_{i,j}} \frac{1}{||b||}$$

2. Common Blocks Scheme (CBS)

Common Blocks Scheme (CBS) biểu thị thuộc tính cơ bản của các tập hợp khối dư thừa mà hai thực thể có nhiều khả năng khớp với nhau hơn khi chúng chia sẻ nhiều khối. Do đó, trọng số của mỗi cạnh bằng với số khối mà các thực thể liên kết có điểm chung sau:

$$CBS(e_{i,j}) = |B_{i,j}|$$

3. Enhanced Common Blocks Scheme (ECBS)

Enhanced Common Blocks Scheme (ECBS) cải thiện CBS bằng cách chiết khấu ảnh hưởng của các thực thể được đặt trong một số lượng lớn các khối:

$$ECBS(e_{i,j}) = CBS(e_{i,j}) \times \log \frac{|B|}{|B_i|} \times \log \frac{|B|}{|B_j|}$$

4. Jaccard Scheme (JS)

Jaccard Scheme (JS) ước tính phần khối được chia sẻ bởi hai thực thể:

$$JS(e_{i,j}) = \frac{|B_{i,j}|}{|B_i| + |B_j| - |B_{i,j}|}$$

5. Enhanced Jaccard Scheme (EJS)

Enhanced Jaccard Scheme (EJS) cải thiện JS bằng cách chiết khấu tác động của các thực thể liên quan đến quá nhiều so sánh không dư thừa (nghĩa là bậc của nút cao):

$$EJS(e_{i,j}) = JS(e_{i,j}) \times \log \frac{|E_B|}{|v_i|} \times \log \frac{|E_B|}{|v_j|}$$

4.8 Entity Matching

So sánh các cặp hồ sơ thực thể, liên kết mỗi cặp với một điểm tương đồng trong khoảng $[0,1]$. Đầu ra của nó là đồ thị tương tự (similarity graph), tức là đồ thị vô hướng, có trọng số trong đó các nút tương ứng với các thực thể và các cạnh kết nối các cặp thực thể được so sánh.

Các phương pháp lược đồ bất khả tri sau hiện được hỗ trợ:

1. Group Linkage
2. Profile Matcher, tổng hợp tất cả các giá trị thuộc tính trong một thực thể riêng lẻ thành một biểu diễn văn bản.

Cả hai phương pháp trên có thể được kết hợp với các mô hình đại diện sau đây.

1. character n-grams (n=2, 3 or 4)
2. character n-gram graphs (n=2, 3 or 4)
3. token n-grams (n=1, 2 or 3)
4. token n-gram graphs (n=1, 2 or 3)

Các bag models có thể được kết hợp với các biện pháp tương tự sau, sử dụng cả trọng số TF và TF-IDF:

1. ARCS similarity
2. Cosine similarity
3. Jaccard similarity
4. Generalized Jaccard similarity
5. Enhanced Jaccard similarity

Các mô hình đồ thị có thể được kết hợp với các phép đo tương tự đồ thị sau:

1. Containment similarity
2. Normalized Value similarity
3. Value similarity
4. Overall Graph similarity

Các mô hình pre-trained embedding ở cấp độ từ hoặc ký tự cũng được hỗ trợ trong việc kết hợp với độ tương tự cosine hoặc khoảng cách Euclidean.

4.9 Entity Clustering

Entity Clustering là một phương pháp nhóm các cụm dữ liệu tương đương với nhau bằng cách sử dụng similarity graph được trả ra bởi phương pháp Entity Matching. Mỗi cụm dữ liệu của Entity Clustering sẽ tương ứng với một đối tượng trong thế giới thực. Ở bài báo cáo này chúng ta sẽ đi bàn về ba thuật toán sau:

- Phương thức Cut Clustering: dựa trên việc tìm các cắt tối thiểu của các cạnh trong đồ thị tương đồng, và được đánh giá trên dữ liệu tham chiếu và web.
- Phương thức Markov Clustering (MCL): là một thuật toán phân cụm không giám sát nhanh và có khả năng mở rộng dựa trên mô phỏng luồng ngẫu nhiên trong các đồ thị. Người ta nhận thấy rằng thuật toán này cho thấy tính hiệu quả và chất lượng cao trong các ứng dụng trong các bài toán về sinh.
- Phương thức Correlation Clustering: Mục đích ban đầu được đề xuất nhằm để phân cụm các đồ thị với các nhãn cạnh nhị phân chỉ ra sự tương quan hoặc thiếu tương quan của các nút được kết nối. Vì thế ta có thể gán nhãn cho các cạnh dựa trên điểm tương đồng của các bản ghi (trọng số cạnh) và một giá trị ngưỡng, điều này làm cho nó hấp dẫn như một thuật toán không ràng buộc cho phân cụm các đồ thị tương đồng.
- Phương thức Unique Mapping Clustering: là thuật toán định vị cơ sở dữ liệu có khả năng xử lý hàng triệu thực thể. Thuật toán có thể được mở rộng dễ dàng với các hàm điểm được tùy chỉnh để tích hợp kiến thức lĩnh vực. Nó cũng cung cấp sự cân đối tự nhiên giữa độ chính xác và độ phủ, cũng như giữa tính toán và độ phủ.
- Phương thức Row-Column Clustering là một phương pháp phân cụm phân loại hàng và/hoặc cột dựa trên các tọa độ của chúng từ một không gian nhiều chiều tương ứng với toàn bộ chiều dữ liệu dạng bảng chéo đầu vào hoặc một không gian nhiều chiều có số chiều nhỏ hơn so với toàn bộ số chiều của tập dữ liệu đầu vào.

4.10 Similarity Join

Similarity Join là một quá trình so sánh các dữ liệu để có thể tìm được các cặp dữ liệu có mức độ tương đồng với nhau vượt qua một ngưỡng nào đó. Similarity joins có nhiều ứng dụng trên nhiều lĩnh vực làm sạch dữ liệu, khai phá dữ liệu. Ví dụ đề xuất các thuật toán nhằm tính toán độ tương đồng giữa các cặp người dùng và sau đó đưa ra đề xuất cho người dùng có sở thích tương tự. Trong các nhiệm vụ làm sạch dữ liệu, similarity Join có thể được sử dụng như một hoạt động nguyên thủy (primitive operation) để xác định các biểu diễn khác nhau (nhưng tương tự nhau) của cùng một thực thể.

Định nghĩa của toán tử Similarity Join giữa hai tập dữ liệu R và S được định nghĩa như sau:

$$R \bowtie_{\theta_\varepsilon(r,s)} S = \{ \langle r, s \rangle \mid \theta_\varepsilon(r, s), r \in R, s \in S \},$$

với $\theta_\varepsilon(r, s)$ biểu diễn cho một ràng buộc tương đồng. Ràng buộc này đặc trưng cho sự tương đồng của một cặp (r, s) thỏa mãn một ngưỡng ε , ví dụ như $\text{dist}(r, s) < \varepsilon$. Ở đây sẽ có một số ví dụ về một số hàm đo độ tương đồng với r và s là hai bản ghi là

- Jaccard similarity: $\text{sim}_J(r, s) = \frac{|r \cap s|}{|r \cup s|}$.
- Dice similarity: $\text{sim}_D(r, s) = \frac{2 \cdot |r \cap s|}{|r| + |s|}$.
- Cosine similarity: $\text{sim}_C(r, s) = \frac{|r \cap s|}{\sqrt{|r| \cdot |s|}}$.
- Overlap similarity: $\text{sim}_O(r, s) = |r \cap s|$.

với $|r|$ là số phần tử của r . Similarity Join là một quá trình có nhiều thuật toán sử dụng trên nhiều dạng dữ liệu nhưng ở bài báo cáo này ta chỉ bàn về hai loại là Token-base similarity join và Character-based similarity join.

Trước hết bàn về **Token-base similarity join** là bài toán chuyển chuỗi thành một tập hợp các từ vựng sau đó tìm các cặp từ vựng của một chuỗi có điểm tương đồng cao bằng các hàm tương đồng như Jaccard hoặc Cosine. Ở đây từ vựng là một đơn vị văn bản có thể là từ trong một chuỗi. Đầu tiên, ta sẽ bàn về thuật toán cơ bản nhất là Allpairs. Cụ thể thuật toán Allpair sẽ sử dụng một độ đo tương đồng và một ngưỡng tương đồng do người dùng xác định. Thuật toán gồm hai giai đoạn: lọc và xác minh. Trong giai đoạn lọc, nó sử dụng một chỉ mục đảo ngược (inverted index) để tìm các cặp khả thi có chung ít nhất một token. Trong giai đoạn xác minh sẽ tính toán độ tương đồng chính xác của từng cặp khả thi và xuất ra những cặp thỏa mãn ngưỡng. Dưới đây là mã giả cho thuật toán AllPairs

```
Data:  $R$ , invertedIndex,  $t$ 
Result:  $\{(r, s) \mid (r, s) \in R \times R, r \neq s, \text{sim}(r, s) \geq t\}$ 
for  $r \in R$  do
    invertedIndex  $\leftarrow \{\}$  ;
    for  $token \in \text{GetPrefix}(r, t)$  do
        for  $s \in \text{GetList}(\text{invertedIndex}, token)$  do
             $candidates \leftarrow candidates \cup s$ ;
        end
    end
    for  $s \in candidates$  do
        Verify( $r, s, t$ );
    end
end
```

Algorithm 1: AllPairs algorithm

Ngoài ra chúng ta còn những bộ lọc để giúp giảm thiểu số lượng các cặp tương đồng cần phải so sánh, như là:

- Bộ lọc tiền tố (Prefix filter): Ý tưởng cơ bản của việc lọc tiền tố là thay vì lập chỉ mục cho tất cả các token trong mỗi bản ghi, ta chỉ cần lập chỉ mục và kiểm tra một vài token đầu tiên của mỗi bản ghi trong giai đoạn sinh ứng viên. Như vậy, kích thước chỉ mục và số cặp khả thi có thể được giảm đáng kể. Bổ đề dưới đây sẽ giúp chúng ta đưa ra mô tả chính thức của nguyên tắc lọc tiền tố:
- Bộ lọc vị trí (Position filter): Trong giai đoạn xây dựng chỉ mục, việc xử lý như vậy là không tránh khỏi bởi vì đối với bất kỳ bản ghi r nào, chúng ta không biết r sẽ được ghép với những bản ghi r khác nào. Tuy nhiên, trong giai đoạn sinh ứng viên, có thể tăng ngưỡng tương đồng lên bằng cách xem xét đến độ dài của cả hai bản ghi. Các ánh xạ ngưỡng tương đồng được cải tiến như sau: (dựa trên đó, một kỹ thuật lọc mạnh hơn, tức là lọc vị trí, có thể được thực hiện). Position filter sử dụng thông tin vị trí của các mã thông báo chung giữa hai bản ghi để giảm số lượng ứng viên một cách đáng kể.
- Bộ lọc hậu tố (Suffix filter): Kích thước số cặp khả thi có thể vẫn tăng với tốc độ bình phương theo số lượng bản ghi, ngay cả khi sử dụng bộ lọc vị trí. Để giảm chi phí tính toán độ tương đồng chính xác, lọc hậu tố được giới thiệu trong Xiao et al. (2008) dựa trên hai quan sát quan trọng. Quan sát đầu tiên là rằng buộc chồng chéo có thể được chuyển đổi thành ràng buộc khoảng cách Hamming tương đương như được hiển thị trong công thức sau

$$\text{sim}_O(r, s) \geq t \Leftrightarrow \text{dis}_H(r, s) \leq |r| + |s| - 2t$$

với $\text{dis}_H(r, s)$ là khoảng cách Hamming của (r_s, s_s) . Gọi r_p và r_s là tiền tố và hậu tố của r . Khoảng cách Hamming của $\langle r_s, s_s \rangle$ sẽ bị chặn bởi đẳng thức sau nếu $\text{sim}_O(r, s) \geq t$.

$$\text{dis}_H(r_s, s_s) \leq H_{\max} = |r| + |s| - 2t - (|r_p| + |s_p| - 2\text{sim}_O(r_p, s_p))$$

Nhận xét thứ hai là khoảng cách Hamming có thể ước lượng một cách hiệu quả bằng cách sử dụng đệ quy. Vì thế ta có thể dễ dàng sử dụng đệ quy tính toán hiệu quả từng phần bên trái và phải của r để có được khoảng cách.

Từ các bộ lọc trên thì một số nhà nghiên cứu đã phát triển ra các thuật toán khác để nhằm mục đích cải thiện hơn quá trình xử lý của AllPairs như thuật toán Position Prefix Join. Position Prefix Join là mở rộng của AllPairs bằng cách sử dụng position filter và prefix filter để giúp loại bỏ các cặp từ vựng khả thi và sử dụng hàm tương tự Jaccard. Đây là kỹ thuật được giới thiệu trong bài báo "Efficient similarity joins for near-duplicate detection." của Xiao, Chuan và một số tác giả khác.

Character-based similarity join tương tự như Token-base similarity join nhưng lại dùng các ký tự để có thể định được sự tương đồng, ví dụ như khoảng cách chỉnh sửa (edit distance) là số lần thực hiện các thao tác chỉnh sửa đơn lẻ trên các ký tự (tức là thêm, xóa hoặc thay thế) để chuyển đổi chuỗi này thành chuỗi kia. So sánh với token-base similarity join dựa trên sự tương đồng trên "token" thì khoảng cách chỉnh sửa sẽ nhạy cảm hơn với của các từ vựng trong chuỗi. Ở đây, ta sẽ quan tâm tới hai thuật toán là AllPairs và FastSS. Về thuật toán AllPairs: Gần như tương tự với thuật toán AllPairs trong trường hợp Token-base similarity join nhưng với phần tử đơn vị là một character.

Trước khi nói về FastSS ta cần biết định nghĩa về edit distance là số lượng thao tác nhỏ nhất để chuyển đổi chuỗi này thành một chuỗi khác bằng các thao tác thêm, xóa và thay thế. Dưới đây

là chương trình quy hoạch động để tính edit distance của hai chuỗi s_1 và s_2 là

$$\begin{aligned}d[i, 0] &= i \\d[0, j] &= j \\d[i, j] &= \min(d[i - 1, j] + 1 \\&\quad d[i, j - 1] + 1 \\&\quad d[i - 1, j - 1] + (\text{if } s1[i] = s2[j] \text{ then } 0 \text{ else } 1))\end{aligned}$$

Trở lại với thuật toán FastSS là Fast Similarity Search sử dụng kết hợp các kỹ thuật và thuật toán như là bảng băm, NR-grep, tìm kiếm với Neighborhood Generation giúp loại bỏ các cặp khả thi và tính toán nhanh edit distance.

4.11 Comparison Prioritization

Comparison Prioritization là kỹ thuật ra quyết định liên quan đến thứ hạng của các sự lựa chọn dựa trên các tiêu chí đã được đặt ra từ trước, kỹ thuật này được dùng khi có quá nhiều sự lựa chọn thỏa mãn và rất khó để quyết định được cái nào là tốt nhất.

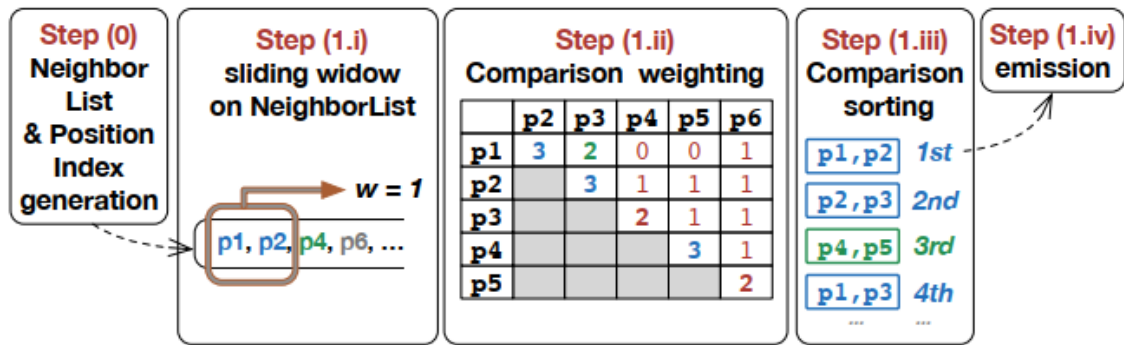
Ban đầu cần xác định được các tiêu chí quan trọng nhất để đưa ra thứ tự ưu tiên cho các phương án. Các tiêu chí này có thể là những yếu tố như chi phí, chất lượng, tính sẵn có hoặc bất kỳ yếu tố liên quan nào khác. Sau khi xác định các tiêu chí, gán trọng số cho mỗi tiêu chí dựa trên mức độ quan trọng của nó.

Kế đó so sánh mỗi lựa chọn với các tiêu chí và gán điểm cho mỗi lựa chọn dựa trên mức độ đáp ứng của nó với các tiêu chí. Các điểm số có thể là số hoặc là đánh giá về chất lượng, phụ thuộc vào tính chất của các tiêu chí. Cuối cùng, bạn tính tổng điểm cho mỗi lựa chọn bằng cách nhân điểm cho mỗi tiêu chí với trọng số tương ứng và cộng các kết quả lại.

Lựa chọn có tổng điểm cao nhất được xem là lựa chọn tốt nhất. Comparison Prioritization có thể hữu ích trong rất nhiều bối cảnh ra quyết định, từ ra quyết định cá nhân đến ra quyết định kinh doanh. Nó cho phép so sánh khách quan các lựa chọn khác nhau dựa trên các tiêu chí quan trọng nhất và giúp đưa ra các quyết định thông minh.

Các phương thức Comparison Prioritization được jedAI hỗ trợ:

- Local Progressive Sorted Neighborhood (LS-PSN): Phương pháp này chỉ áp dụng trọng số đã được chọn cho các so sánh của một kích thước cửa sổ cụ thể, từ đó xác định một thứ tự thực hiện cục bộ. Ở trung tâm của phương pháp này là hai cấu trúc dữ liệu:
 - (i) NL, đó là một mảng bao gồm Neighbor List để bao trọn profile, sao cho $NL[i]$ chỉ định id của profile được đặt ở vị trí thứ i trong Neighbor List, được hiển thị ở Bước 1.i của hình dưới.
 - (ii) PI, viết tắt của "Position Index", là một chỉ mục đảo ngược trở từ id của profile đến các vị trí trong NL. Nó được triển khai bằng một mảng sử dụng id của profile như các chỉ số, sao cho $PI[i]$ trả về danh sách các vị trí liên quan đến profile trong NL. Mảng này tăng tốc độ ước tính trọng số so sánh, vì nó giảm thiểu chi phí tính toán để truy xuất các hàng xóm của bất kỳ profile nào trong cửa sổ hiện tại, như mô tả bên dưới. Lưu ý rằng thay vì một chỉ mục vị trí, LS-PSN có thể sử dụng một chỉ mục băm có so sánh làm khóa và trọng số làm giá trị. Tuy nhiên, phương pháp này sẽ tăng cả không gian lưu trữ và độ phức tạp thời gian của trọng số so sánh.



- Global Progressive Sorted Neighborhood (GS-PSN): Nhược điểm chính của LS-PSN là thứ tự thực hiện cục bộ nó xác định cho một kích thước cửa sổ cụ thể. Điều này có nghĩa là LS-PSN có thể phát ra các so sánh giống nhau nhiều lần cho hai hoặc nhiều kích thước cửa sổ khác nhau, vì nó không nhớ các so sánh đã làm trong quá khứ. GS-PSN nhằm vượt qua điều này bằng cách xác định một thứ tự thực hiện toàn cục cho tất cả các so sánh trong một phạm vi các kích thước cửa sổ $[1; w_{max}]$. Điều này cho phép phát ra các so sánh một lần, và chỉ trả về so sánh tốt nhất tiếp theo, cho đến khi Comparison List trống.
- Progressive Block Scheduling (PBS): Nó gán cho mỗi khối (block) một trọng số tỉ lệ với khả năng chứa profile trùng nhau và sau đó, nó sắp xếp tất cả các khối theo thứ tự giảm trọng số. Tuy nhiên, mặc dù chúng ta muốn sử dụng chức năng này cho Progressive ER, nó không áp dụng được, vì:
 - Trọng số của nó không thể tổng quát hóa cho Dirty ER, áp dụng độc quyền cho Clean-clean ER
 - Nó không chỉ định thứ tự thực hiện so sánh bên trong các khối có hơn hai profiles
- Progressive Entity Scheduling (PPS): PPS dựa trên khái niệm về khả năng trùng lặp (duplication likelihood). Trong Clean-clean ER, khả năng trùng lặp của profile thuộc P_1 tương ứng với khả năng nó có một trùng khớp (match) trong P_2 . Tuy nhiên, trong Dirty ER, khả năng trùng lặp của một profile tương đương với kích thước của cụm tương đương của nó. Thực tế, PPS nhằm sắp xếp tất cả các profile theo khả năng trùng lặp giảm dần, tạo thành một cấu trúc dữ liệu được gọi là Sorted Profile List. Sau đó, di chuyển từ đầu đến cuối danh sách này, PPS lần lượt đi qua mỗi profile, phát ra các so sánh được đánh trọng số top-k mà profile đó tham gia, với khả năng khớp giảm dần.
- Progressive Local Top Comparisons (PLTC): Thuật toán bắt đầu bằng cách xác định kích thước cửa sổ và tạo ra một danh sách được sắp xếp của tất cả các so sánh có thể có giữa các profile trong cửa sổ đó. Sau đó, nó lặp lại việc chọn top-k so sánh tiềm năng nhất từ danh sách, thực hiện phép so sánh đó, cập nhật trọng số của các profile bị ảnh hưởng. Quá trình này tiếp tục với phương pháp cửa sổ trượt, trong đó cửa sổ dịch chuyển một lần theo mỗi profile cho đến khi tất cả các profile đã được so sánh với tất cả các profile khác trong kích thước cửa sổ đã chỉ định.
- Progressive Global Top Comparisons (PGTC): Thuật toán bắt đầu bằng việc tạo ra một danh sách được sắp xếp cho tất cả các so sánh có thể có giữa các profile. Sau đó, thuật toán lặp đi lặp lại chọn lựa top-k so sánh tiềm năng nhất từ danh sách, thực hiện phép so sánh đó, cập nhật trọng số của các profile bị ảnh hưởng. Quá trình này tiếp tục cho đến khi đạt được tiêu chí dừng, chẳng hạn như số lượng so sánh tối đa hoặc ngưỡng tối thiểu cho trọng số.

5 Kết quả và đánh giá

5.1 Hướng dẫn chạy source code

- **Bước 1:** Cài đặt Docker trên máy tính. Để đơn giản trong quá trình sử dụng, khuyến khích cài đặt Docker Desktop (download ở [đây](#))

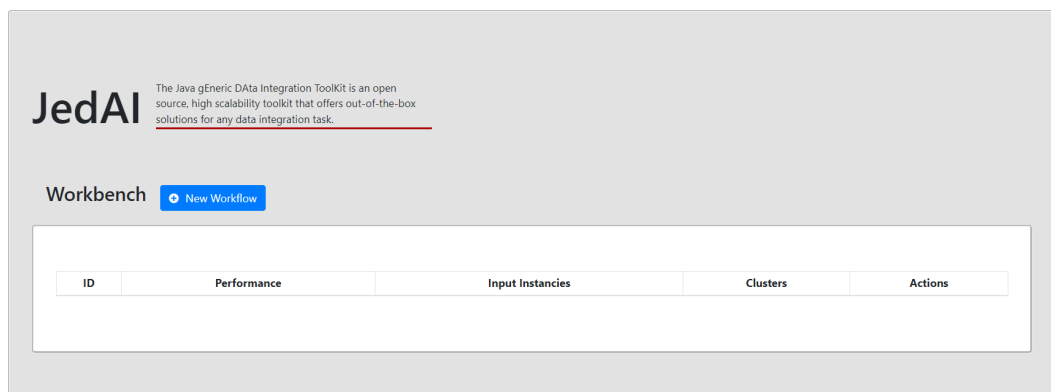
- **Bước 2:** Pull container JedAI/webapp về máy local bằng lệnh sau:

```
docker pull gmandi/jedai-webapp
```

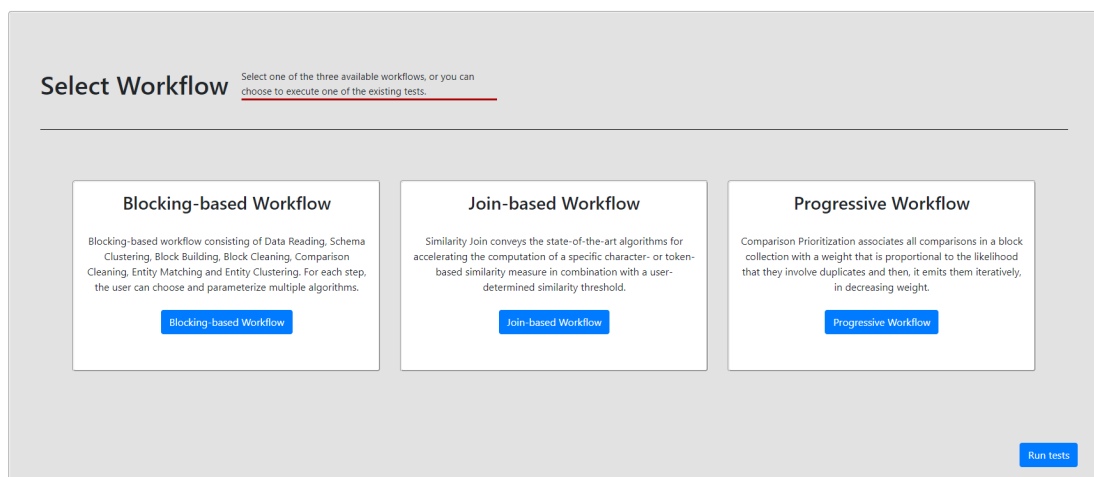
- **Bước 3:** Sau khi pull về máy, chúng ta sẽ tiến hành chạy ứng dụng ở port 8080 như sau:

```
docker run -p 8080:8080 gmandi/jedai-webapp
```

- **Bước 4:** Sau khi cài đặt thành công, chúng ta sẽ truy cập vào ứng dụng như sau:



- **Bước 5:** Đối với phương pháp tiếp cận, chúng ta sẽ có 3 workflow như sau:



5.2 Hướng dẫn load dữ liệu

Phần mô tả của các tập dữ liệu đã được trình bày ở phần 3 của báo cáo. Ở đây để hướng dẫn minh họa cho phần load dữ liệu, nhóm sẽ tiến hành phân tích data matching giữa tập dữ liệu của **Amazon** và **Google Product**.

- Tập dữ liệu sản phẩm của Amazon

Entity ID: 1	Entity URL: b000jz4hqo
manufacturer: broderbund price: 0 title: clickart 950 000 - premier image pack (dvd-rom)	
Entity ID: 2	Entity URL: b0006zf55o
title: ca international - arcserve lap/desktop oem 30pk description: oem arcserve backup v11.1 win 30u for laptops and desktops manufacturer: computer associates price: 0	
Entity ID: 3	Entity URL: b00004tkvy
price: 0 title: noah's ark activity center (jewel case ages 3-8) manufacturer: victory multimedia	
Entity ID: 4	Entity URL: b000g80lqo
manufacturer: sage software title: peachtree by sage premium accounting for nonprofits 2007 description: peachtree premium accounting for nonprofits 2007 is the affordable easy to use accounting solution that provides you with donor/grantor management. price: 599.99	
Entity ID: 5	Entity URL: b0006se5bq
title: singing coach unlimited price: 99.99 description: singing coach unlimited - electronic learning products (win me nt 2000 xp) manufacturer: carry-a-tune technologies	

- Tập dữ liệu sản phẩm của Google Product

Entity ID: 1

Entity URL: <http://www.google.com/base/feeds/snippets/11125907881740407428>

price: 38.99
description: learning quickbooks 2007
title: learning quickbooks 2007
manufacturer: intuit

Entity ID: 2

Entity URL: <http://www.google.com/base/feeds/snippets/11538923464407758599>

title: superstart! fun with reading & writing!
price: 8.49
description: fun with reading & writing! is designed to help kids learn to read and write better through exercises puzzle-solving creative

Entity ID: 3

Entity URL: <http://www.google.com/base/feeds/snippets/11343515411965421256>

price: 637.99
title: qb pos 6.0 basic software
description: qb pos 6.0 basic retail mngmt software. for retailers who need basic inventory sales and customer tracking.
manufacturer: intuit

Entity ID: 4

Entity URL: <http://www.google.com/base/feeds/snippets/12049235575237146821>

price: 12.95
description: save spectacle city by disrupting randall underling's plan to drive all the stores out of business and take over the city. solve
title: math missions: the amazing arcade adventure (grades 3-5)

Entity ID: 5

Entity URL: <http://www.google.com/base/feeds/snippets/12244614697089679523>

price: 805.99
manufacturer: adobe software
description: adobe cs3 production premium mac upgrade from production studio premium or standard
title: production prem cs3 mac upgrad



- Ground-Truth file

Entity ID: 650

Entity URL: b000nlwxii

price: 0
manufacturer: emedia music
title: emedia guitar basics

Entity ID: 233

Entity URL: http://www.google.com/base/feeds/snippets/18046511712294457433

price: 16.64
description: a fun way to learn acoustic or electric guitar at your own pace. songs
title: emedia music corp emedia guitar basics

Entity ID: 577

Entity URL: b000ehs6ic

price: 49.95
title: roxio popcorn 2 (mac)
manufacturer: roxio
description: popcorn 2 helps you easily make high quality copies of your dvds and

Entity ID: 148

Entity URL: http://www.google.com/base/feeds/snippets/9897777989987457856

description: requirements: macintosh computer with intel or powerpc processor (c
title: roxio popcorn 2 video conversion software
price: 44.99

Entity ID: 1286

Entity URL: b000ht55i

title: pcdefense
manufacturer: avanquest software
price: 29.95
description: pcdefense protects you from: spyware - spyware scan removes the sp

Entity ID: 697

Entity URL: http://www.google.com/base/feeds/snippets/17411408037622367354

price: 26.11
description: offers full protection against a wide range of spyware crimeware malw
title: avanquest usa llc pcdefense

Entity ID: 1098

Entity URL: b000g0lkcg

title: pitstop pro 7.0
description: pitstop pro 7.0
manufacturer: enfocus software
price: 699

Entity ID: 1965

Entity URL: http://www.google.com/base/feeds/snippets/14235968984166622494

description: enfocus software pp7.0-sg-001 : enfocus software pitstop pro 7.0 - p
price: 558.97
title: enfocus software pp7.0-sg-001 - pitstop pro 7.0

Entity ID: 891

Entity URL: b000qxd2tc

title: onone genuine fractals 5 - full 1u
manufacturer: onone software
price: 159.95
description: enlarge your images over 1000% with no loss in image quality. genuir

Entity ID: 2763

Entity URL: http://www.google.com/base/feeds/snippets/7653075580455328736

title: onone software inc. gfs-50211 - genuine fractals 5 full 1u
description: onone software inc. gfs-50211 : enlarge your images over 1000% with
price: 148.97

5.3 Kết quả

Đối với tập dữ liệu trên, nhóm tiến hành đánh giá data matching dựa trên **title** của các sản phẩm

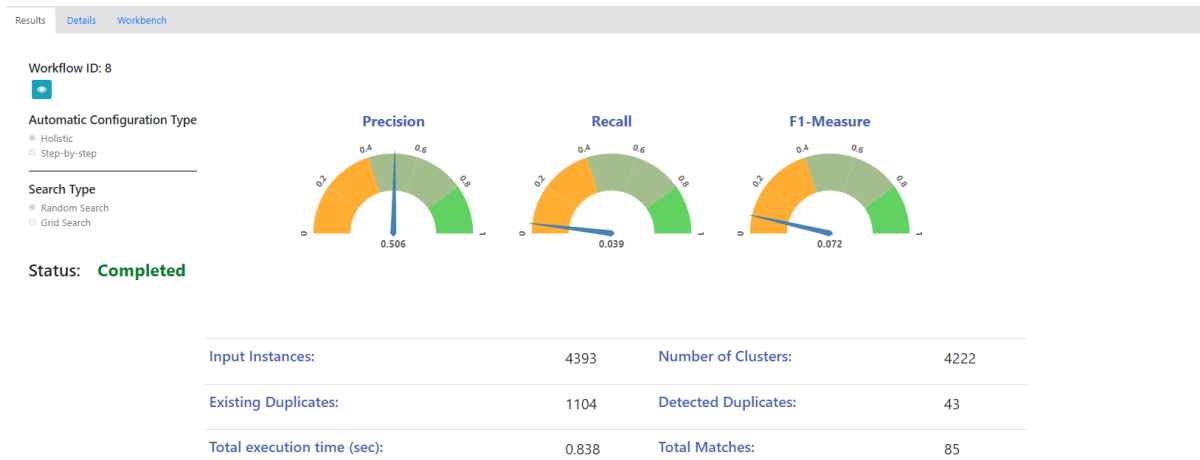
5.3.1 Block-based workflow

- Ở bước **Schema Clustering**, nhóm sẽ đánh giá bằng phương pháp **No Schema Clustering**.
- Ở bước **Block Building**, nhóm sẽ đánh giá bằng phương pháp **LSH MinHash Blocking**
- Ở bước **Block Cleaning**, nhóm sẽ đánh giá bằng phương pháp **Block Filtering**
- Ở bước **Comparison Cleaning**, nhóm sẽ đánh giá bằng phương pháp **No Cleaning**
- Ở bước **Entity Matching**, nhóm sẽ đánh giá bằng phương pháp **Group Linkage**
- Ở bước **Entity Clustering**, nhóm sẽ đánh giá bằng phương pháp **Unique Mapping Clustering**

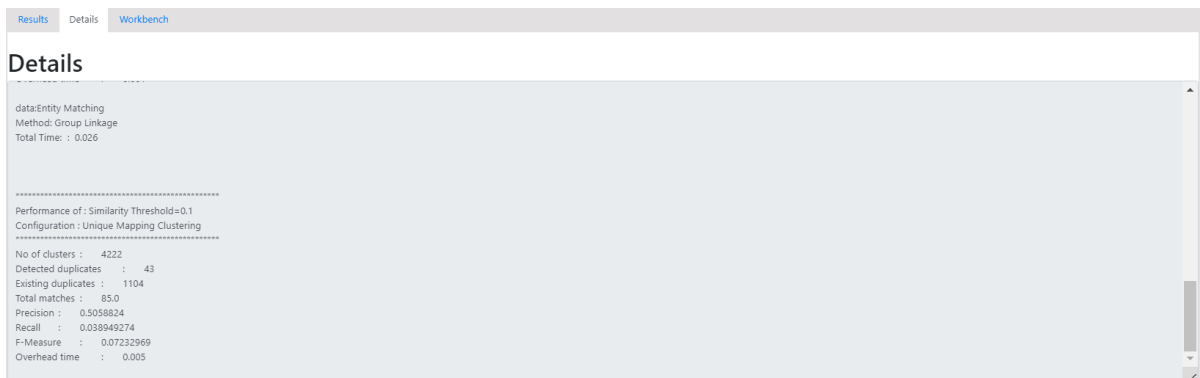


– Đánh giá:

• Kết quả:



• Chi tiết:



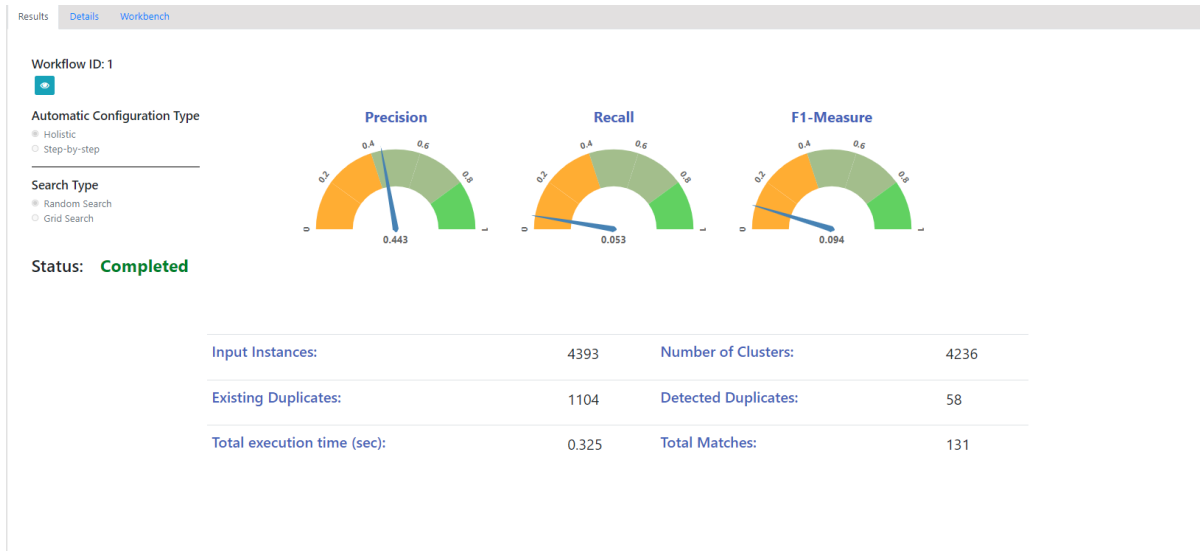
• Workbench:

	Method	Precision	Recall	F1 Measure	AUC	Time (sec)			
8	Total	0.51	0.04	0.07	-	0.84	4393	4222	

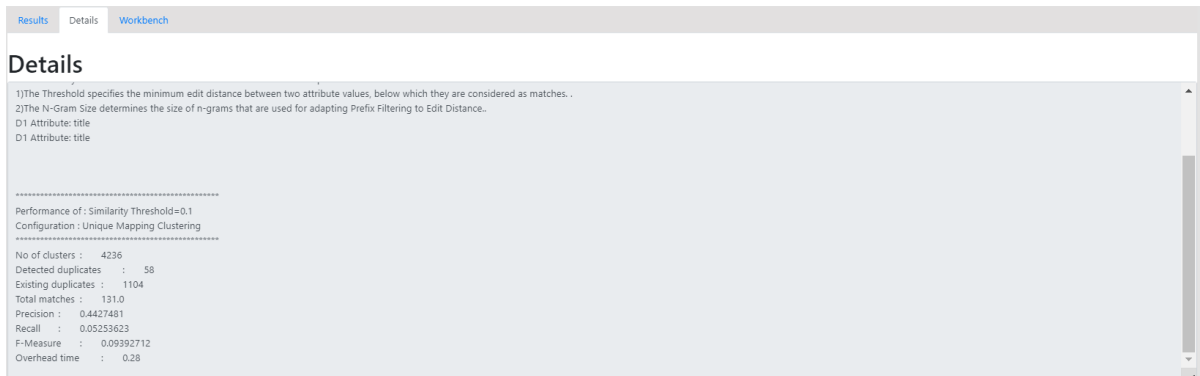
5.3.2 Join-based workflow

- Ở bước **Similarity Join**, nhóm sẽ đánh giá bằng phương pháp **All Pairs (character-based)**.
- Ở bước **Entity Clustering**, nhóm sẽ đánh giá bằng phương pháp **Unique Mapping Clustering**
- Đánh giá:


• Kết quả:



• Chi tiết:



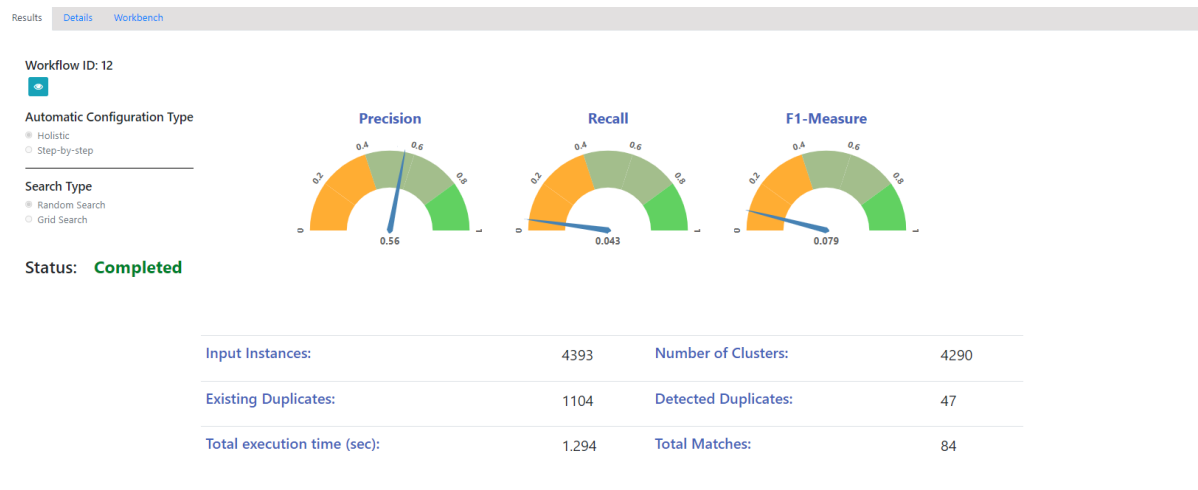
• Workbench:

ID	Performance							Input Instances	Clusters	Actions
	Method	Precision	Recall	F1 Measure	AUC	Time (sec)				
1	Total	0.44	0.05	0.09	-	0.33		4393	4236	

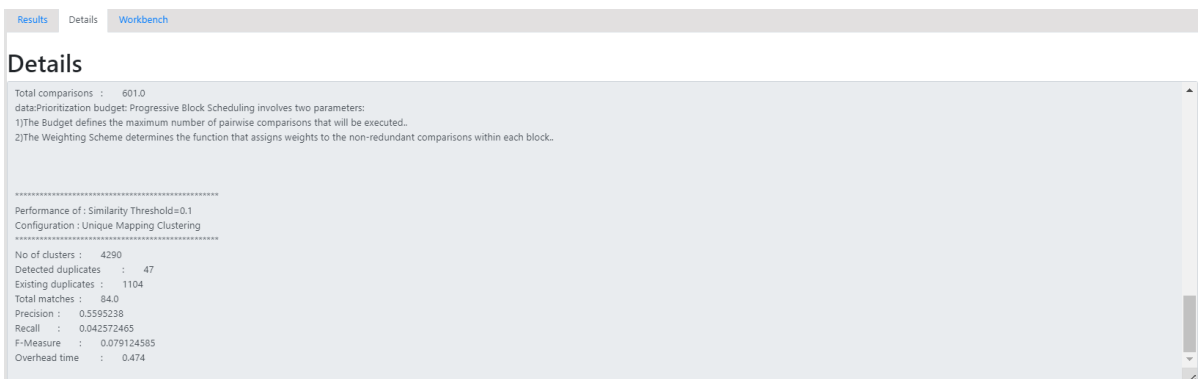
5.3.3 Progressive workflow

- Ở bước **Schema Clustering**, nhóm sẽ đánh giá bằng phương pháp **No Schema Clustering**.
- Ở bước **Block Building**, nhóm sẽ đánh giá bằng phương pháp **LSH MinHash Blocking**
- Ở bước **Block Cleaning**, nhóm sẽ đánh giá bằng phương pháp **Block Filtering**
- Ở bước **Comparison Cleaning**, nhóm sẽ đánh giá bằng phương pháp **No Cleaning**
- Ở bước **Prioritization**, nhóm sẽ đánh giá bằng phương pháp **Progressive Block Scheduling**
- Ở bước **Entity Matching**, nhóm sẽ đánh giá bằng phương pháp **Group Linkage**
- Ở bước **Entity Clustering**, nhóm sẽ đánh giá bằng phương pháp **Unique Mapping Clustering**
- Đánh giá:

• Kết quả:



• Chi tiết:



• Workbench:

12	Method	Precision	Recall	F1 Measure	AUC	Time (sec)	4393	4290	
	Total	0.56	0.04	0.08	0.024	1.29			

5.4 Đánh giá













Sau đây là đánh giá chung của quá trình chạy theo các workflow trên

- Workbench

Workflow Execution

Press "Execute Workflow" to run the algorithm. You can export the results to a file with the "Export" button.

ResultsDetailsWorkbench

ID	Performance							Input Instances	Clusters	Actions
1		Method	Precision	Recall	F1 Measure	AUC	Time (sec)	4393	4236	  
		Total	0.44	0.05	0.09	-	0.33			
8		Method	Precision	Recall	F1 Measure	AUC	Time (sec)	4393	4222	  
		Total	0.51	0.04	0.07	-	0.84			
12		Method	Precision	Recall	F1 Measure	AUC	Time (sec)	4393	4290	  
		Total	0.56	0.04	0.08	0.024	1.29			

- Explore

Explore

Entity ID: 33
Entity URL: b000cpmtwk
description: internet movies 2 lets you take advantage of t
manufacturer: x-oom
price: 0
title: x-oom internet movies 2

Entity ID: 1978
Entity URL: http://www.google.com/base/feeds/snippets/141770843571919
description: video clips trailers or even entire movies the in
price: 22.9
title: x-oom internet movies 2

Entity ID: 34
Entity URL: b0002qnd2y
price: 29.99
description: customize your forms to your specific needs p
manufacturer: valusoft
title: form workshop 1200

Entity ID: 1990
Entity URL: http://www.google.com/base/feeds/snippets/975032998818002
description: overview the complete solution for all of your
title: form workshop 1200
price: 12.9

Entity ID: 68
Entity URL: b000fa5ens
description: mom standard ops mgmt lic 2005 eng mlp 5 o
manufacturer: microsoft
price: 0
title: mom standard ops mgmt lic 2005 english mlp 5 oml

Entity ID: 1538
Entity URL: http://www.google.com/base/feeds/snippets/183841366897466
description: mom standard ops mgmt lic 2005 english mlp
title: mom standard ops mgmt lic 2005 english mlp 5 oml
price: 889.99

Entity ID: 97
Entity URL: b0002719lk
title: tournament poker 2005
description: tournament poker: no limit texas hold'em lets
manufacturer: eagle games
price: 20.99

Entity ID: 1596
Entity URL: http://www.google.com/base/feeds/snippets/184160933655156
description: no other texas hold'em poker software lets you
price: 7.95
manufacturer: eagle games
title: eagle games egl 150 tournament poker - no limit texa

Entity ID: 105
Entity URL: b000j4k804
manufacturer: topics entertainment
description: instant immersion spanish (audio book) (audio
price: 0
title: instant immersion spanish (audio book)

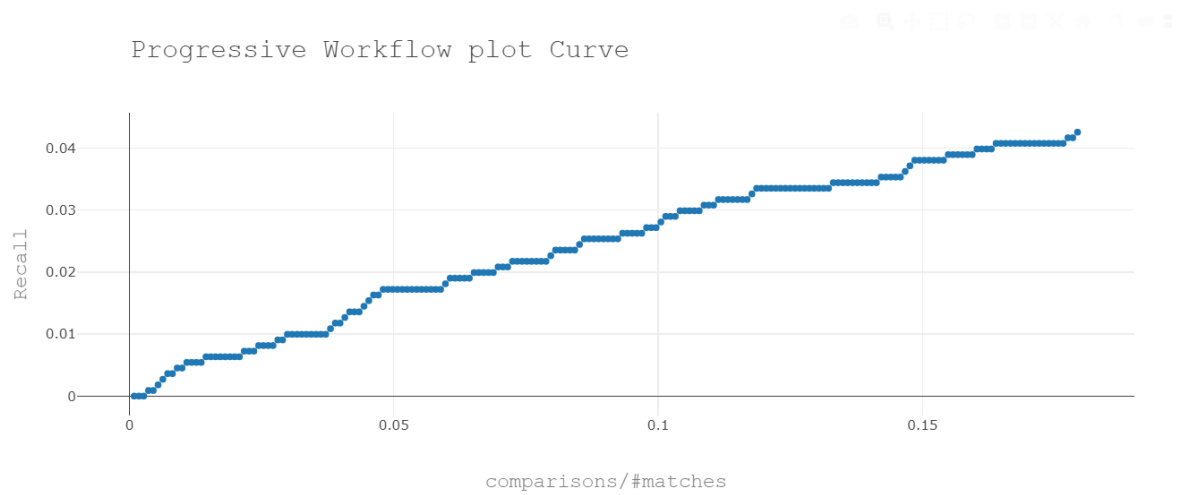
Entity ID: 1614
Entity URL: http://www.google.com/base/feeds/snippets/184118751625621
manufacturer: topics entertainment
description: instant immersion spanish (audio book) (audio
price: 23.61
title: topics entertainment 40248 instant immersion spanis

< 1 2 3 ... 16 > >>

- Plot

Recall Curve

AUC: 0.024



6 Kết luận

Nhìn chung, việc dữ liệu phân tán ở nhiều nơi đã không còn là vấn đề quá xa lạ trong kỷ nguyên của Big Data. Dữ liệu của chính tổ chức, dữ liệu từ đối tác, dữ liệu từ các bên thứ ba,... và còn rất nhiều nguồn dữ liệu khác nữa cần được tích hợp để có cái nhìn tổng quan và chính xác về dữ liệu. Để giải quyết một trong các thử thách của data integration thì data matching là một bước không thể thiếu để loại bỏ các trùng lặp trong dữ liệu. Trong bài tập lớn lần này, nhóm chúng em đã tìm hiểu về một số cách tiếp cận phổ biến được sử dụng trong data matching cụ thể là ruled-based, learning-based, clustering và phương pháp tiếp cận bằng xác suất. Mỗi phương pháp tiếp cận đều được tìm hiểu về cách thức tiếp cận, các ưu và nhược điểm. Bên cạnh đó nhóm cũng tìm hiểu về một công cụ mã nguồn mở hỗ trợ quá trình tích hợp dữ liệu đó là JedAI, cụ thể hơn là tìm hiểu sơ lược về các thuật toán được áp dụng trong từng bước của workflow của công cụ này. Nhóm đã tiến hành sử dụng công cụ này để thực hiện data matching từ nguồn dữ liệu thầy cung cấp và có một số đánh giá nhận xét kết quả đạt được. Trong quá trình làm bài nhóm cũng gặp một số khó khăn nhất định vì đây là vấn đề lần đầu tiên nhóm tiếp xúc và tìm hiểu, có thể còn nhiều sai sót và chưa cập nhật nhưng sơ lược cũng đã hoàn thành trọn vẹn bài làm.

7 Tài liệu tham khảo

- [1] George Papadakis, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas, Manolis Koubarakis (February 2020). “*Domain and Structure-Agnostic End-to-End Entity Resolution with JedAI*”, Paris Descartes University, France.
- [2] George Papadakis, George Mandilaras, Luca Gagliardelli, Giovanni Simonini, Emmanouil Thanos, George Giannakopoulos, Sonia Bergamaschi, Themis Palpanas, Manolis Koubarakis (November 21, 2020), “*Three-Dimensional Entity Resolution with JedAI*”, National and Kapodistrian University of Athens, Greece.
- [3] Marco Franke, Konstantin Klein, Karl A. Hribernik and Klaus-Dieter Thoben (February 2014). "Identification of Interface Information for a Virtual Data .Integration".
- [4] Giovanni Simonini, George Papadakis, Themis Palpanas, and Sonia Bergamaschi (May 15, 2019), "Schema-agnostic Progressive Entity Resolution"
- [5] Yasin N. Silva, Spencer S. Pearson, Jaime Chon, Ryan Roberts (2015). Similarity Joins: Their implementation and interactions with other database operators. Information Systems (Vol 52). page 149-162
- [6] Wang, H., Yang, L. & Xiao, Y. SETJoin: a novel top-k similarity join algorithm. Soft Computing 24, 14577–14592 (2020)
- [7] T. Bocek, E. Hunt, and B. Stiller. Fast similarity search in large dictionaries. Technical Report ifi-2007.02, University of Zurich, Department of Informatics, 2007