Block Building:

Entity Resolution costituisce un core per il data integration, e a causa della complessità quadratica scala su grandi dataset attraverso metodi di blocking.

I metodi di blocking possono essere configurati in due modi:

* Schema based : La configurazione basata sullo schema si basa sulle informazioni dello schema informativo per selezionare firme di elevata riconoscibilità e bassa rumorosità,
* schema-agnostico tratta ogni token da tutti gli attributi

come firma

ER ha l’obiettivo di identificare tutti i profili di entità che corrispondono alle stesse entità del mondo reale.

Con il blocking clusterizza entità di profili simili in blocchi e poi fa una comparazione a coppie dentro ogni blocco .

Da ogni entità profilo i metodi di blocking estraggono uno o + chiavi di blocking che riassumono un sottoinsieme dei valori dei suoi attributi.

Conseguentemente profili con chiavi di blocking simile sono messe nello stesso blocco.

* Schema based: sfrutta lo schema di informazione a priori per identificare rilevanti chiavi di blocking che riassumono il contenuto degli attributi con maggiori valori distintivi 🡪 crea blocchi che coinvolgono poche comparazioni mentre la maggior parte dei duplcati viene messa in un blocco comune almeno.
* Schema agnostic più comparazioni nell’esempio.

Schema based configuration 🡪 blocking method such as: Sorted Neighborhood , Suffix Arrays che sono creati per la raccolta di dati omogenei. Queste collezioni si riferiscono a dati strutturati, che si attengono a uno schema specifico con caratteristiche semantiche e qualitative note per ogni attributo.

Schema agnostico🡪 blocking method: Token Blocking raccolgono collezioni di dati eterogenei. Queste collezioni contengono dati semi / non strutturati con rumore nei valori degli attributi e nei nomi degli attributi.

Dubbio : qui fa test su dati omogeni ma noi non li abbiamo ? -.-

**Misure di valutazione:**

M1(Recall) ; M2(Precision);M3;M4

Obiettivo: teoricamente di max in numero di duplicati trovati e minimazzare il numero di blocchi . Trade off Precision e Recall.

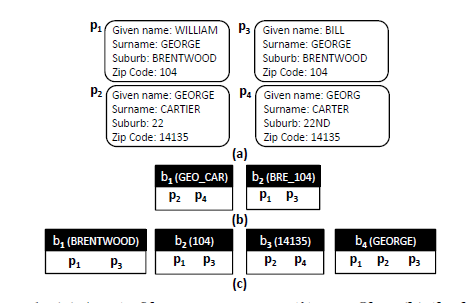
Blocking per dati strutturati:

* **Standard Blocking** : rappresenta ogni entità da una o + chiavi e crea blocchi sulla loro uguaglianza

cioè, ogni blocco corrisponde ad un chiave specifica e contiene tutte le entità che l'hanno nella loro rappresentazione. (si fa nello schema based / agnostic (token blocking))

* **Sorted Neighborhood**: usa le stesse chiavi di blocco dello standard blocking ma costruisce i blocchi sulla loro similarità invece che uguaglianza: ordina le chiavi di blocking in ordine alfabetico e

scorre una finestra di dimensione fissa w sopra l'elenco delle entità ordinate.



In questo esempio b (schema based) la lista ordinata chiavi sarà : {BRE 104, GEO CAR} and of entities {p1,p3,p2,p4}

Thus, for a window of size w = 2, we get the blocks b1 = {p1}; p3},

b2 = {p3; p2}, b3 = {p2; p4}

La schema based di solito utilizza definizioni di chiavi multiple e forma una lista separata ordinata per ognuno di loro. La configurazione schema-agnostic utilizza una lista unica di chiavi ordinate che comprende tutti i token nei valori di tutte le entità. Per entrambe le configurazioni, le entità con la stessa chiave sono posizionate nella lista ordinati in ordine casuale.

* **Extended Sorted Neighborhood**:

obiettivo di indirizzare il problema del vicinato ordinato che si basa sulla dimensione della finestra . Piccole finestre abbassano la recall ma grandifinestre abbassano la precision e fanno + comparazioni.

Tale tecnica offre performance + robusta facendo scorrere la finestra sulle chiavi ordinate invece delle entità ordinate: nell’esempio in figuara con una finetra d 2 crea un singolo blocco per le due chiavi di blocking b1{p1,p2,p3,p4}. Schema based configuration impiega differenti liste per ogni definizione di chiave di blocking mentre agnostico usa un'unica lista di chiavi o tokens ordinati.

* **Q-grammi blocking** : Si basa sullo Standard Blocking ma transforma loro in un formato che è resistente al rumore. Invece di considerare l’intera chiave considera i q-grammi a costruisce i blocchi sulla loro uguaglianza. Es q=3 la chiave di blocco BRE\_104 🡪 BRE,RE\_,E\_1, E COSI VIA . Nello schema based si applica a tutte le chiavi estratte da specifici omi di attributi mentre per l’agnostico si applica a tutti i valori dei token di tutte le entità . In entrambi i casi un nuovo blocco è creato per ogni q-gram che appare in almeno due entità .
* **Extended Q-grammi blocking**: migliora la tecnica di base producendo chiavi di blocking ad alta discriminazione. Al fine di diminuire il numero di comparazioni di coppie nei blocchi risultanti senza mancare nessuno dei duplicati individuati. Invece dei q-grammi individuali , si considera considera combinazioni che derivano dalla concatenazione di almeno L q-grammi. L deriva da una soglia con una formula : L=max(1; bk \_ Tc), where k is the number of q-grams in the original blocking key.

T è definito nell'intervallo [0; 1), con valori maggiori che riducono il numero di combinazioni. Blocchi sono creati per ogni chiave che appare in almeno due entità.

* **Canopy clustering** : usa le stesse chiavi di Qgrammi blocking ma crea blocchi sulla loro similarità invece di uguaglianza. Inizialmente mette tutte le entità in un inseme di candidati P . In ogni iterazione rimuove random il rumore da P e compara le sue chiavi di blocking(q-grammi ) con tutte le altre entità in P. quelle entità con una similarità alta rispetto a w1 sono messe in un nuovo blocco insieme al rumore p i mentre quelle che eccedono un'altra soglia sono rimosse dall insieme.
* **Extended Canopy Clustering** : si rivolge al principale inconveniente del Canopy Clustering, ovvero la sua sensibilità ai valori delle soglie di peso: se w1 è troppo alto, molte entità non saranno posizionate in nessun blocco. Per evitare di perdere qualsiasi entità, ECaCl sostituisce

w1 e w2 con due soglie di cardinalità, n1 e n2, tali che

1<=n2<=n1. Simile al CaCl, inizialmente mette tutte le entità in un pool

P e gli rimuove in modo iterativo un pi rumore. In ogni iterazione,

confronta il pi con tutte le entità ancora in P utilizzando le chiavi di blocco

definite da Q-grams Blocking. Poi, invece di usare il peso

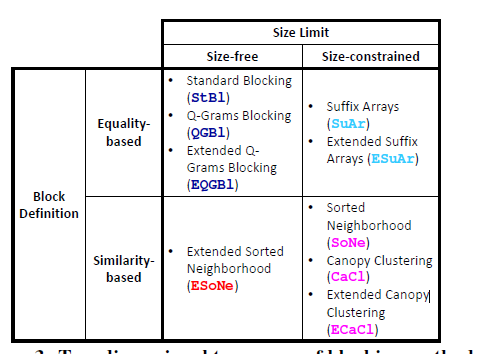
inserisce in un nuovo blocco gli n1 più vicini in un nuovo blocco - insieme

con pi - e rimuove il n2 vicini vicinato a P.

* **Suffix Arrays** migliora la tolleranza al rumore dello standard blocking applicando altre trasformazioni alle sue chiavi. Le converte in suffissi che sono piu lunghi di una minima lunghezza m. es lm = 5 🡪 BRE 104, RE 104 and E 104

Ricordandoci che schema agnostic lo fa ai token di tutti i valori degli attributi. Blocchi creati su uguaglianza delle chiavi di blocco.

* **Extended suffix arrays**: migliora la robustezza sostenendo il rumore alla fine delle chiavi di blocco . Converte ogni chiave dello standard blocking in tutti le sottostringhe che sono + grandi della lunghezza minima lm.



Sperimentazioni : agnostico con stb1 e eqgb1