40TH INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING

Recent Advances in Entity Resolution

Bing Li, Yaoshu Wang, and Wei Wang

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

SINGAPORE Agency for Science, Technology and Research

VENI VIDI SICS VICI 深圳计算科学研究院 Shenzhen Institute of Computing Sciences

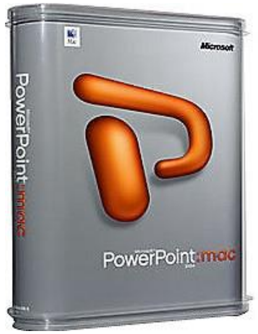
UNSW SYDNEY

What is Entity Resolution?

- **Entity Resolution**: Problem of identifying co-referent manifestations that refer to the same real-world entity from different data sources.
- Examples of co-referent manifestations:
  - Different descriptions of a same product on different e-commerce websites (e.g., Google shopping, amazon)

Amazon

DESCRIPTION	MANUFACTURER	PRICE
powerpoint 2004 mac by microsoft	microsoft	229.99



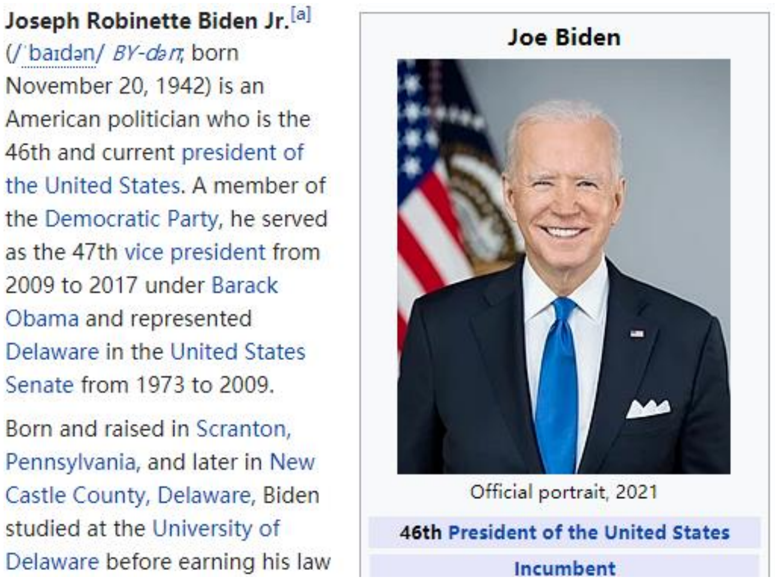
Google Shopping

TITLE	PRICE
microsoft powerpoint 2004 mac apple	228.95

What is Entity Resolution?

- **Entity Resolution**: Problem of identifying co-referent manifestations that refer to the same real-world entity from different data sources.
- Examples of co-referent manifestations:
  - Web pages with differing descriptions of the same person.

https://en.wikipedia.org/wiki/Joe_Biden



Joseph Robinette Biden Jr.[a] (/ˈbaɪdən/ *BY-dən*; born November 20, 1942) is an American politician who is the 46th and current president of the United States. A member of the Democratic Party, he served as the 47th vice president from 2009 to 2017 under Barack Obama and represented Delaware in the United States Senate from 1973 to 2009.

Born and raised in Scranton, Pennsylvania, and later in New Castle County, Delaware, Biden studied at the University of Delaware before earning his law

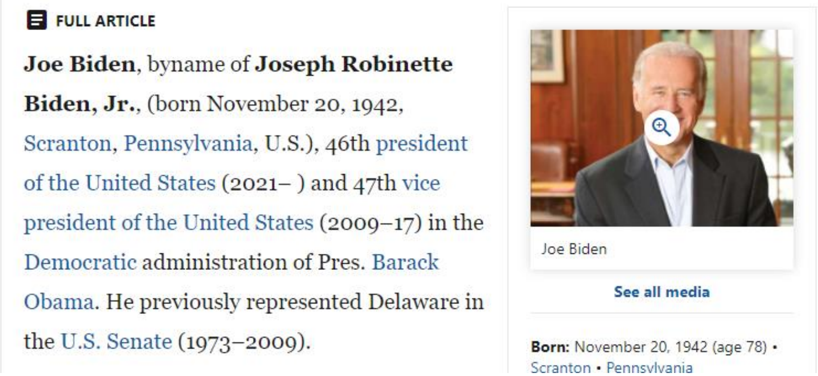
Joe Biden

Official portrait, 2021

46th President of the United States

Incumbent

https://www.britannica.com/



FULL ARTICLE

Joe Biden, byname of **Joseph Robinette Biden, Jr.**, (born November 20, 1942, Scranton, Pennsylvania, U.S.), 46th president of the United States (2021–) and 47th vice president of the United States (2009–17) in the Democratic administration of Pres. Barack Obama. He previously represented Delaware in the U.S. Senate (1973–2009).

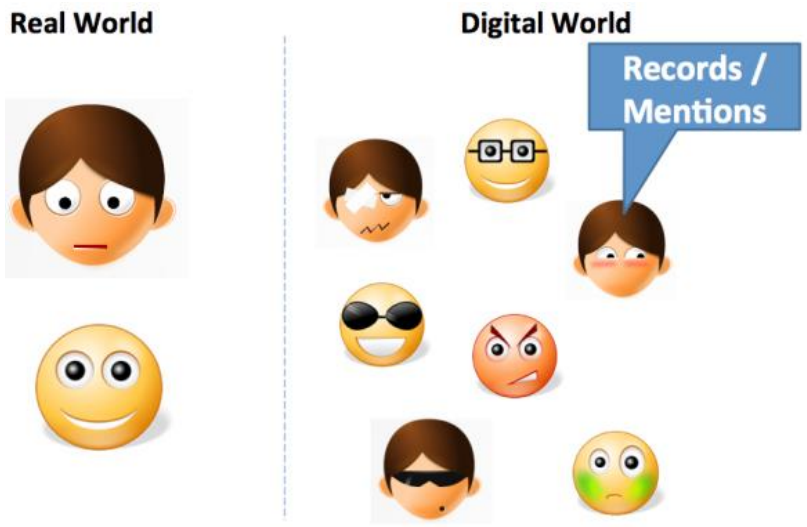
Joe Biden

See all media

Born: November 20, 1942 (age 78) • Scranton • Pennsylvania

What is Entity Resolution?

- **Entity Resolution**: Problem of identifying co-referent manifestations that refer to the same real-world entity from different data sources.
- Examples of co-referent manifestations:
  - Different photos of the same object.



Ironically, Entity Resolution has many duplicate names

Record linkage

Duplicate detection

Coreference resolution

Reference reconciliation

Fuzzy match

Object consolidation

Object identification

Deduplication

Entity clustering

Approximate match

Identity uncertainty

Merge/purge

Household matching

Hardening soft databases

Householding

Reference matching

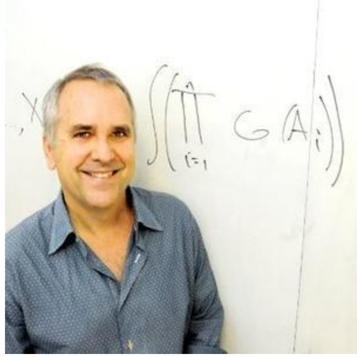
Doubles

http://www.cs.umd.edu/~getoor/Tutorials/ER_VLDB2012.pdf

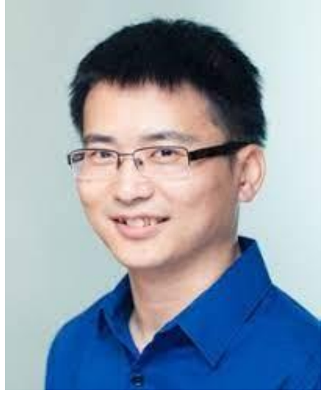
Why is Entity Resolution Hard?

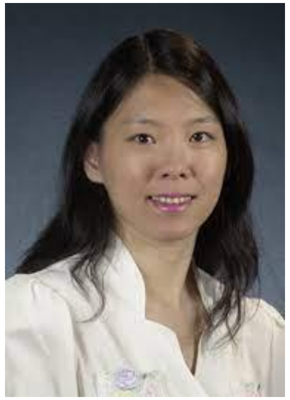
- Heterogeneity everywhere
    - Name/Attribute ambiguity

Michael Jordan



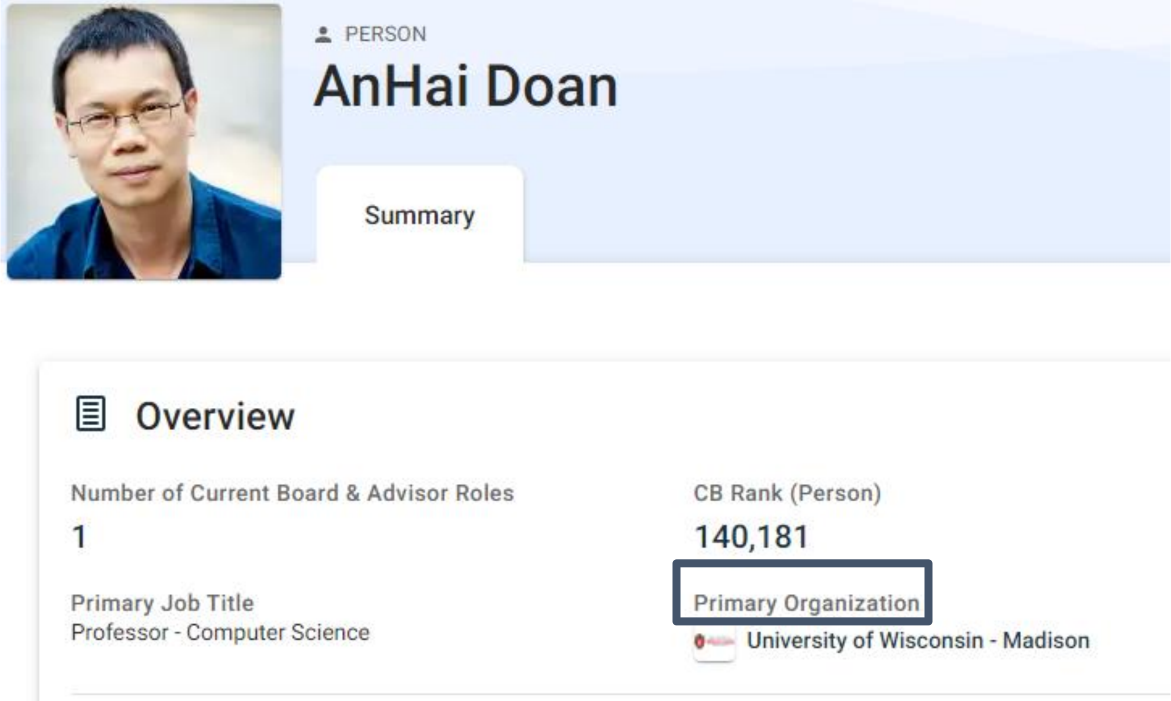
Prof. Wei Wang





Why is Enity Resolution Hard?

- Heterogeneity everywhere
  - Changing attribute names



PERSON

AnHai Doan

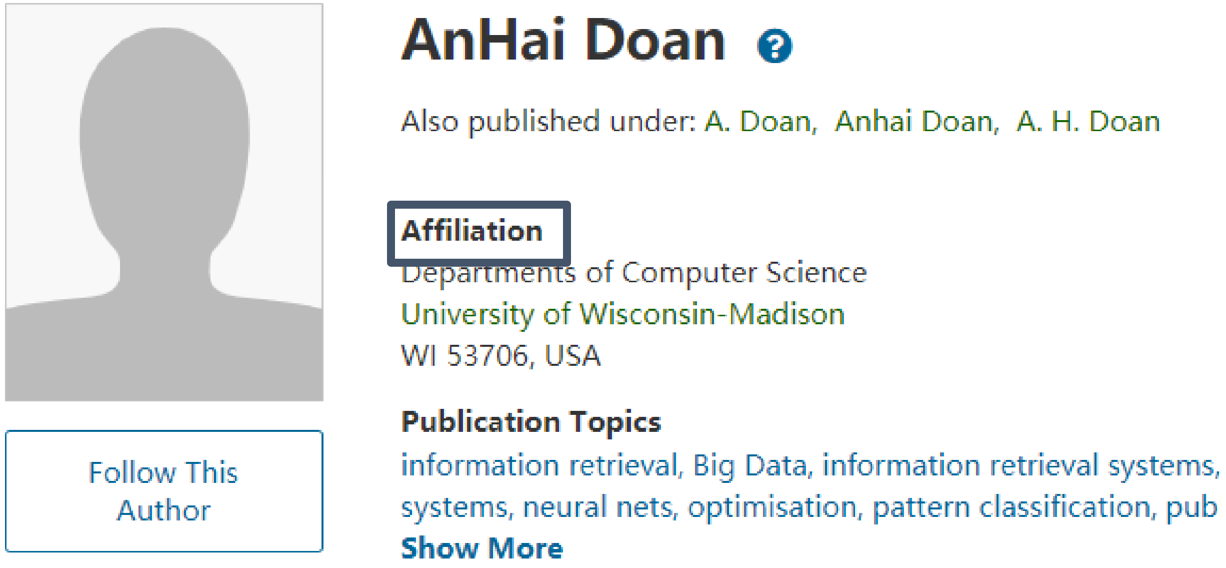
Overview

Number of Current Board & Advisor Roles
1

CB Rank (Person)
140,181

Primary Job Title
Professor - Computer Science

Primary Organization
University of Wisconsin - Madison



AnHai Doan

Also published under: A. Doan, Anhai Doan, A. H. Doan

Affiliation
Departments of Computer Science
University of Wisconsin-Madison
WI 53706, USA

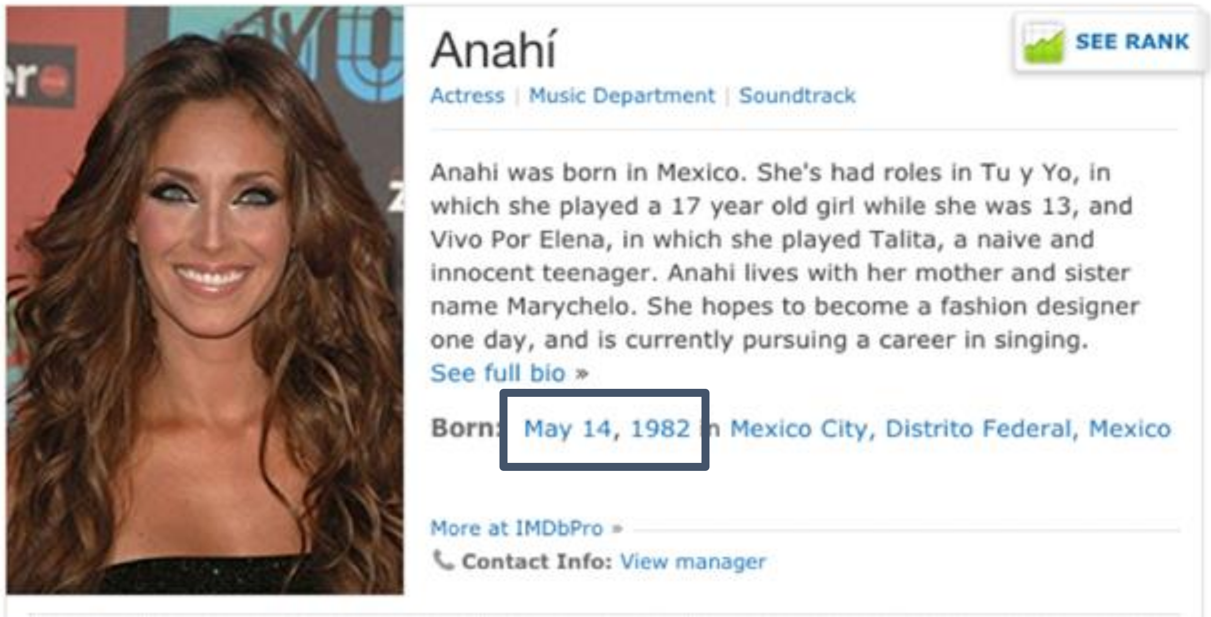
Publication Topics
information retrieval, Big Data, information retrieval systems, systems, neural nets, optimisation, pattern classification, pub

Example by Xin Luna Dong

Why is Entity Resolution Hard?

- Heterogeneity everywhere
  - Conflicting and erroneous values

IMDB



Anahí

Actress | Music Department | Soundtrack

Anahi was born in Mexico. She's had roles in Tu y Yo, in which she played a 17 year old girl while she was 13, and Vivo Por Elena, in which she played Talita, a naive and innocent teenager. Anahi lives with her mother and sister name Marychelo. She hopes to become a fashion designer one day, and is currently pursuing a career in singing.

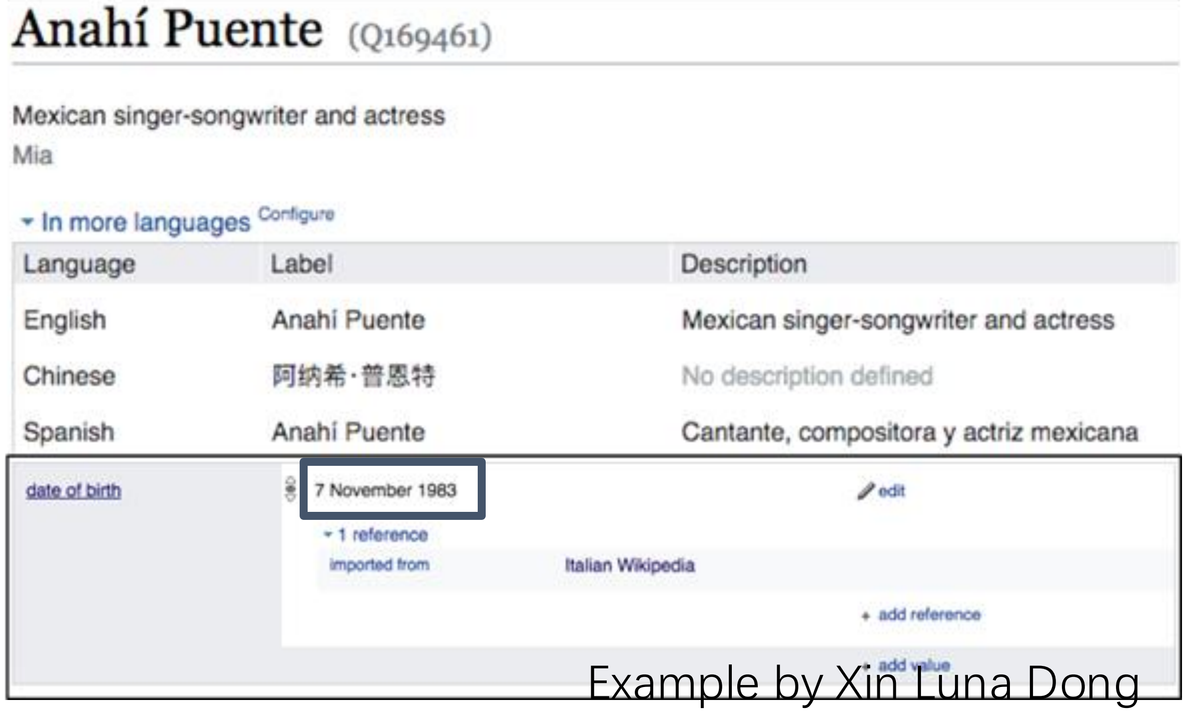
See full bio »

Born: May 14, 1982 in Mexico City, Distrito Federal, Mexico

More at IMDbPro »

Contact Info: View manager

WikiData



Anahí Puente (Q169461)

Mexican singer-songwriter and actress

Mia

▾ In more languages Configure

Language	Label	Description
English	Anahí Puente	Mexican singer-songwriter and actress
Chinese	阿纳希·普恩特	No description defined
Spanish	Anahí Puente	Cantante, compositora y actriz mexicana

date of birth — 7 November 1983 — edit

▾ 1 reference

imported from — Italian Wikipedia

+ add reference

+ add value

Example by Xin Luna Dong

Why is Entity Resolution Hard?

- Heterogeneity everywhere
  - Missing values

Google

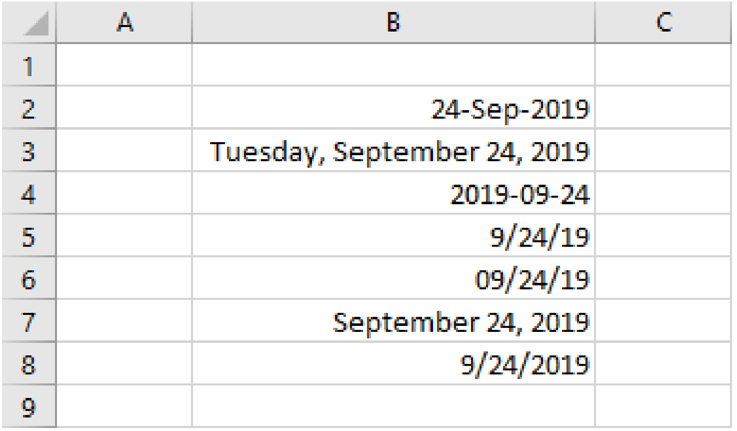
TITLE	MANUFACTURER	PRICE
microsoft powerpoint 2004 mac apple	--	228.95
microsoft powerpoint 2004 for mac upgrade	microsoft	97.99

Amazon

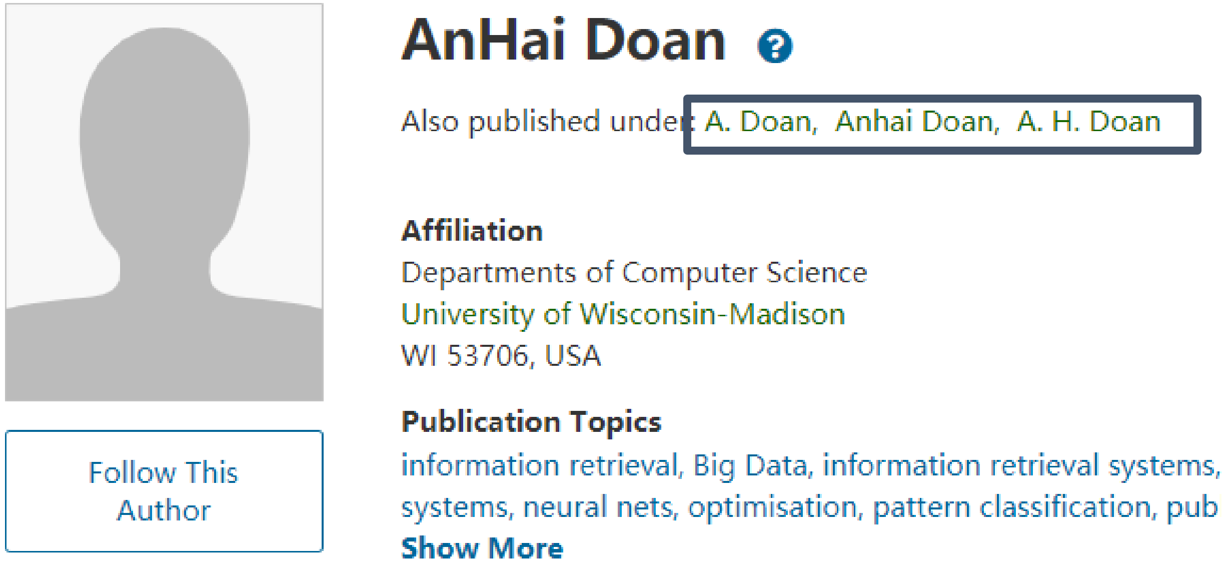
DESCRIPTION	MANUFACTURER	PRICE
powerpoint 2004 mac by microsoft	microsoft	229.99
powerpoint 2004 upgrade mac	microsoft	109.99

Why is Entity Resolution Hard?

- Heterogeneity everywhere
  - ○ Different value formatting



	A	B	C
1			
2		24-Sep-2019	
3		Tuesday, September 24, 2019	
4		2019-09-24	
5		9/24/19	
6		09/24/19	
7		September 24, 2019	
8		9/24/2019	
9			



AnHai Doan

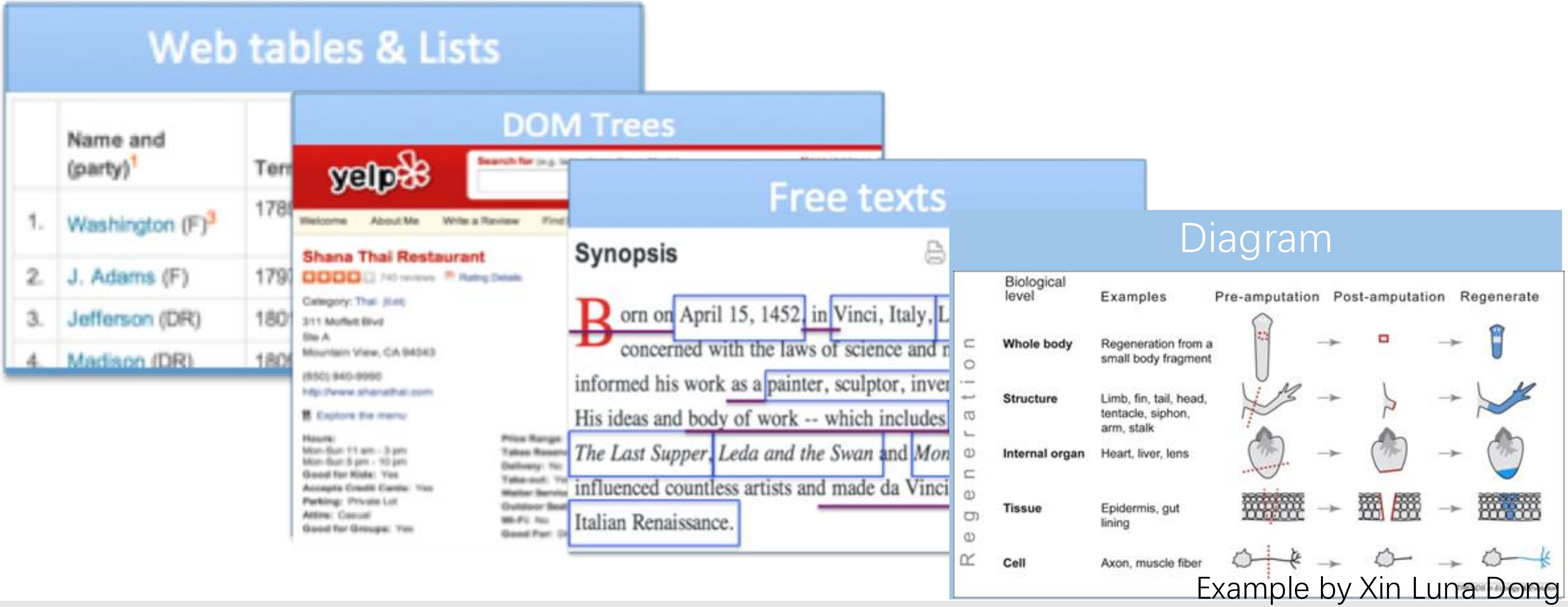
Also published under A. Doan, Anhai Doan, A. H. Doan

Affiliation
Departments of Computer Science
University of Wisconsin-Madison
WI 53706, USA

Publication Topics
information retrieval, Big Data, information retrieval systems, systems, neural nets, optimisation, pattern classification, pub

Why is Entity Resolution Hard?

- Heterogeneity everywhere
  - Different data types



	Name and (party)[1]	Ter
1.	Washington (F)[3]	178
2.	J. Adams (F)	179
3.	Jefferson (DR)	180
4.	Madison (DR)	180

Shana Thai Restaurant

740 reviews Rating Details

Category: Thai [Edit]

311 Moffett Blvd
Ste A
Mountain View, CA 94043

(650) 940-9990
http://www.shanathai.com

Explore the menu

Hours:
Mon-Sun 11 am - 3 pm
Mon-Sun 5 pm - 10 pm
Good for Kids: Yes
Accepts Credit Cards: Yes
Parking: Private Lot
Attire: Casual
Good for Groups: Yes

Price Range:
Takes Reserv
Delivery: No
Take-out: Ye
Waiter Servic
Outdoor Seat
Wi-Fi: No
Good For: D

Synopsis

Born on April 15, 1452, in Vinci, Italy, L
concerned with the laws of science and n
informed his work as a painter, sculptor, inve
His ideas and body of work -- which includes
The Last Supper, *Leda and the Swan* and *Mon*
influenced countless artists and made da Vinci
Italian Renaissance.

Biological level	Examples	Pre-amputation	Post-amputation	Regenerate
Whole body	Regeneration from a small body fragment			
Structure	Limb, fin, tail, head, tentacle, siphon, arm, stalk			
Internal organ	Heart, liver, lens			
Tissue	Epidermis, gut lining			
Cell	Axon, muscle fiber			

Example by Xin Luna Dong

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

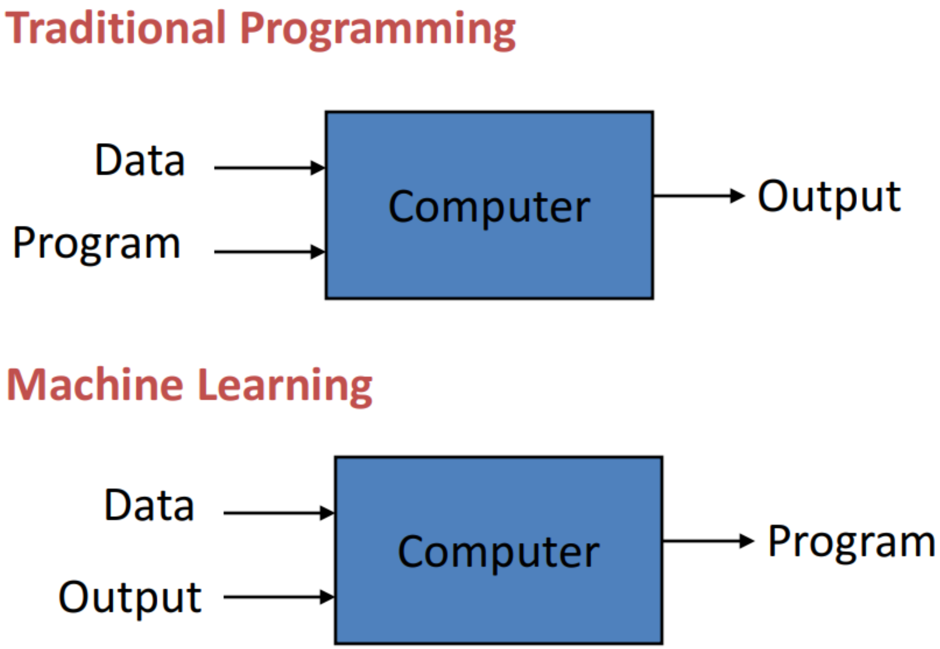
- Herbert Simon

Definition by Tom Mitchell:

Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E

A well-defined learning task is given by <P, T, E>

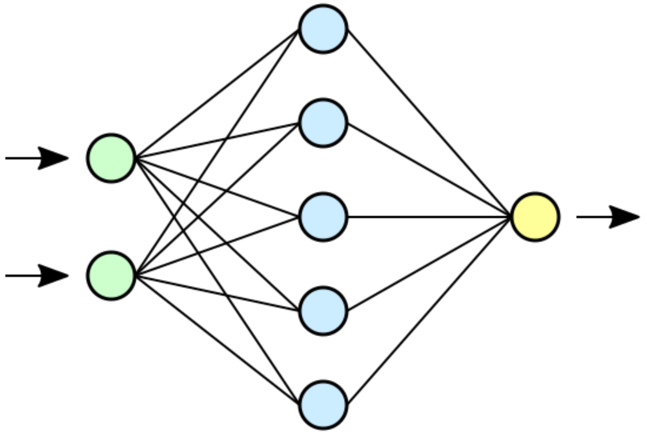


Based on slides by Eric Eaton

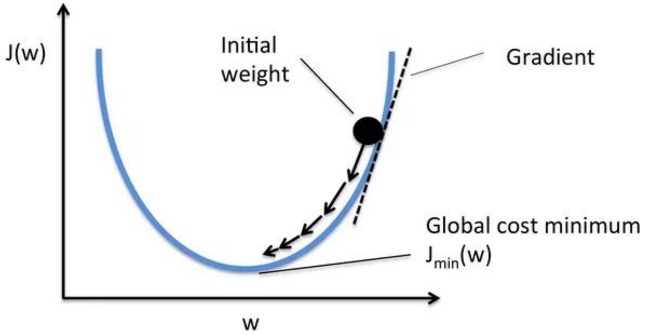
What is Deep Learning?

Deep Learning – Extract patterns from Data using Neural Networks

- *Model*
  - CNN, RNN, LSTM, Transformer
- *Objective*
  - Cross-entropy, L2 Loss, Hinge-loss
- *Optimization*
  - SGD, Adam, AdamW



$$J(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{y^{(i)}}_{\text{Actual}} - \underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}}\Big)^2$$



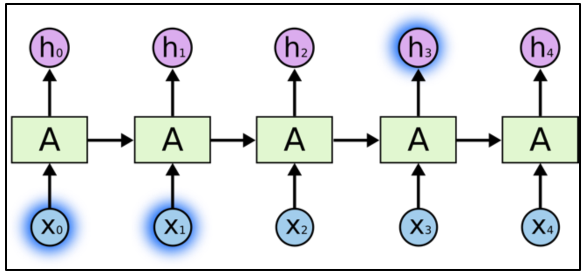
Why Deep Learning Help?

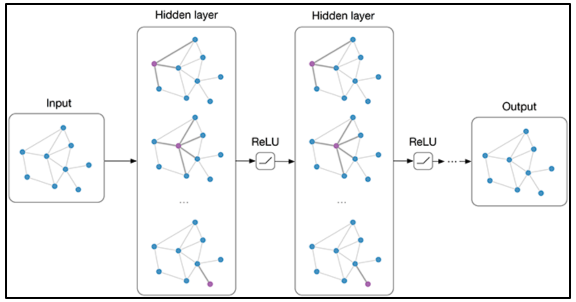
Deep learning models could

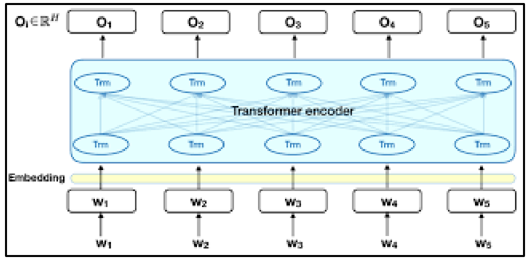
- Bridge vocabulary mismatch
  - Different value formatting or Changing attribute names
    - *E.g., AnHai Doan – A. Doan – A.H. Doan; Affiliation – Primary organization*
- Represent data in an unified vector space
  - Different data types
    - *E.g., Multimodality: image – free text – table*
- Capture contextual information
  - Name/Attribute ambiguity
    - *E.g., Prof. Wei Wang – UKUST; Prof. Wei Wang – UCLA*
- Better Generalization
  - Conflicting and erroneous values
  - Missing values

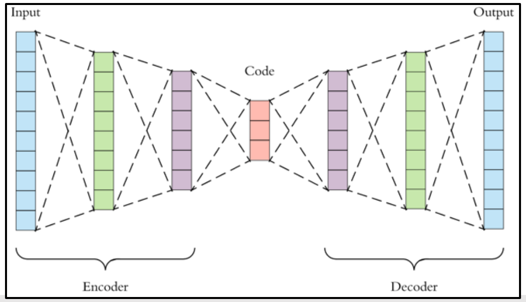
What Deep Learning Model is Used in ER?

Deep Learning Model	*LSTM*	DeepMatcher [SIGMOD'18]	DeepER [VLDB'18]	
	GCN	GraphER [AAAI'20]		
	Transformer-based LMs	BERT-ER[AAAI'21]	DITTO [VLDB'21]	Sbert [EMNLP'19]
	VAE	VAER [ICDE'21]	VAR-Siamese [NIPS'18]	Autoencoder / Trans-encoder [VLDB'21]
	Ensemble	RISK [JMLR'21]		









A Brief History of Entity Resolution

Rule-based

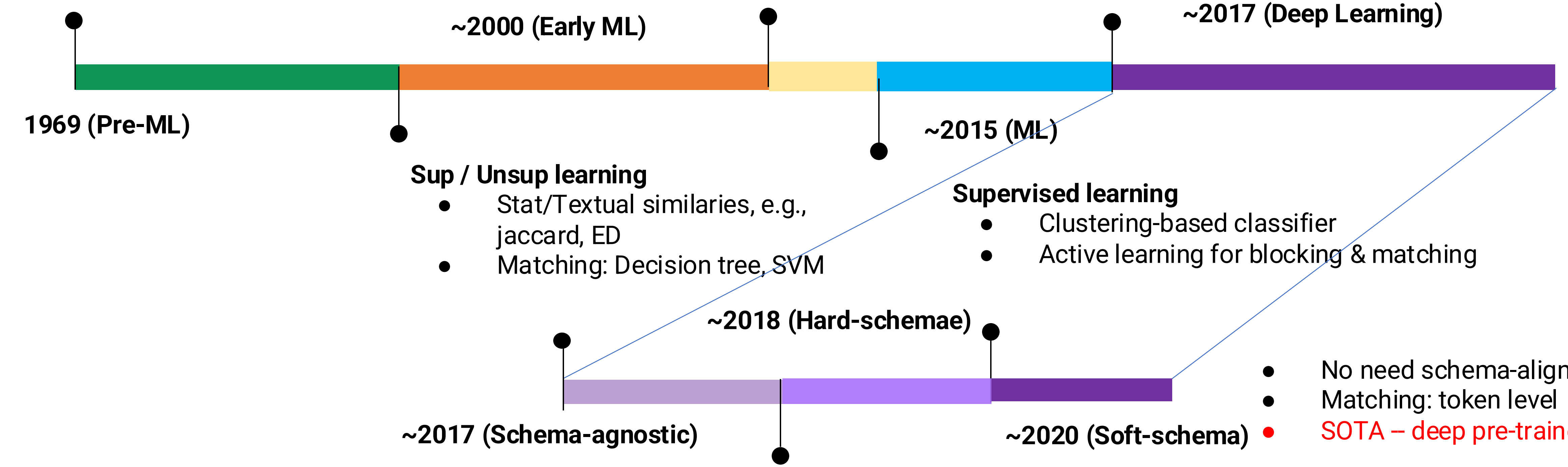
- Declarative matching rules
  - Pre-defined or synthesized
  - Blocking: static keys, e.g., same name

Crowd-sourcing

- Matching: manually annotate tuples

Deep learning

- Deep neural models
- Attribute embedding
- Blocking: Hashing-based



1969 (Pre-ML)

~2000 (Early ML)

~2012 (Crowd-sourcing)

~2015 (ML)

~2017 (Deep Learning)

Sup / Unsup learning

- Stat/Textual similaries, e.g., jaccard, ED
- Matching: Decision tree, SVM

Supervised learning

- Clustering-based classifier
- Active learning for blocking & matching

~2017 (Schema-agnostic)

~2018 (Hard-schemae)

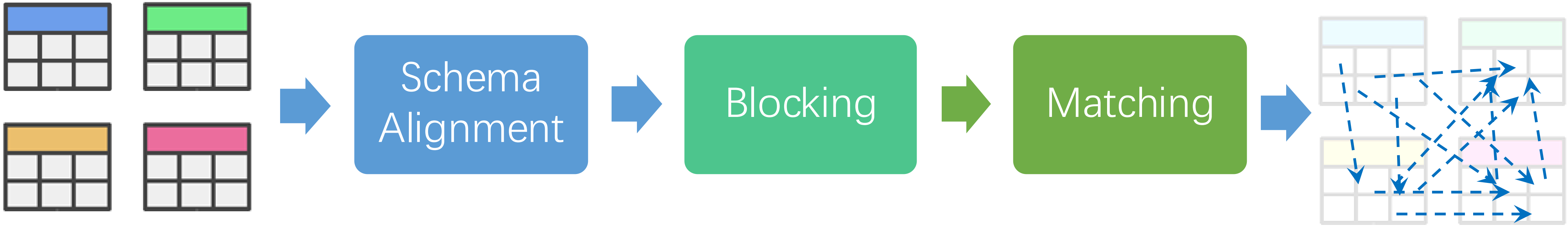
~2020 (Soft-schema)

- No need schema-alignment
- Matching: token level
- SOTA -- deep pre-trained LMs

- Text matching problem

- Need schema-alignment
- Matching: attribute or entity level

Quick Tour for Entity Resolution



Schema Alignment

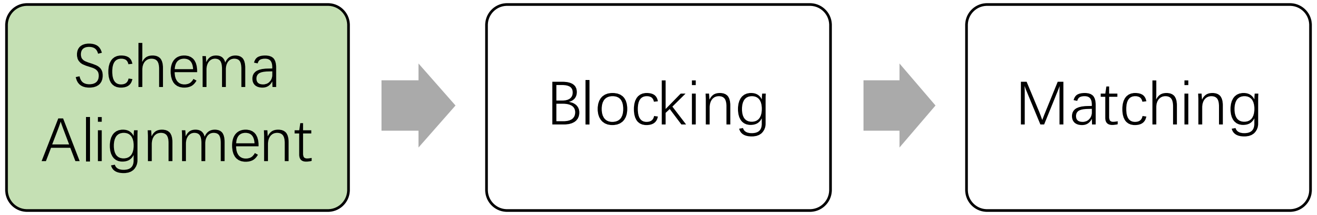
Blocking

Matching

Data from different sources
(Structural tables, Raw Text, HTML)

Co-referent relations

Quick Tour for Entity Resolution



Schema Alignment

Blocking

Matching

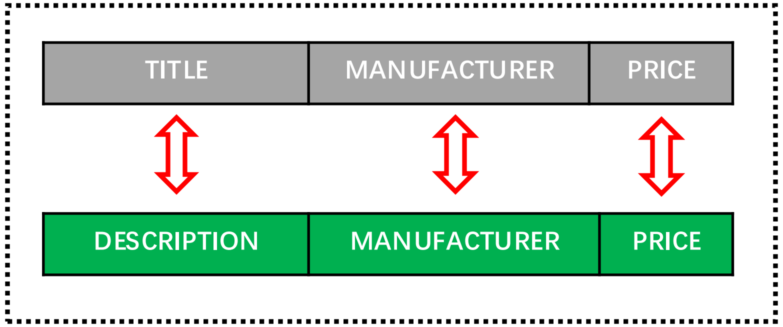
- Generate a mediate schema

Google

TITLE	MANUFACTURER	PRICE
microsoft powerpoint 2004 mac apple	--	228.95

Amazon

DESCRIPTION	MANUFACTURER	PRICE
powerpoint 2004 upgrade mac	microsoft	109.99

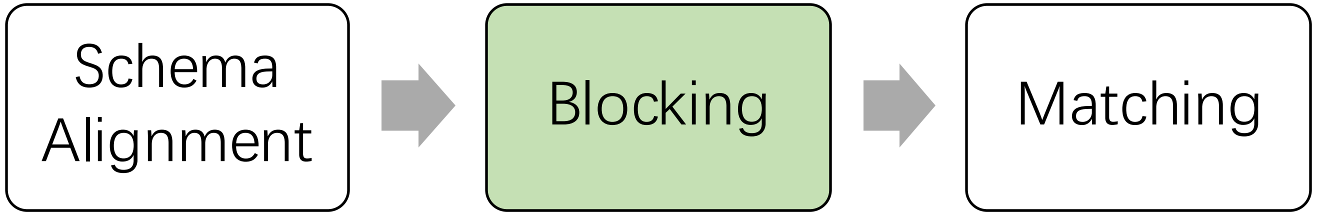


TITLE MANUFACTURER PRICE

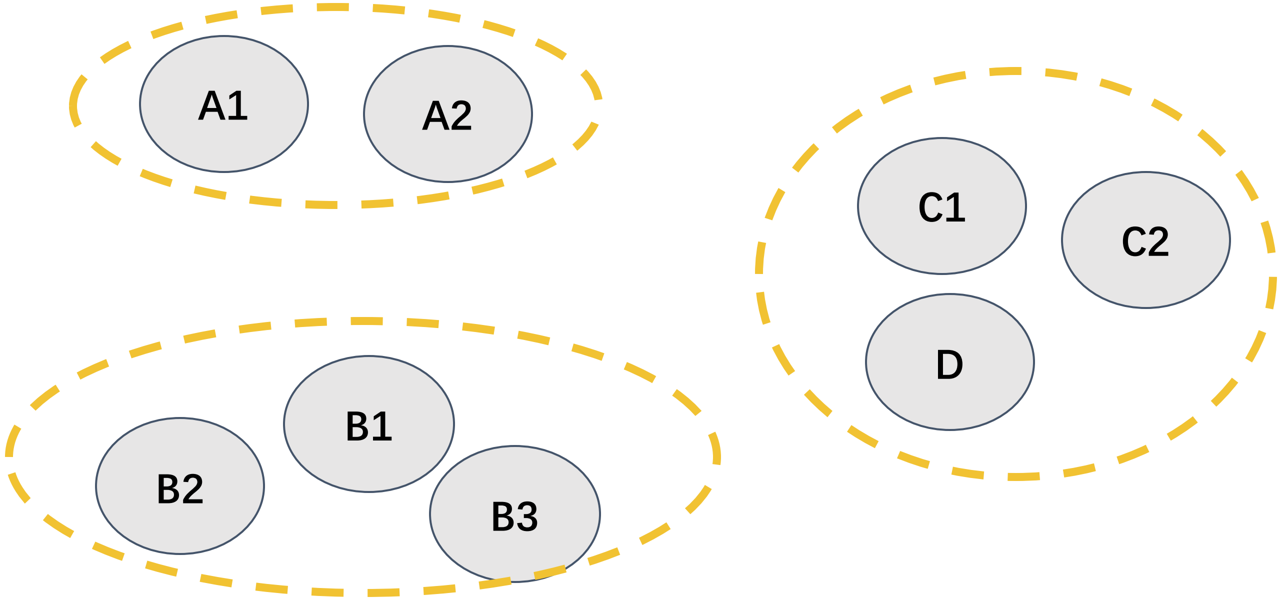
DESCRIPTION MANUFACTURER PRICE

Schema mapping

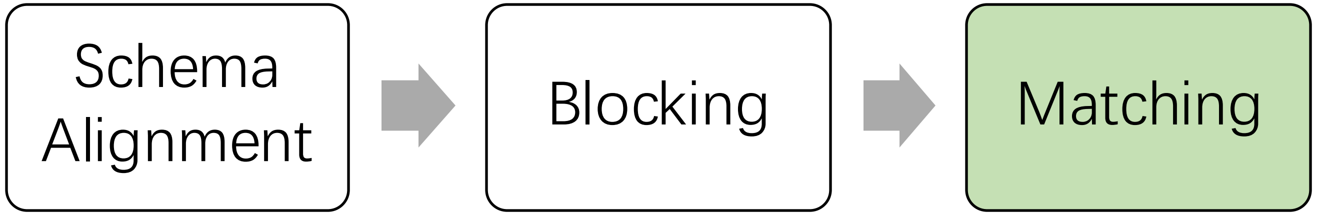
Quick Tour for Entity Resolution



- Grouping tuple pairs into blocks (or top-k ranking)
    - Avoid unnecessary matching between obviously dissimilar pairs



Quick Tour for Entity Resolution

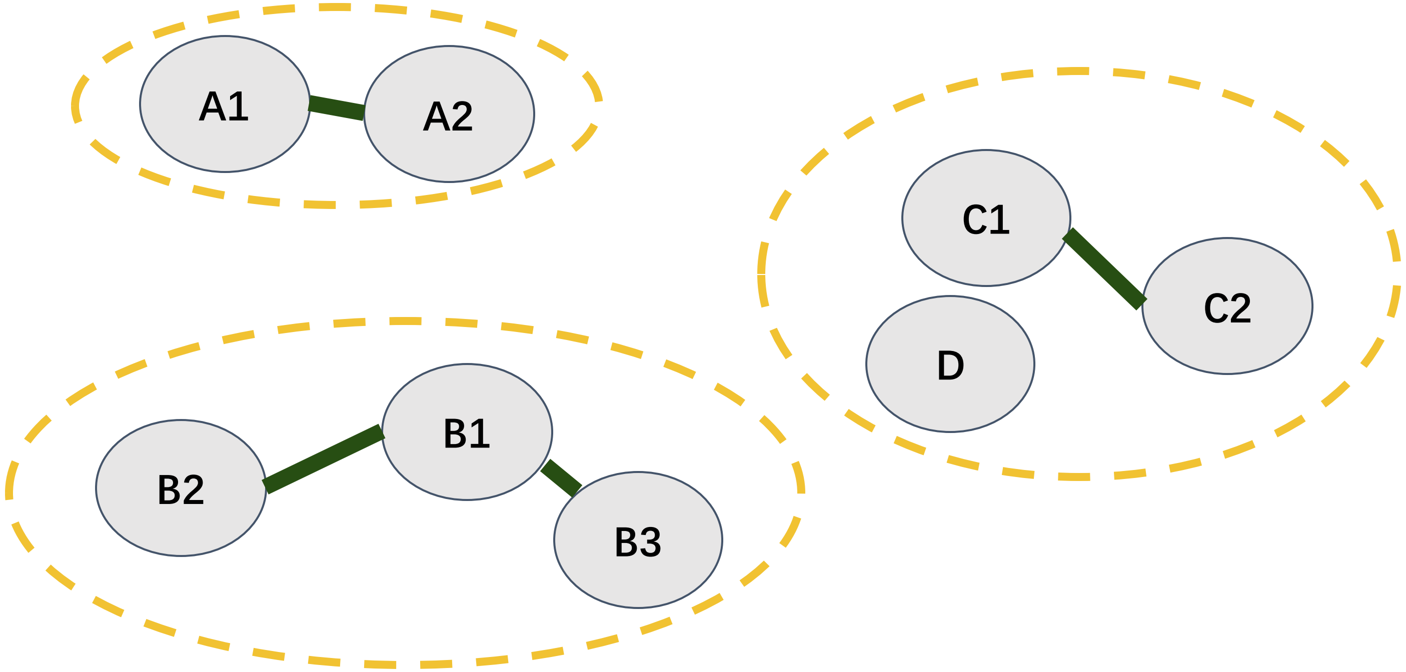


Schema Alignment

Blocking

Matching

- Find co-references within each blocks



Entity Blocking – Problem Definition

- **Problem Definition**: Given two relational tables A and B with the same schema, find all tuple pairs ($a \in A$, b $\in B$) that match, i.e., refer to the same real-world entity. (R-S join)
- **Evaluation**
  - **Efficiency**
    - Pairs Quality (PQ) or precision

      $$\mathrm{PQ} = \frac{|\,\mathrm{TruePair(Cand)}\,|}{|\,\mathrm{Cand}\,|}$$

    - Reduction Ratio (RR)

      $$\mathrm{RR} = 1 - \frac{|\,\mathrm{Cand}\,|}{|\,\mathrm{A}\,| \times |\,\mathrm{B}\,|}$$

    - Running Time
  - **Effectiveness**
    - Pair Completeness (PC) or recall

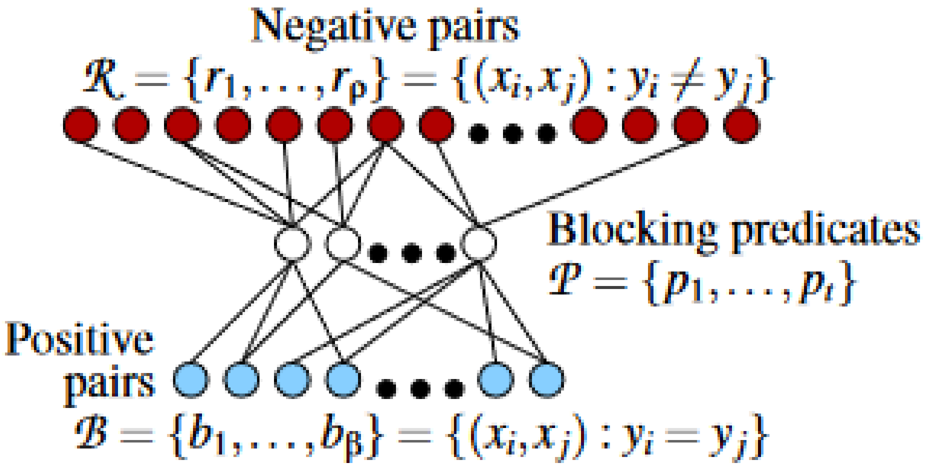
      $$\mathrm{PC} = \frac{|\,\mathrm{TruePair(Cand)}\,|}{|\,\mathrm{TruePair(A \infty B)}\,|}$$

Entity Blocking - Overview

- **Non-learning methods**
  - Baseline: Hash-based, sort-based, size-based similarity-based, etc.
  - Improved: meta-blocking, rule-based (e.g., MD), etc.
- **Learning methods** (**Our main focus**)
  - Learning rules: ApproxDNF, BSL, Fisher, etc.
  - Learning no-DL model: CBLOCK, Smurf, Supervised meta-block, etc.
  - Learning representations: DeepER, autoencoder, etc.
  - Learn to hash: BERT-ER

Entity Blocking – ApproxDNF [Bilenko et al., ICDM’06]

- Rule-based learning
- **Schema-aware**
- Disjunctive Normal Form (DNF) blocking
- Rely on **predefined predicates**, e.g., Jaccard, Same n First Chars, exact match, n-gram, etc.
- **Red-Blue Set Cover**
- **Smaller reduction ratio and recall** than unlearned basesline.



Negative pairs
$\mathcal{R} = \{r_1, \ldots, r_\rho\} = \{(x_i, x_j) : y_i \neq y_j\}$
Blocking predicates
$\mathcal{P} = \{p_1, \ldots, p_t\}$
Positive pairs
$\mathcal{B} = \{b_1, \ldots, b_\beta\} = \{(x_i, x_j) : y_i = y_j\}$

$$w^* = \operatorname*{argmin}_{w} \sum_{(x_i,x_j)\in\mathcal{R}} [\![w^T p(x_i,x_j) > 0]\!]$$

$$\text{s.t.} \quad |\mathcal{B}| - \sum_{(x_i,x_j)\in\mathcal{B}} [\![w^T p(x_i,x_j) > 0]\!] < \varepsilon$$

$$w \text{ is binary}$$

Entity Blocking – BSL [Michelson et al., AAAI'06], BSL$^+$ [Cao et al. IJCAI11]

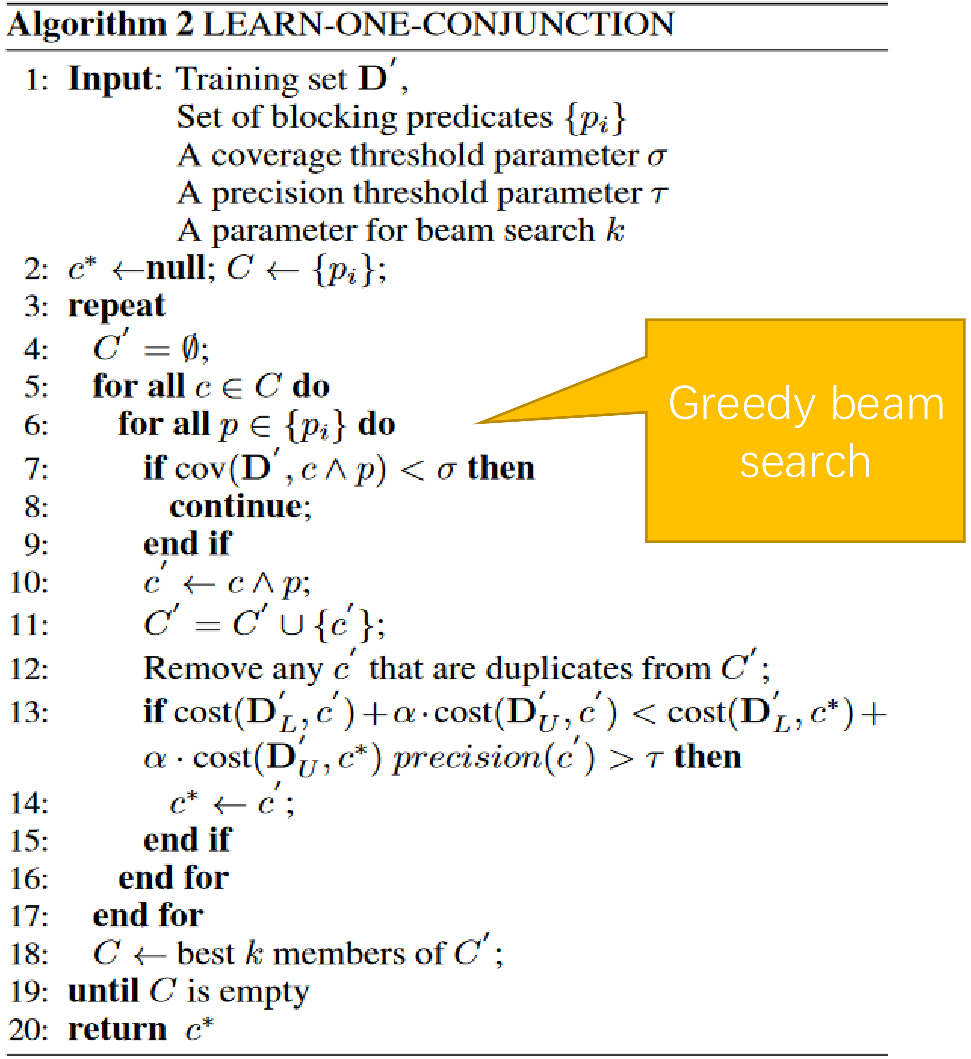
- Disjunctive Normal Form (DNF)
- **Schema-aware**
- Rely on **predefined predicates**
- Set Cover problem

Incorporate unlabeled data

$$\underset{h_P}{\arg\min} \quad \mathrm{cost}(\mathbf{D}_L^x, P) + \boxed{\alpha \cdot \mathrm{cost}(\mathbf{D}_U, P)} \qquad (1a)$$

$$\text{subject to} \quad \mathrm{cov}(\mathbf{D}_L, P) > 1 - \epsilon \qquad (1b)$$

- **Obj:** Minimize RR (using labeled and unlabeled data)
- **Cond**: Recall is above a threshold



Algorithm 2 LEARN-ONE-CONJUNCTION

```
Input: Training set $\mathbf{D}'$,
       Set of blocking predicates $\{p_i\}$
       A coverage threshold parameter $\sigma$
       A precision threshold parameter $\tau$
       A parameter for beam search $k$
$c^* \leftarrow$ null; $C \leftarrow \{p_i\}$;
repeat
  $C' = \emptyset$;
  for all $c \in C$ do
    for all $p \in \{p_i\}$ do
      if cov$(\mathbf{D}', c \wedge p) < \sigma$ then
        continue;
      end if
      $c' \leftarrow c \wedge p$;
      $C' = C' \cup \{c'\}$;
      Remove any $c'$ that are duplicates from $C'$;
      if cost$(\mathbf{D}'_L, c') + \alpha \cdot$ cost$(\mathbf{D}'_U, c') <$ cost$(\mathbf{D}'_L, c^*) +$
      $\alpha \cdot$ cost$(\mathbf{D}'_U, c^*)$ $precision(c') > \tau$ then
        $c^* \leftarrow c'$;
      end if
    end for
  end for
  $C \leftarrow$ best $k$ members of $C'$;
until $C$ is empty
return $c^*$
```

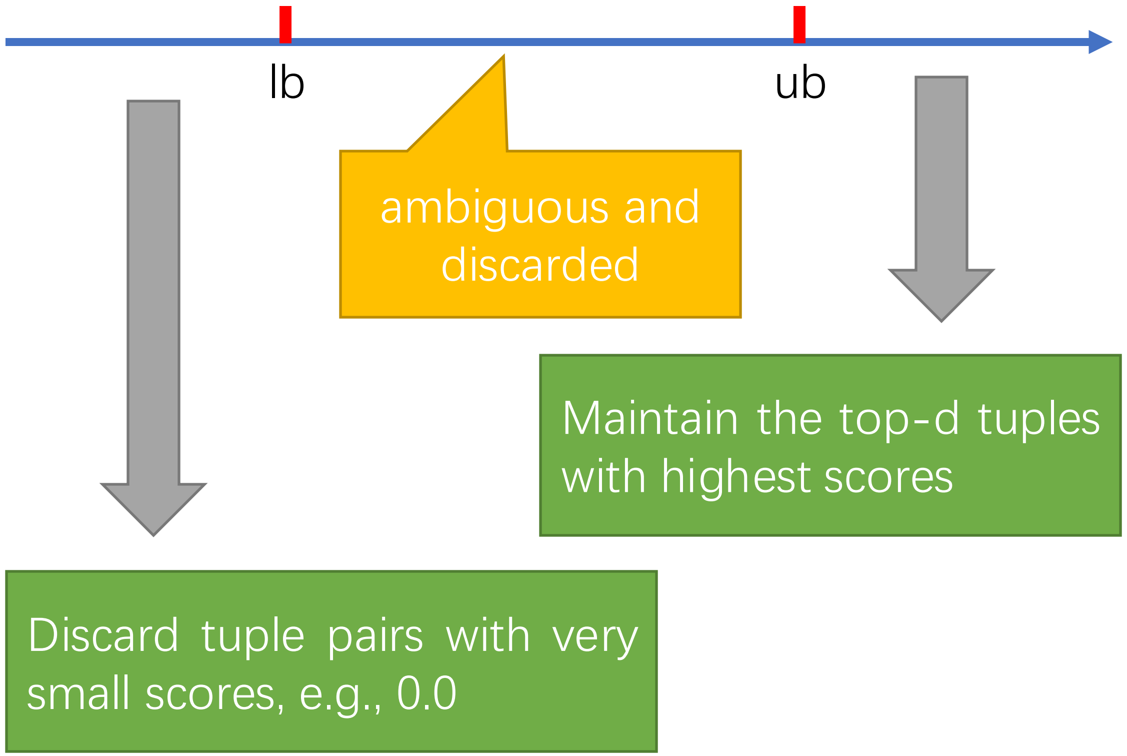
Greedy beam search

Entity Blocking – Fisher [Kejriwal et al., ICDM’13]

- **Unsupervised** rule-based learning
- **Schema-aware**
- Disjunctive Normal Form (DNF) blocking
- **Automatically** generate training instances.
- **Fisher feature selection**

- **> 25% recall** than unsupervised baseline

$$sim(t_1, t_2) = \sum_{q \in t_1 \cap t_2} w(t_1, q).w(t_2, q)$$



ambiguous and discarded

Maintain the top-d tuples with highest scores

Discard tuple pairs with very small scores, e.g., 0.0

Entity Blocking – EM-GBF [Singh et al., VLDB’17]

- Rule-based learning
- **Schema-aware**
- General Boolean Formula(GBF)
- **Large search space**:
  - Combinations of predicates
  - Unknown thresholds for similari functions
- **Interpretable** and **competitive** with tree-based methods (e.g., random forest)

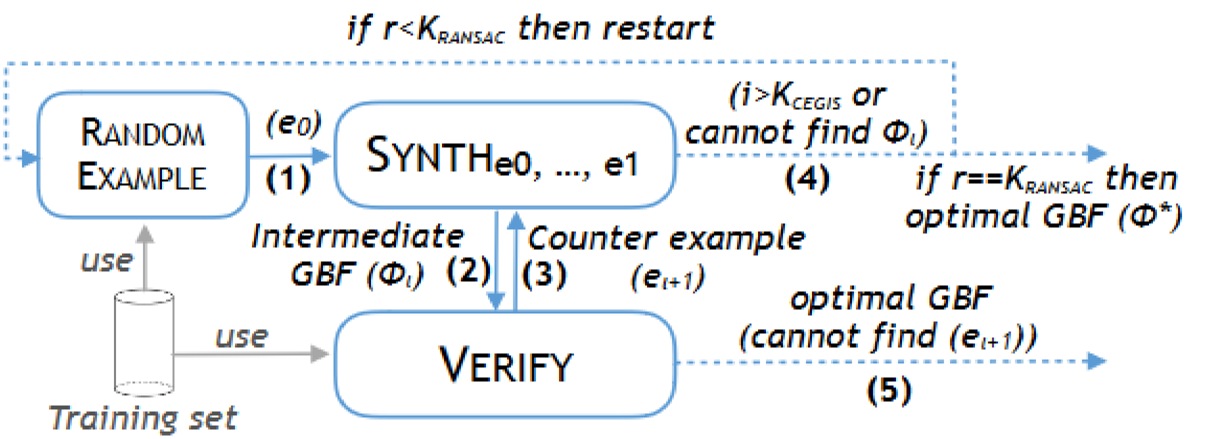
GBF:

grammar $G_{\text{attribute}} \rightarrow r[\mathsf{A_i}] \approx_{(f,\theta)} s[\mathsf{A_i'}]$

$i \in [1,n];\ f \in \mathcal{F};\ \theta \in [0,1]$

grammar $G_{\mathbf{GBF}} \rightarrow G_{\text{attribute}}\ (\mathbf{bound}: N_a)$

$$\left.\begin{array}{l} G_{\mathbf{GBF}} \rightarrow \neg G_{\mathbf{GBF}} \\ G_{\mathbf{GBF}} \rightarrow G_{\mathbf{GBF}} \wedge G_{\mathbf{GBF}} \\ G_{\mathbf{GBF}} \rightarrow G_{\mathbf{GBF}} \vee G_{\mathbf{GBF}} \end{array}\right\} (\mathbf{bound}: N_d)$$


if r<K_RANSAC then restart
RANDOM EXAMPLE
(e0)
(1)
SYNTHe0, ..., e1
(i>K_CEGIS or cannot find Φi)
(4)
if r==K_RANSAC then optimal GBF (Φ*)
use
Intermediate GBF (Φi) (2)
Counter example (3) (ei+1)
use
Training set
VERIFY
optimal GBF (cannot find (ei+1))
(5)

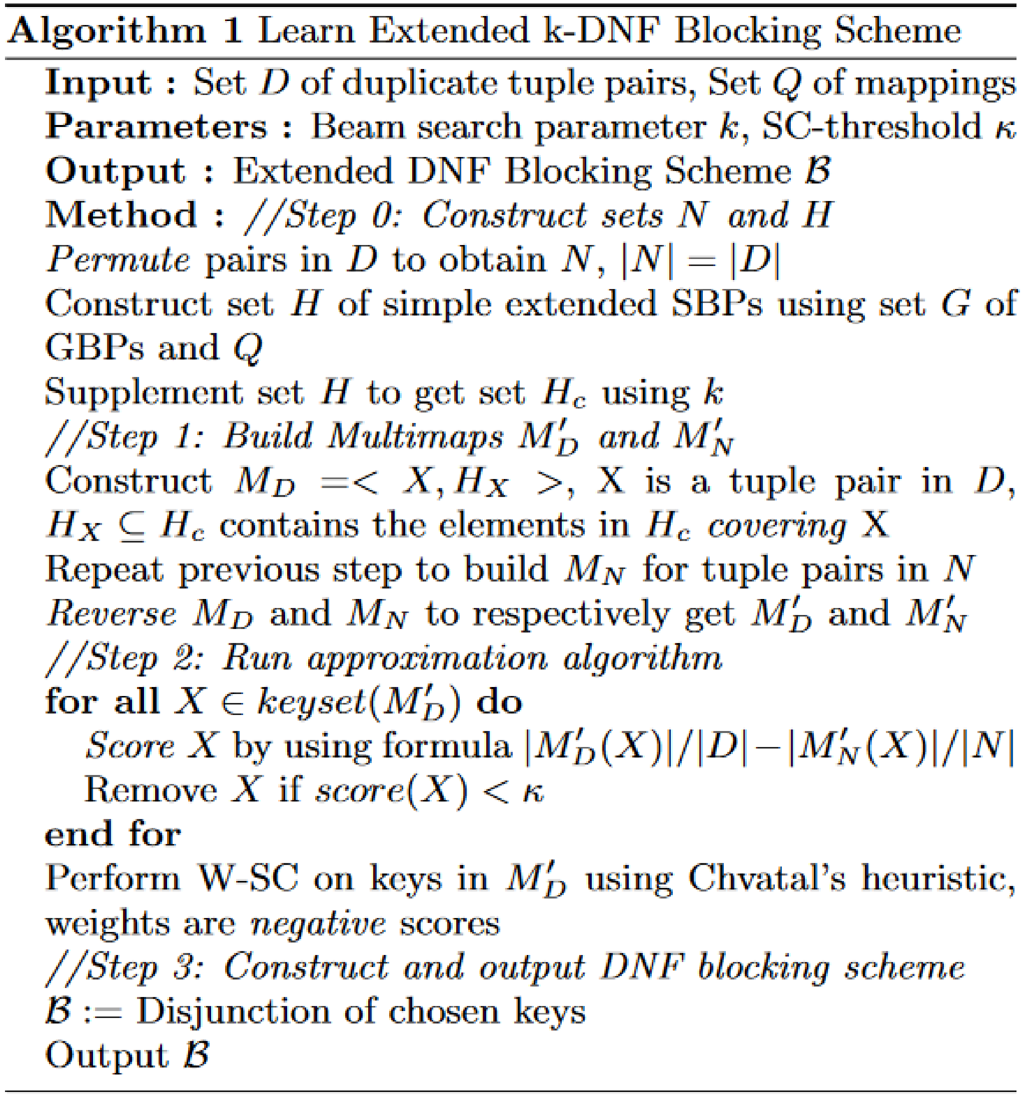
Entity Blocking – DNF-BSL [Kejriwal et al., 2015]

- **Unsupervised** rule-based learning

- **Schema-agnostic**

- DNF blocking

- **Data**: RDF graph, heterogeneous tables



```
Algorithm 1 Learn Extended k-DNF Blocking Scheme
Input : Set $D$ of duplicate tuple pairs, Set $Q$ of mappings
Parameters : Beam search parameter $k$, SC-threshold $\kappa$
Output : Extended DNF Blocking Scheme $\mathcal{B}$
Method : //Step 0: Construct sets N and H
Permute pairs in $D$ to obtain $N$, $|N| = |D|$
Construct set $H$ of simple extended SBPs using set $G$ of GBPs and $Q$
Supplement set $H$ to get set $H_c$ using $k$
//Step 1: Build Multimaps $M'_D$ and $M'_N$
Construct $M_D =< X, H_X >$, X is a tuple pair in $D$, $H_X \subseteq H_c$ contains the elements in $H_c$ covering X
Repeat previous step to build $M_N$ for tuple pairs in $N$
Reverse $M_D$ and $M_N$ to respectively get $M'_D$ and $M'_N$
//Step 2: Run approximation algorithm
for all $X \in keyset(M'_D)$ do
    Score $X$ by using formula $|M'_D(X)|/|D| - |M'_N(X)|/|N|$
    Remove $X$ if $score(X) < \kappa$
end for
Perform W-SC on keys in $M'_D$ using Chvatal's heuristic, weights are negative scores
//Step 3: Construct and output DNF blocking scheme
$\mathcal{B}$ := Disjunction of chosen keys
Output $\mathcal{B}$
```

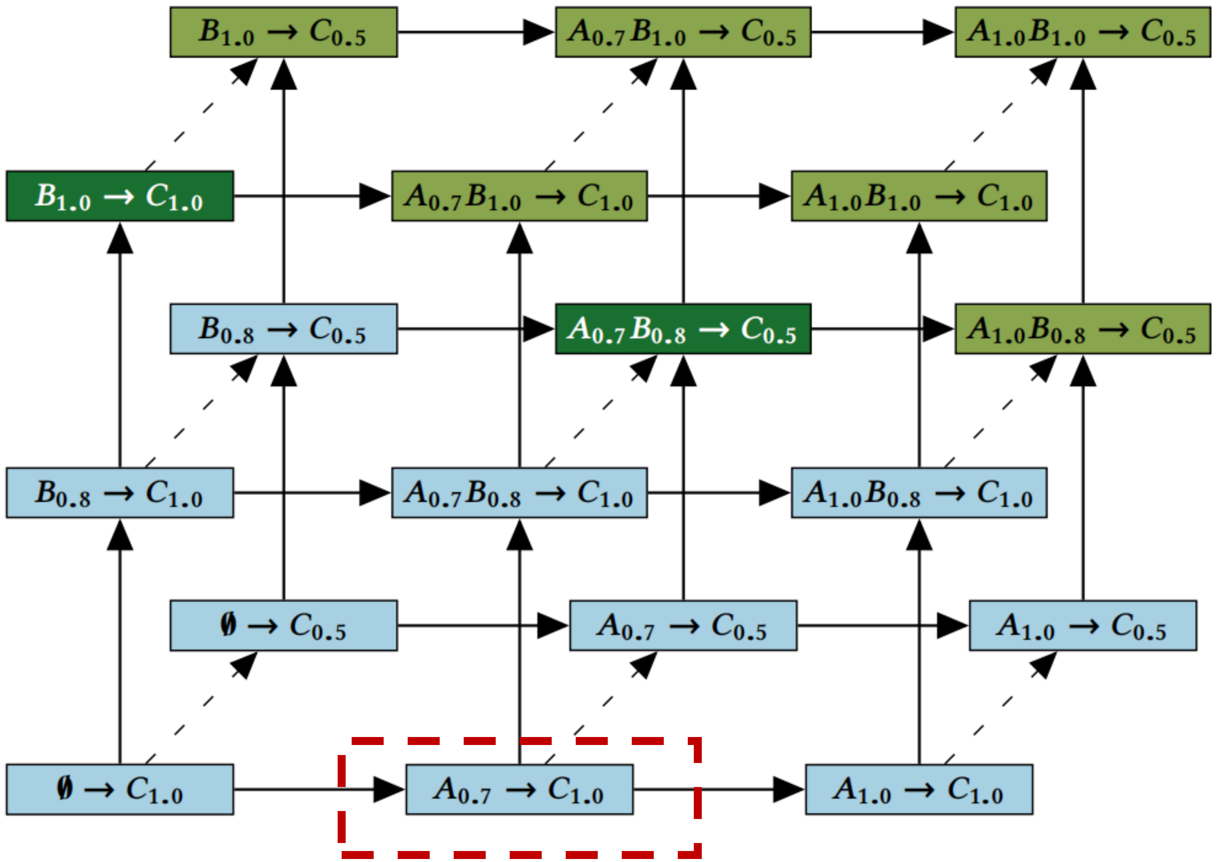
Entity Blocking – HyMD [Schirmer et al., TODS’20]

- NOT learning, based on **mining**
- Need labeled instances
- Matching Dependencies (MDs)

$$\left(\bigwedge_{i=1}^{m} R[A_i] \approx_{i,\lambda_i} S[B_i]\right) \rightarrow R[A_j] \approx_{j,\rho_j} S[B_j]$$

Mine **all minimal MDs** based on some interestingness measures, e.g.,

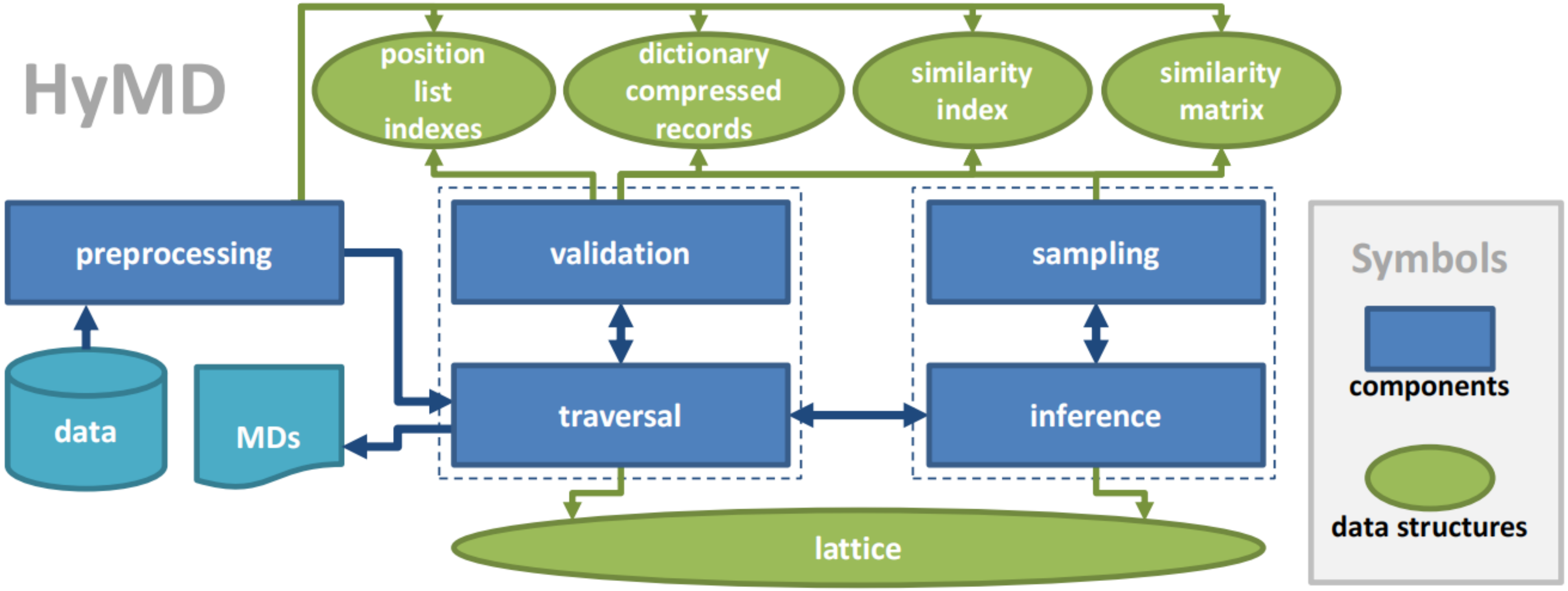
- Large support
- High confidence


$B_{1.0} \rightarrow C_{0.5}$
$A_{0.7}B_{1.0} \rightarrow C_{0.5}$
$A_{1.0}B_{1.0} \rightarrow C_{0.5}$
$B_{1.0} \rightarrow C_{1.0}$
$A_{0.7}B_{1.0} \rightarrow C_{1.0}$
$A_{1.0}B_{1.0} \rightarrow C_{1.0}$
$B_{0.8} \rightarrow C_{0.5}$
$A_{0.7}B_{0.8} \rightarrow C_{0.5}$
$A_{1.0}B_{0.8} \rightarrow C_{0.5}$
$B_{0.8} \rightarrow C_{1.0}$
$A_{0.7}B_{0.8} \rightarrow C_{1.0}$
$A_{1.0}B_{0.8} \rightarrow C_{1.0}$
$\emptyset \rightarrow C_{0.5}$
$A_{0.7} \rightarrow C_{0.5}$
$A_{1.0} \rightarrow C_{0.5}$
$\emptyset \rightarrow C_{1.0}$
$A_{0.7} \rightarrow C_{1.0}$
$A_{1.0} \rightarrow C_{1.0}$

1. Try all **valid combinations** of similarity functions
2. Different thresholds, e.g., 0.7 of A

Entity Blocking – HyMD [Schirmer et al., TODS'20]

- **Predicates**: exact match and similarity functions (e.g., Jaccard, Edit distance, etc.)
- Hybrid search: levelwise + depth-first search

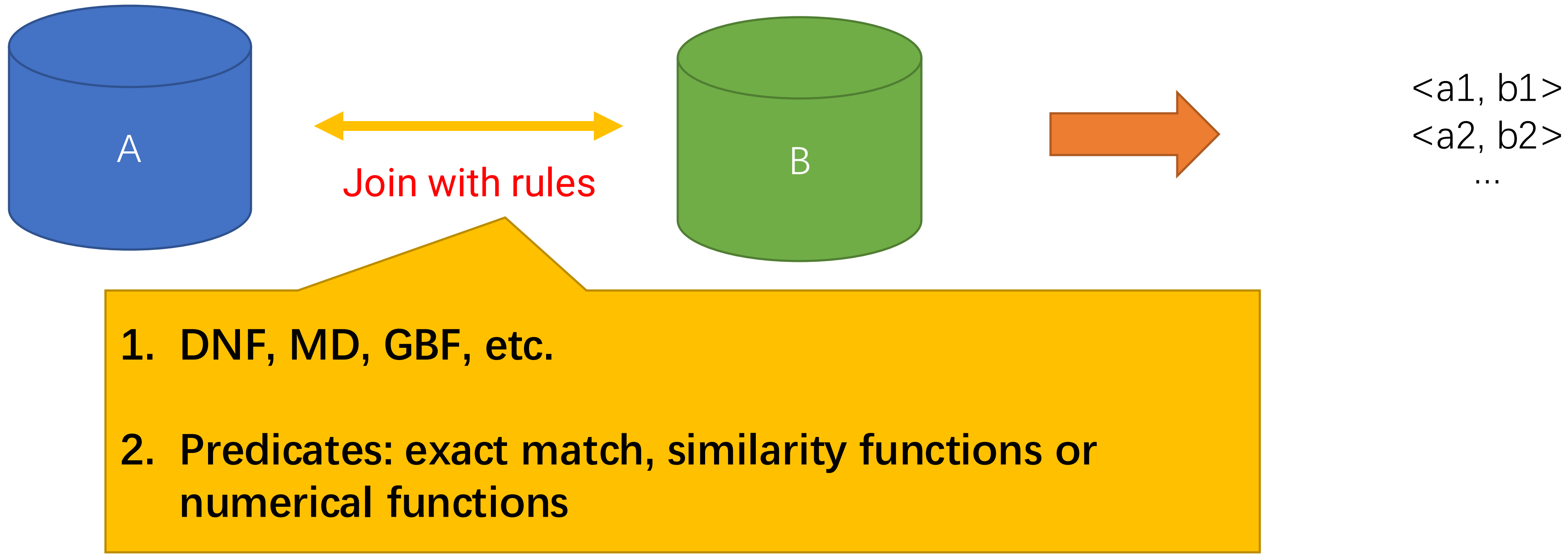


Performance

- **High precision and low recall**
- **F-1 is not higher than RF.**

Entity Blocking – Fast Query Processing

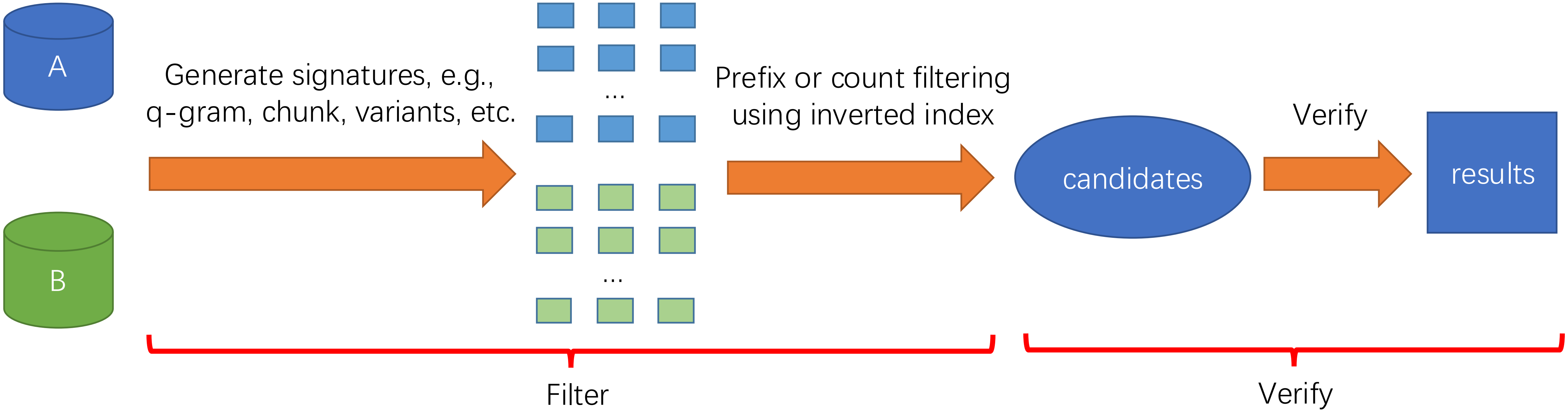
- **Problem**: Given two large relational tables A and B, and **multiple learned rules**, efficiently find all satisfied tuple pairs.



1. **DNF, MD, GBF, etc.**
2. **Predicates: exact match, similarity functions or numerical functions**

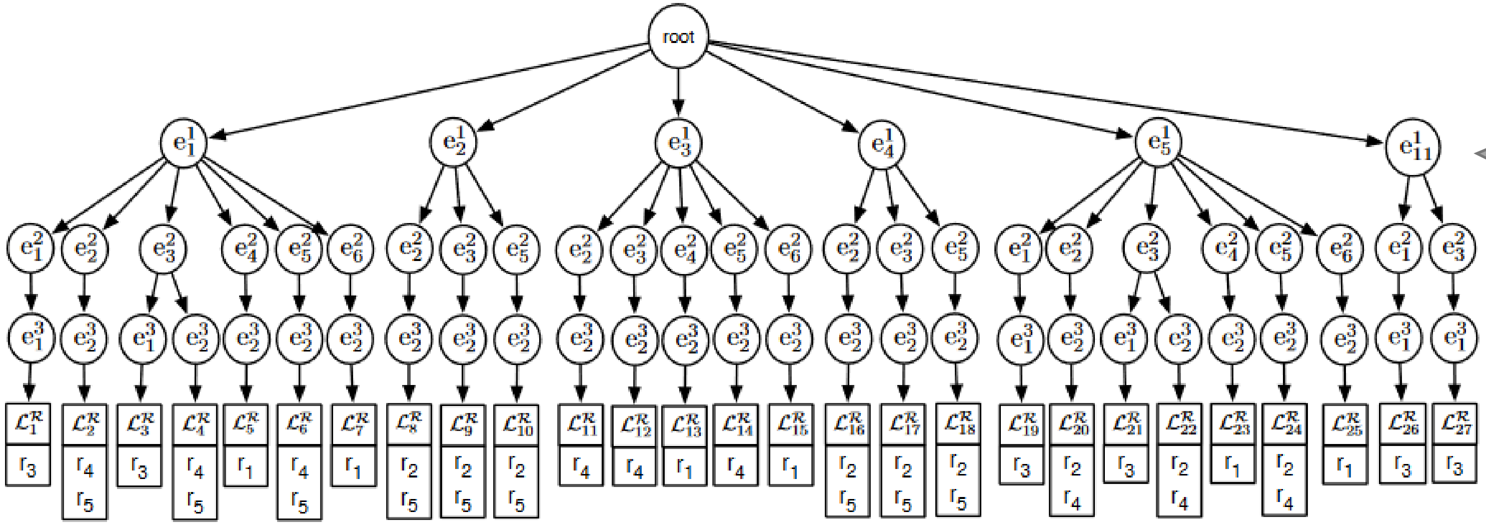
Entity Blocking – Fast Query Processing

- **Problem**: Given two large relational tables A and B, and multiple **learned rules**, how to efficiently find all satisfied tuple pairs ?
- **Case I**: rule is **ONE single similarity function**, e.g., Jaccard(a, b) >= 0.8
- **Algorithm**: Similarity search and join (e.g., prefix/count filtering)


A
B
Generate signatures, e.g., q-gram, chunk, variants, etc.
...
...
Prefix or count filtering using inverted index
candidates
Verify
results
Filter
Verify

Entity Blocking – Fast Query Processing

- **Problem**: Given two large relational tables A and B, and multiple **learned rules**, how to efficiently find all satisfied tuple pairs ?
- **Case II**: rule is ONE **conjunctive query of similarity functions**, e.g., Jaccard(t1, s1) >= 0.8 ^ ED(t1, s1) < 2
- **Algorithm**: Multi-attribute similarity join[Li et al. SIGMOD15']



root
e_1^1 e_2^1 e_3^1 e_4^1 e_5^1 e_{11}^1
e_1^2 e_2^2 e_3^2 e_4^2 e_5^2 e_6^2 e_2^2 e_3^2 e_5^2 e_2^2 e_3^2 e_4^2 e_5^2 e_6^2 e_2^2 e_3^2 e_5^2 e_1^2 e_2^2 e_3^2 e_4^2 e_5^2 e_6^2 e_1^2 e_3^2
e_1^3 e_2^3 e_1^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_2^3 e_1^3 e_2^3 e_1^3 e_2^3 e_2^3 e_2^3 e_2^3 e_1^3 e_1^3
$\mathcal{L}_1^{\mathcal{R}}$ r_3
$\mathcal{L}_2^{\mathcal{R}}$ r_4 r_5
$\mathcal{L}_3^{\mathcal{R}}$ r_3
$\mathcal{L}_4^{\mathcal{R}}$ r_4 r_5
$\mathcal{L}_5^{\mathcal{R}}$ r_1
$\mathcal{L}_6^{\mathcal{R}}$ r_4 r_5
$\mathcal{L}_7^{\mathcal{R}}$ r_1
$\mathcal{L}_8^{\mathcal{R}}$ r_2 r_5
$\mathcal{L}_9^{\mathcal{R}}$ r_2 r_5
$\mathcal{L}_{10}^{\mathcal{R}}$ r_2 r_5
$\mathcal{L}_{11}^{\mathcal{R}}$ r_4
$\mathcal{L}_{12}^{\mathcal{R}}$ r_4
$\mathcal{L}_{13}^{\mathcal{R}}$ r_1
$\mathcal{L}_{14}^{\mathcal{R}}$ r_4
$\mathcal{L}_{15}^{\mathcal{R}}$ r_1
$\mathcal{L}_{16}^{\mathcal{R}}$ r_2 r_5
$\mathcal{L}_{17}^{\mathcal{R}}$ r_2 r_5
$\mathcal{L}_{18}^{\mathcal{R}}$ r_2 r_5
$\mathcal{L}_{19}^{\mathcal{R}}$ r_3
$\mathcal{L}_{20}^{\mathcal{R}}$ r_2 r_4
$\mathcal{L}_{21}^{\mathcal{R}}$ r_3
$\mathcal{L}_{22}^{\mathcal{R}}$ r_2 r_4
$\mathcal{L}_{23}^{\mathcal{R}}$ r_1
$\mathcal{L}_{24}^{\mathcal{R}}$ r_2 r_4
$\mathcal{L}_{25}^{\mathcal{R}}$ r_1
$\mathcal{L}_{26}^{\mathcal{R}}$ r_3
$\mathcal{L}_{27}^{\mathcal{R}}$ r_3

- Construct an optimal prefix tree
- Each level is one similarity function

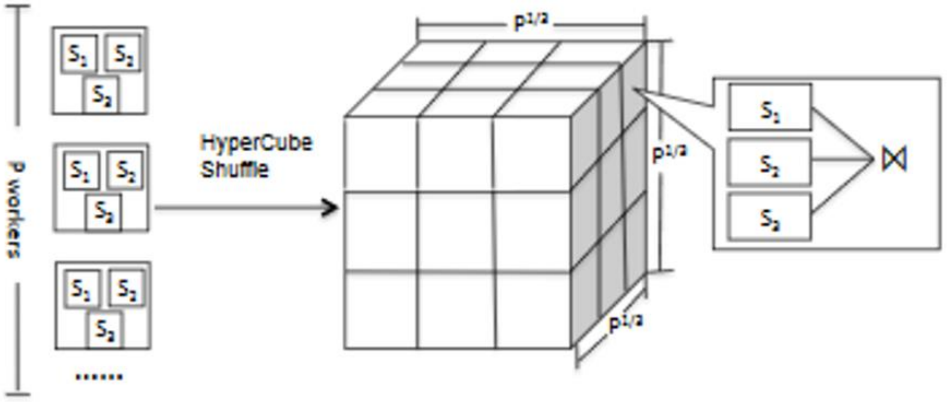
Entity Blocking – Fast Query Processing

- **Problem**: Given two large relational tables A and B, and multiple **learned rules**, how to efficiently find all satisfied tuple pairs ?
- **Case III**: rules are **GBF, DNF, or multi-MDs,** e.g., (Jaccard(t1, s1) >= 0.8 ^ ED(t1, s1) < 2) V (Jaro-Winkler(t1, s1) > 0.75) V …
- **Algorithm**: ErrorDetect [Fan et al. VLDB20']

A

B

Hash and partition for each used attributes



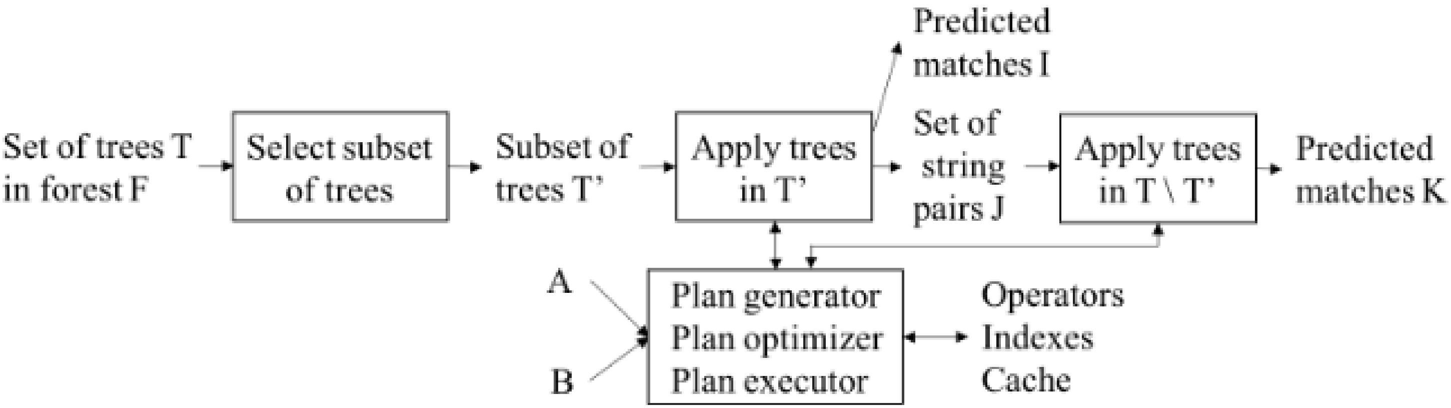
Extended Hyper-Cube for Similarity, ML and inequality

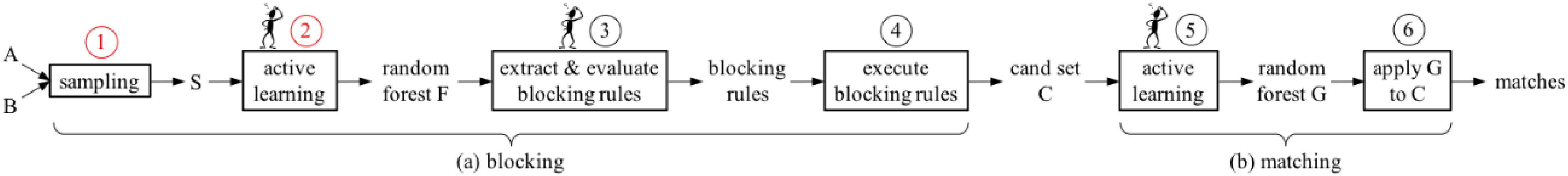
Distribute cubes to workers

Aggregate results

Entity Blocking – Smurf [Suganthan G. C. et al., VLDB'19]

- Learn a tree-based binary classifier, e.g., decision tree, **random forest**
- Use labeled data
- **Active learning**
- Blocking with **random forest**
- Reduction of candidate pairs:
  - **42.8-75.6%**





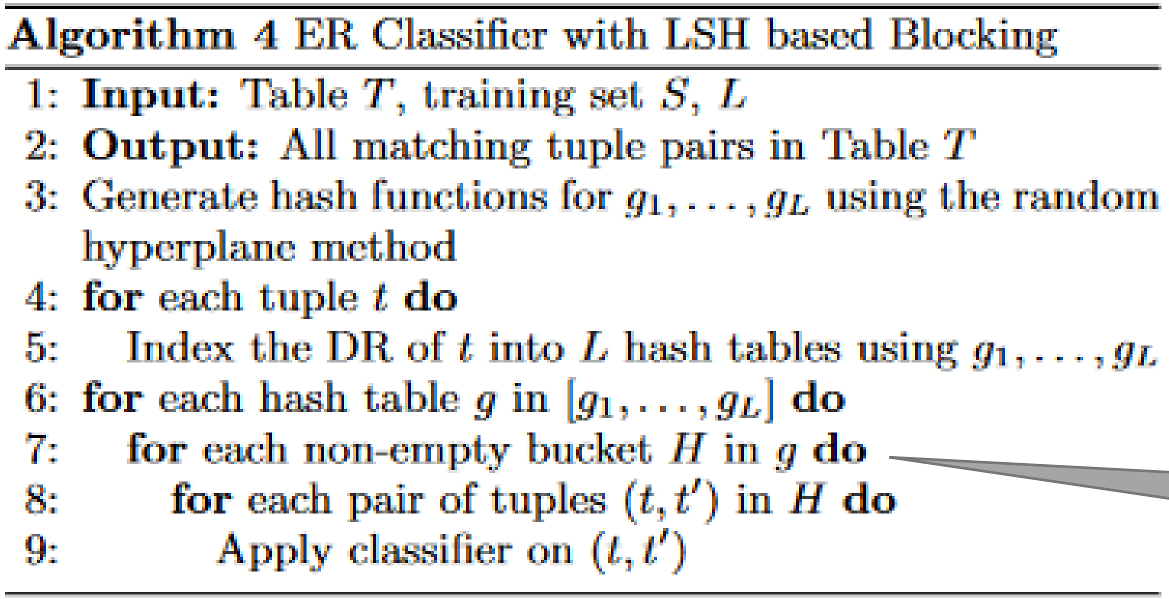
Entity Blocking – Meta-Blocking [Papadakis et al., VLDB'14]

- Construct a blocking graph
- Learn a binary classifier to predict match or non-match for each edge
- **Feature engineering**

Entity Blocking – DeepER [Ebraheem et al., VLDB’18]

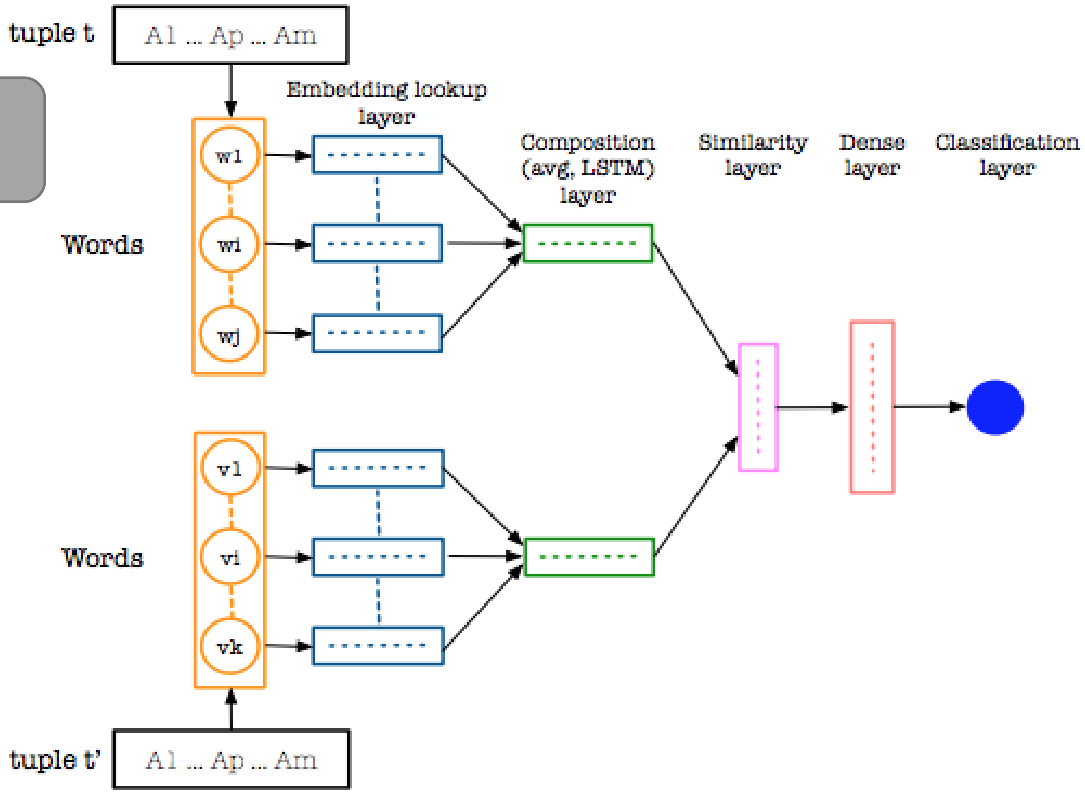
- Learn **tuple representation**
- LSH-based blocking
- **Multi-Probe LSH** for Blocking

Increase recall



Algorithm 4 ER Classifier with LSH based Blocking

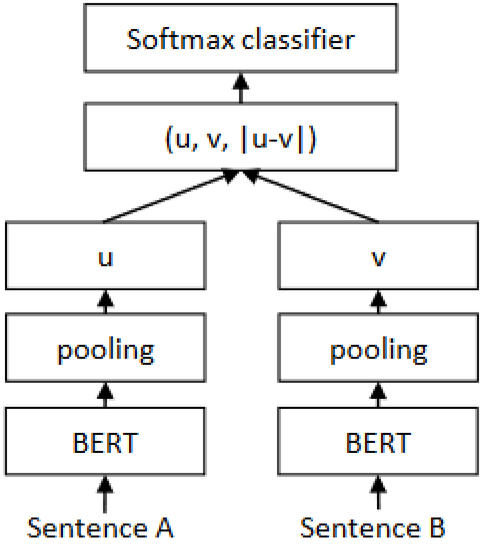
```
Input: Table $T$, training set $S$, $L$
Output: All matching tuple pairs in Table $T$
Generate hash functions for $g_1, \ldots, g_L$ using the random
hyperplane method
for each tuple $t$ do
    Index the DR of $t$ into $L$ hash tables using $g_1, \ldots, g_L$
for each hash table $g$ in $[g_1, \ldots, g_L]$ do
    for each non-empty bucket $H$ in $g$ do
        for each pair of tuples $(t, t')$ in $H$ do
            Apply classifier on $(t, t')$
```

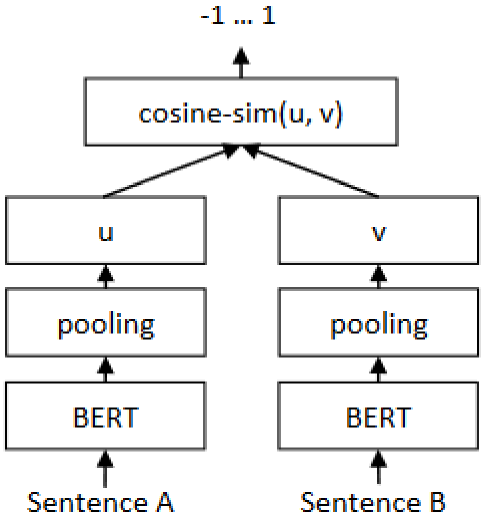


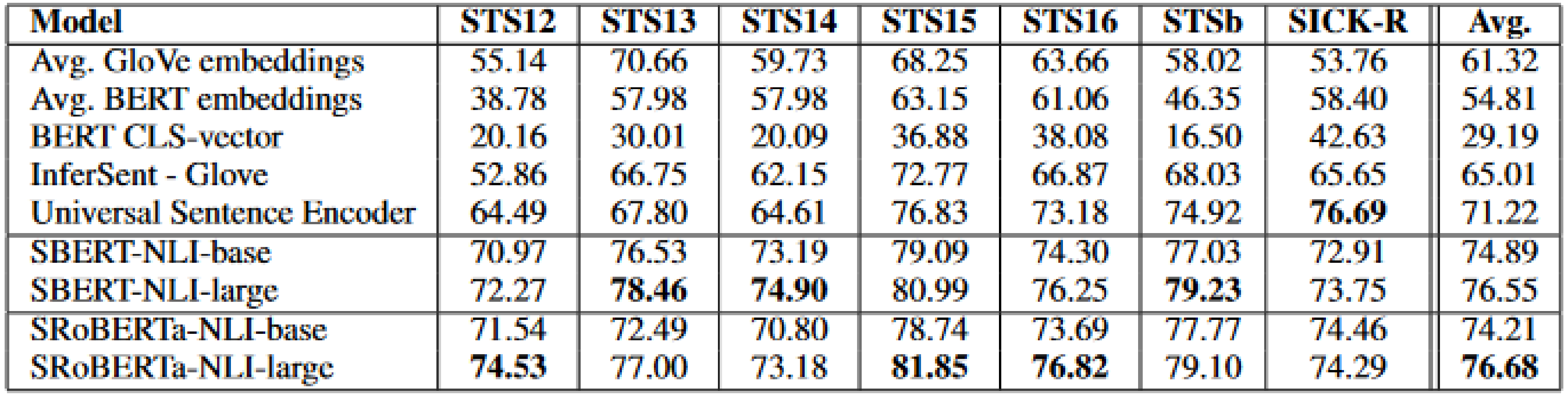
Tuples in the same buckets are considered as candidates

Entity Blocking – SBert [Reimers et al., EMNLP'19]

- **Siamese** Bert
- Generate tuple embedding
- **Cosine similarity**
- Better than SOTA embedding methods, but worse than matching models.

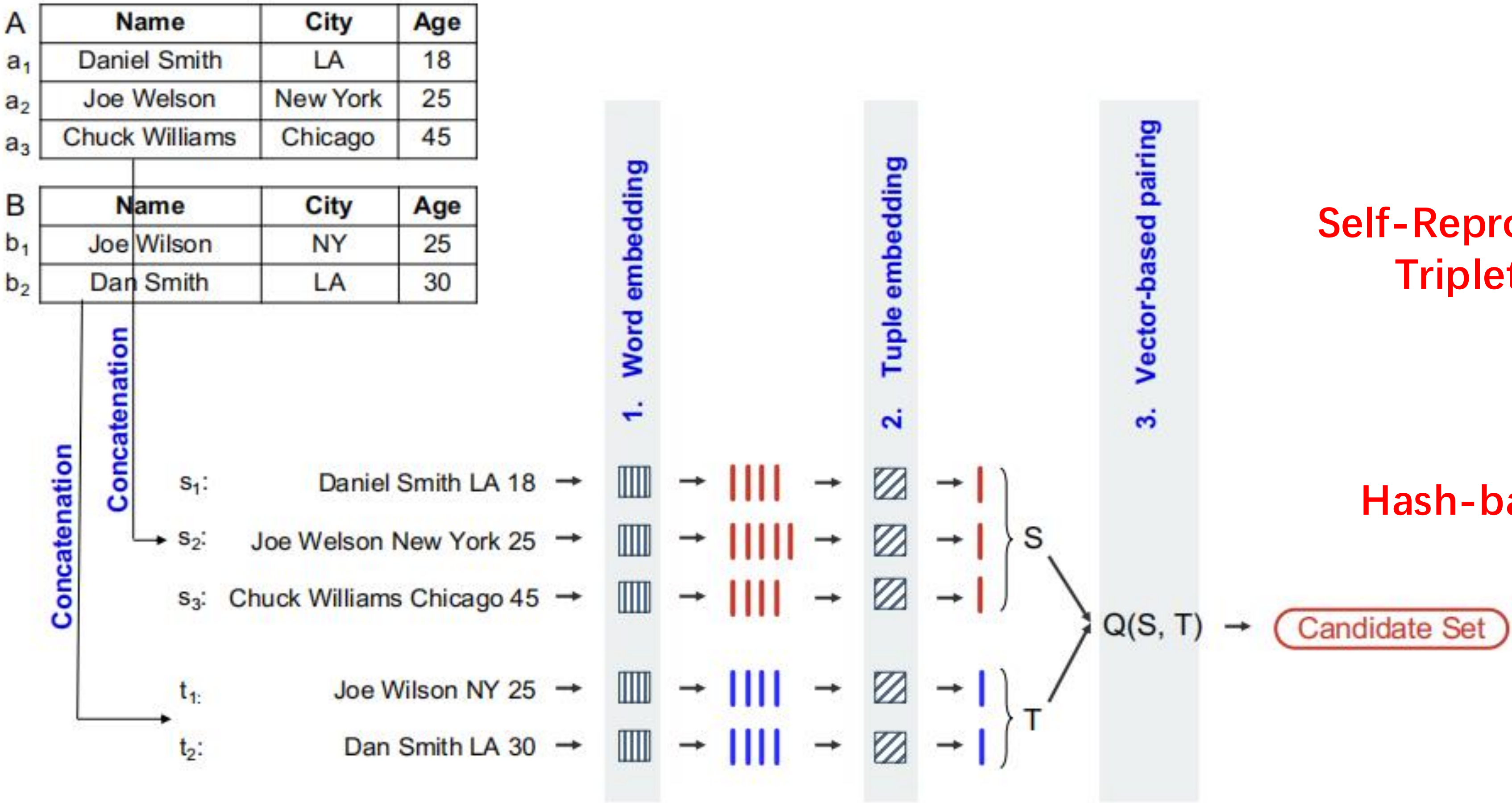






Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
Avg. GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - Glove	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
SBERT-NLI-base	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-NLI-large	72.27	**78.46**	**74.90**	80.99	76.25	**79.23**	73.75	76.55
SRoBERTa-NLI-base	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa-NLI-large	**74.53**	77.00	73.18	**81.85**	**76.82**	79.10	74.29	**76.68**

Entity Blocking – DL blocking [Thirumuruganathan et al., VLDB'21]



A	Name	City	Age
a_1	Daniel Smith	LA	18
a_2	Joe Welson	New York	25
a_3	Chuck Williams	Chicago	45

B	Name	City	Age
b_1	Joe Wilson	NY	25
b_2	Dan Smith	LA	30

Word Embedding

Self-Reproduction, Cross-Tuple Training, Triplet Loss Minimization, Hybrid

Tuple Embedding

Hash-based, Similarity-based, and Composite Pairing

Vector-based pairing

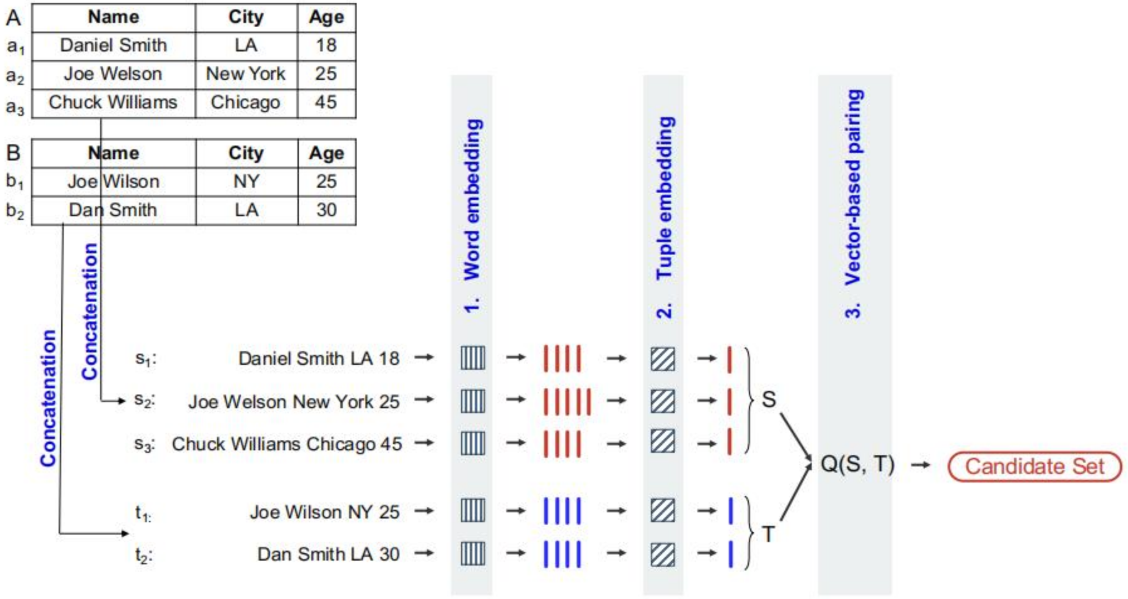
Entity Blocking – DL blocking [Thirumuruganathan et al., VLDB’21]

- **SFT:**

(1) Averaged averaging; (2) PCA

$$f(w) = a/(a + p(w))$$

$$\mathbf{u}_t = \mathbf{v}_t - \mathbf{p}\mathbf{p}^T\mathbf{v}_t$$

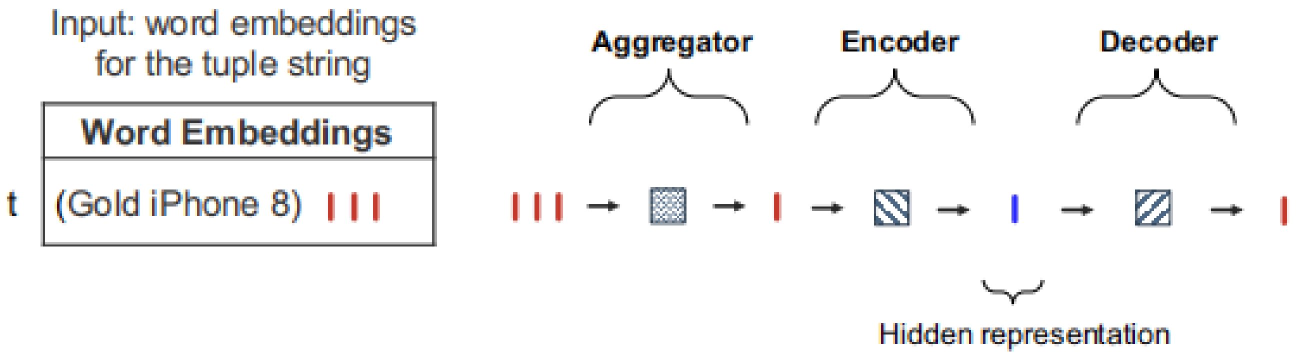


A	Name	City	Age
a_1	Daniel Smith	LA	18
a_2	Joe Welson	New York	25
a_3	Chuck Williams	Chicago	45

B	Name	City	Age
b_1	Joe Wilson	NY	25
b_2	Dan Smith	LA	30

- **Auto-Encoder**

Self-Reproduction, **do not need labeled data**



SIFT + Encoder-Decoder (FFN)

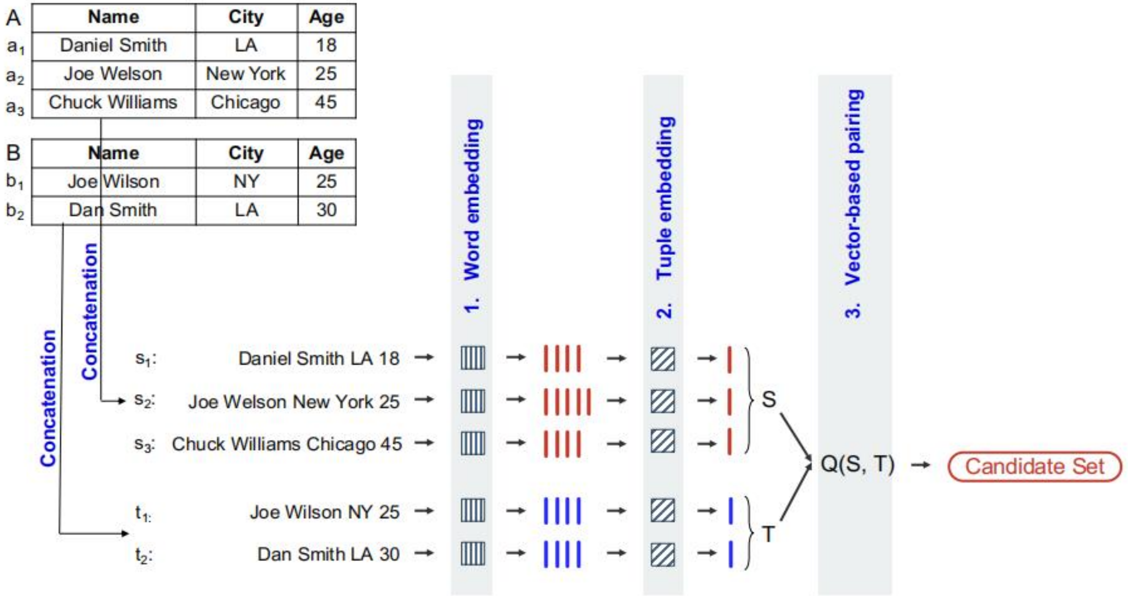
Entity Blocking – DL blocking [Thirumuruganathan et al., VLDB'21]

- **Trans-encoder**

Transformer as encoder/decoder

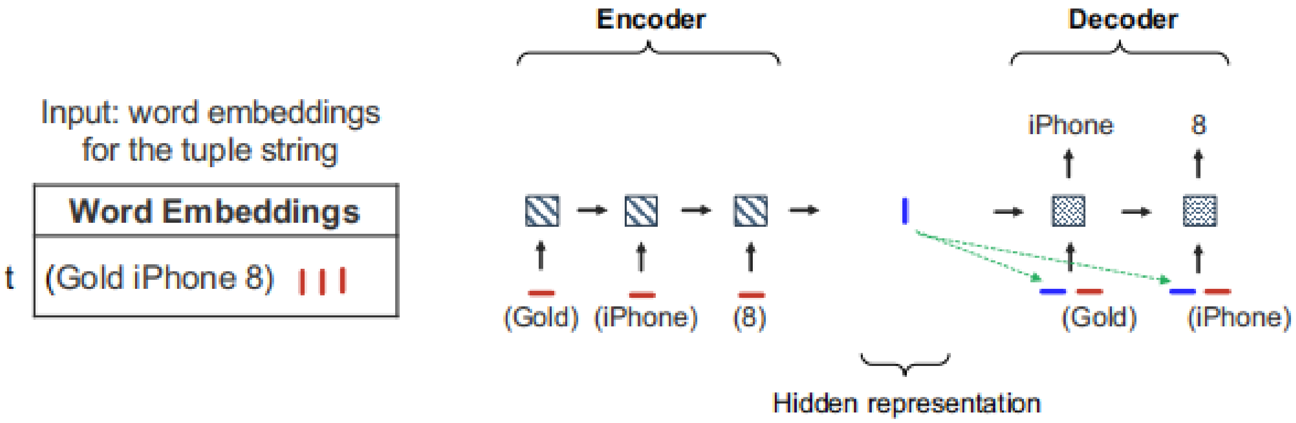
- **Seq2seq**

LSTM-RNN as encoder/decoder



A	Name	City	Age
a_1	Daniel Smith	LA	18
a_2	Joe Welson	New York	25
a_3	Chuck Williams	Chicago	45

B	Name	City	Age
b_1	Joe Wilson	NY	25
b_2	Dan Smith	LA	30



Entity Blocking – Trans-encoder [Thirumuruganathan et al., VLDB’21]

- **Transformer** [Vaswani et al. NeurIPS17’]
  - Scaled Dot-Product Attention

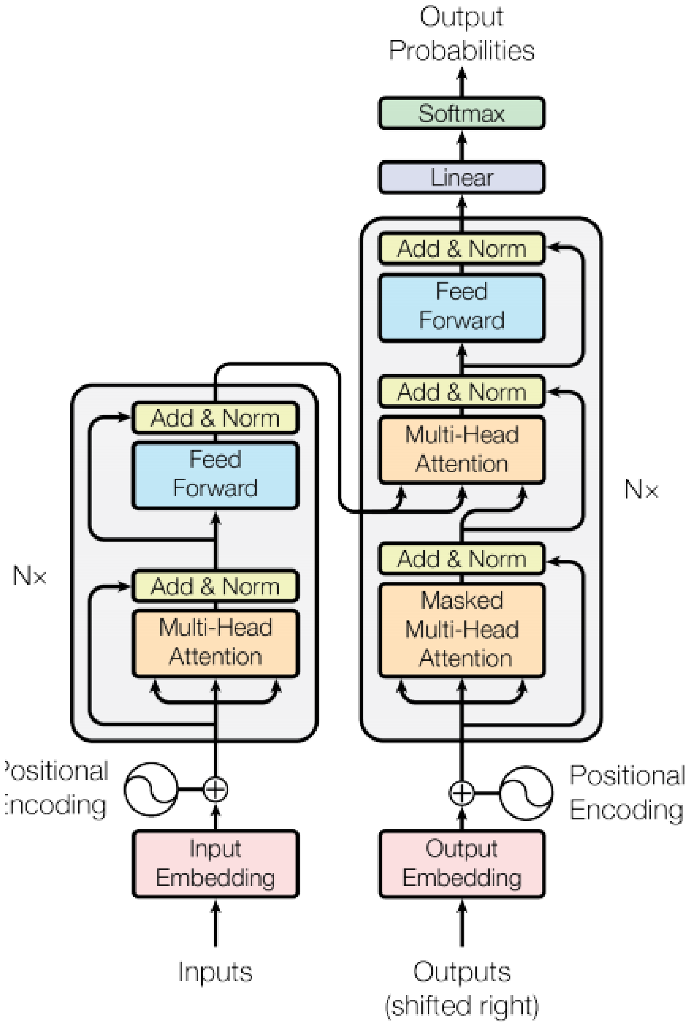
$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

  - Multi-head attention

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1},...,\mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

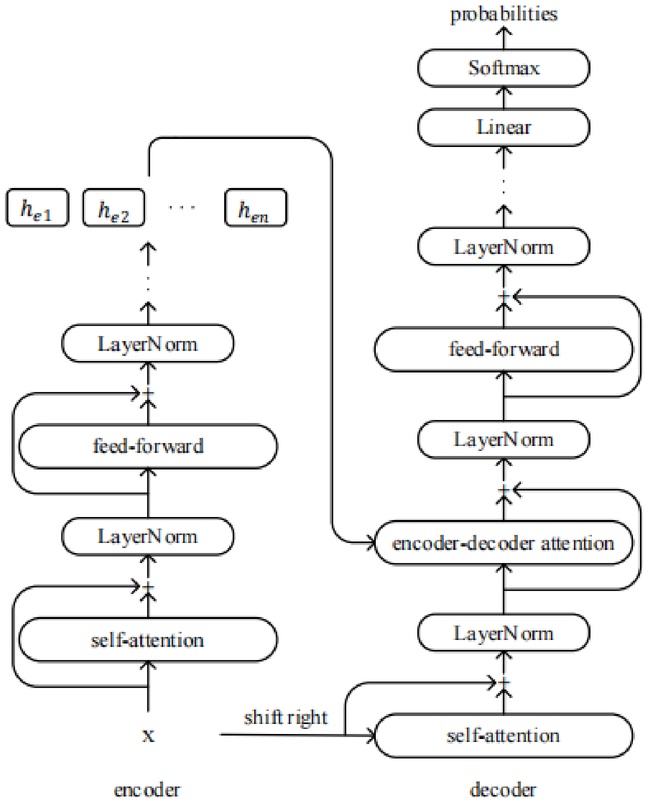
  - Position-wise Feed-Forward Networks
  - Positional encoding

The final representation is the embedding of [CLS]

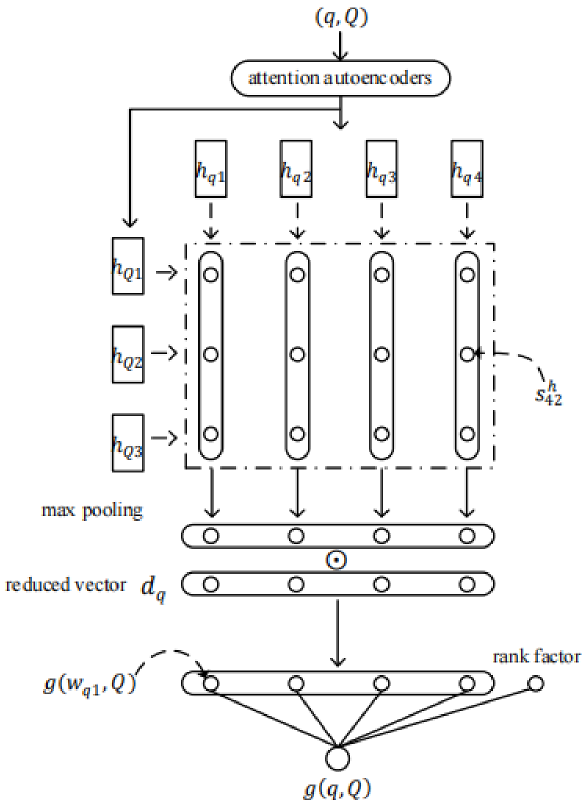


Entity Blocking – AttentionAE [Zhang et al., AAAI'18]

- **Attention Autoencoder**



Attention Autoencoder



Attentive Matching Network

Similarity based on hidden representation

Add lexical matching

Rank Factor

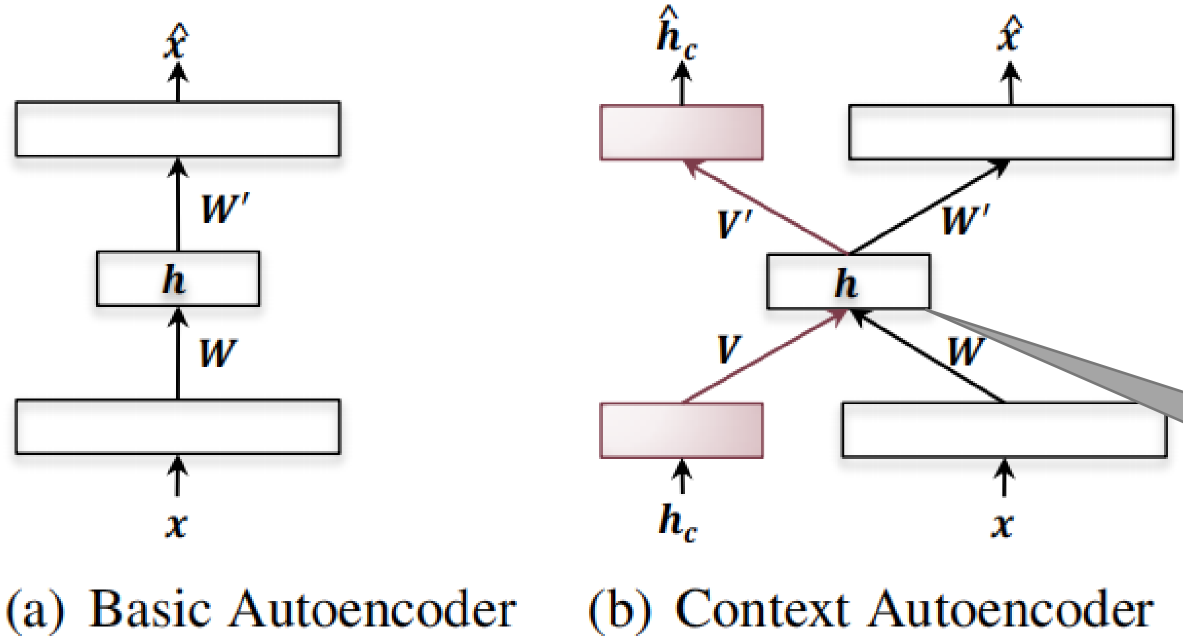
$$R = 1 - \alpha * rank$$

$$g(q, Q) = R \prod_i g(w_{qi}, Q)$$

Entity Blocking – CSAE [Zhang et al., AAAI'18]

- **CSAE**

Add context information into AE



(a) Basic Autoencoder (b) Context Autoencoder

$$l(\mathbf{x}, \mathbf{h}_c) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 + \lambda||\mathbf{h}_c - \hat{\mathbf{h}_c}||^2$$

$$\min_{\boldsymbol{\Theta}} \sum_{i=1}^{n} l(\mathbf{x}^{(i)}, \mathbf{h}_c^{(i)})$$

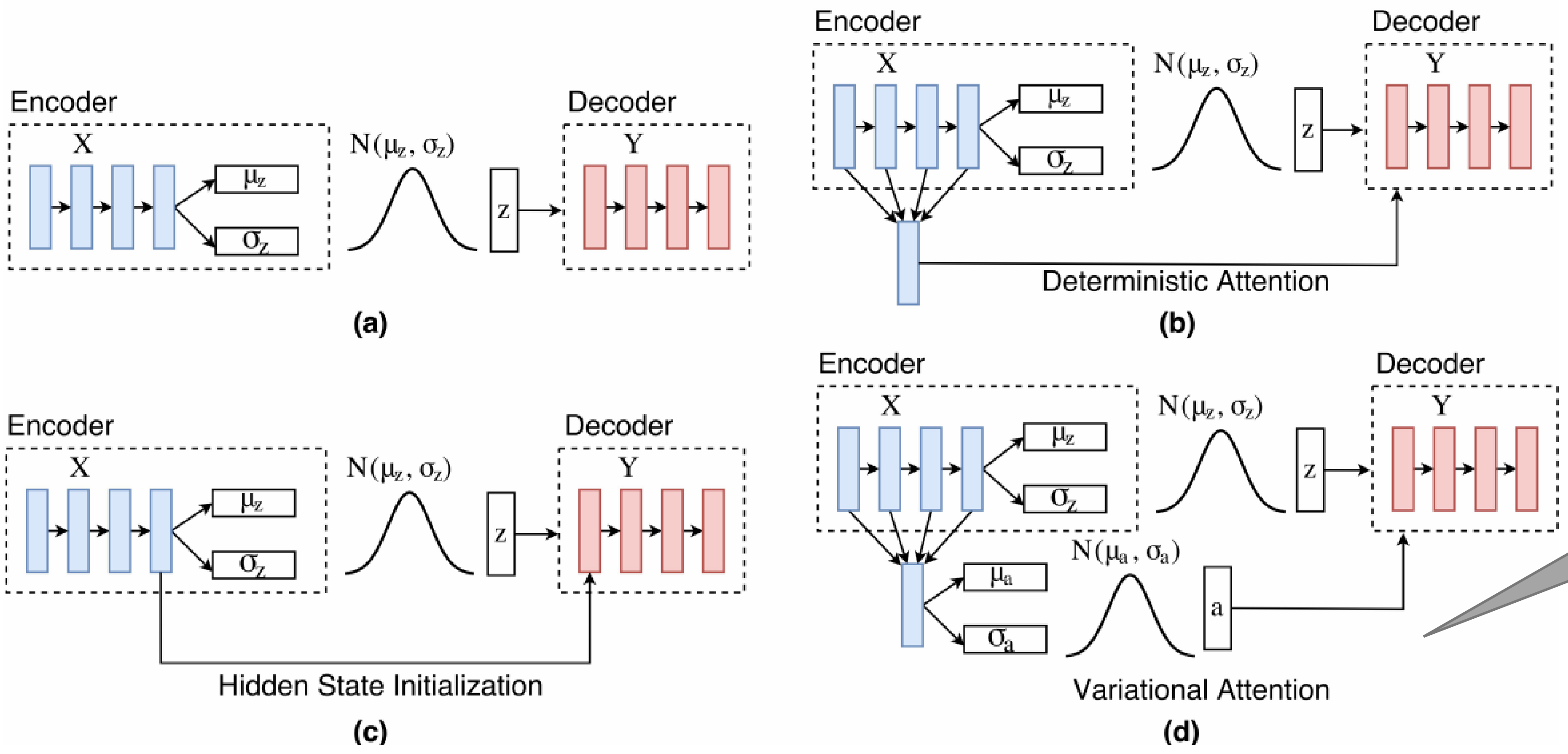
$$\boldsymbol{\Theta} = \{\mathbf{W}, \mathbf{W}', \mathbf{V}, \mathbf{V}', \mathbf{b_h}, \mathbf{b_{\hat{x}}}, \mathbf{b_{\hat{h}_c}}\},$$

Reconstruction loss of both the original data and context information

h is the dense representation of the original data and context

Entity Blocking – VED [Bahuleyan et al., COLING'18]

- **Variational Encoder-Decoder**



Add an extra Variational Attention mechanism in Seq2seq model

$$J^{(n)}(\boldsymbol{\theta}, \boldsymbol{\phi}) = J_{\text{rec}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{y}^{(n)}) + \lambda_{\text{KL}} \left[\text{KL}\left(q_{\phi}^{(z)}(\boldsymbol{z}|\boldsymbol{x}^{(n)}) \| p(\boldsymbol{z}) \right) + \gamma_a \sum_{j=1}^{|\boldsymbol{y}|} \text{KL}\left(q_{\phi}^{(a)}(\boldsymbol{a}_j|\boldsymbol{x}^{(n)}) \| p(\boldsymbol{a}_j) \right) \right]$$

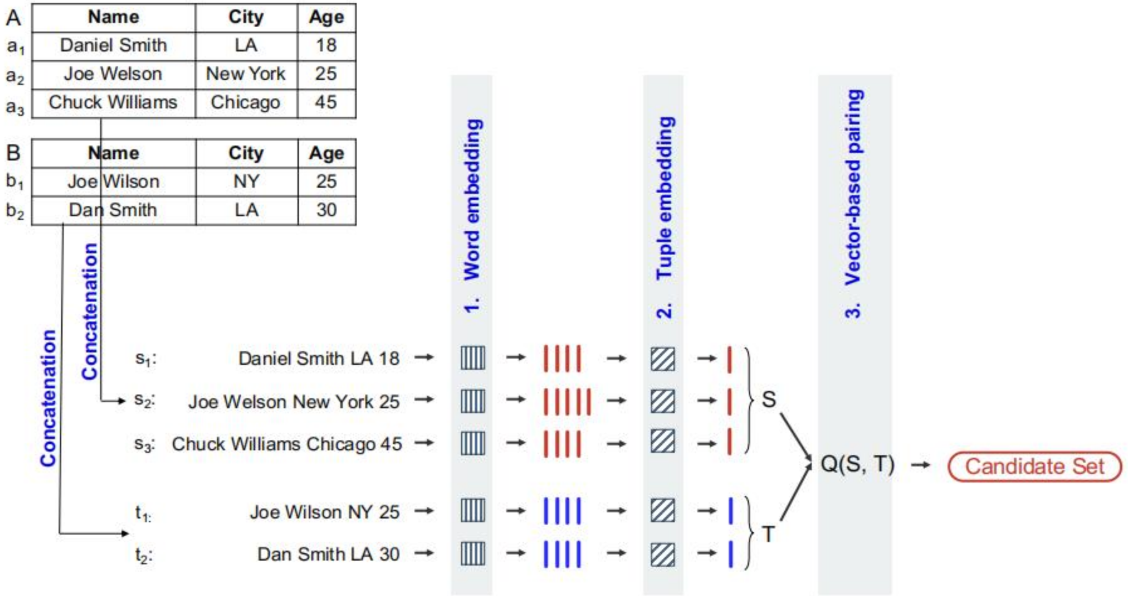
Entity Blocking – DL blocking [Thirumuruganathan et al., VLDB'21]

- **CTT**

Automatically generate labeled data

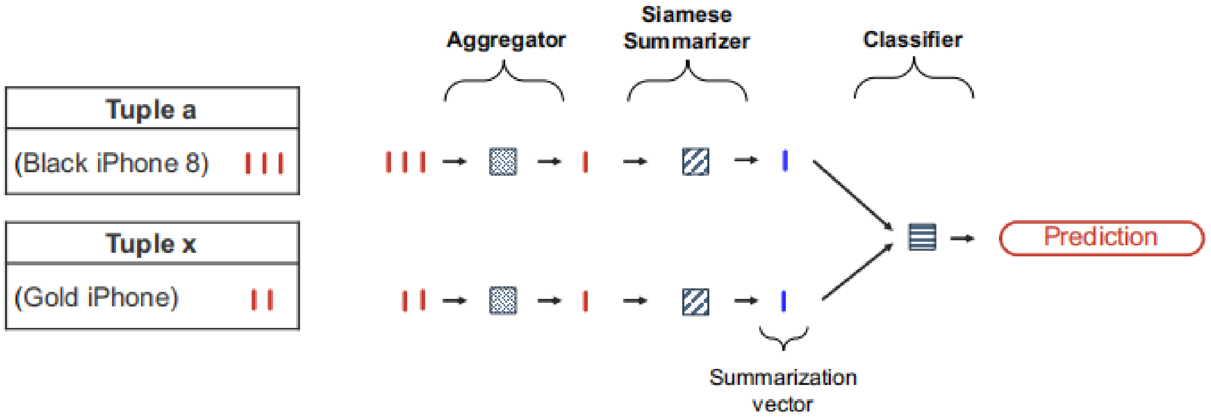
(1)**Positive**: synthetic matching (randomly select a subset of words, at least 60% overlap)

(2)**Negative**: Randomly select one tuple.



A	Name	City	Age
a_1	Daniel Smith	LA	18
a_2	Joe Welson	New York	25
a_3	Chuck Williams	Chicago	45

B	Name	City	Age
b_1	Joe Wilson	NY	25
b_2	Dan Smith	LA	30



- **Cross Entropy loss**

e.g., DeepER [Ebraheem et al., VLDB'18]

- **Triple loss**

$$\max\left(||Emb(x)-Emb(y)||^2-||Emb(x)-Emb(z)||^2+\alpha,0\right)$$

Entity Blocking – DL blocking [Thirumuruganathan et al., VLDB'21]

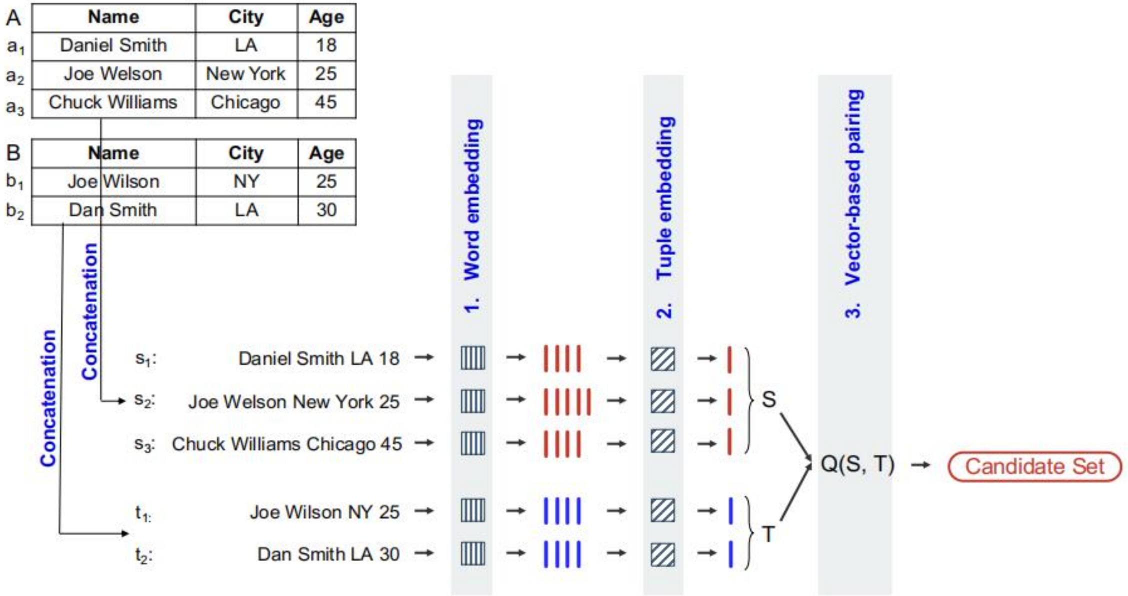
- **CTT-cosine**

Replace the classifier with **Cosine similarity**.

- **Hybrid**

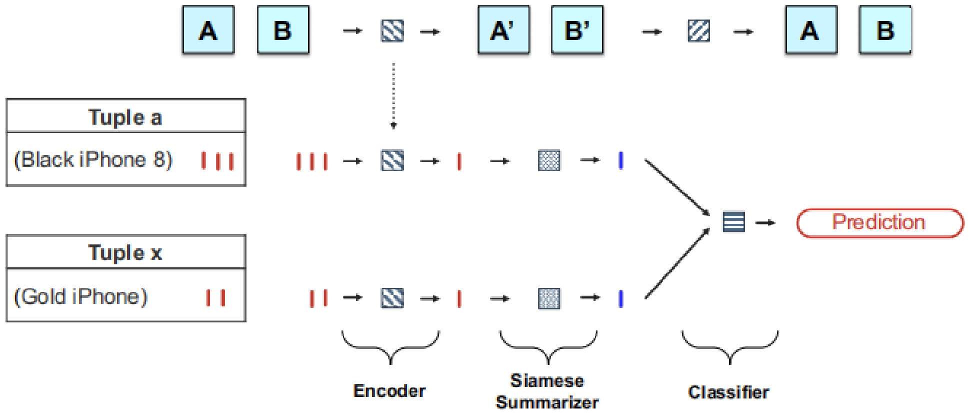
Combine CCT and AE.

Replace the aggregator of CCT **with the encoder of AE**.



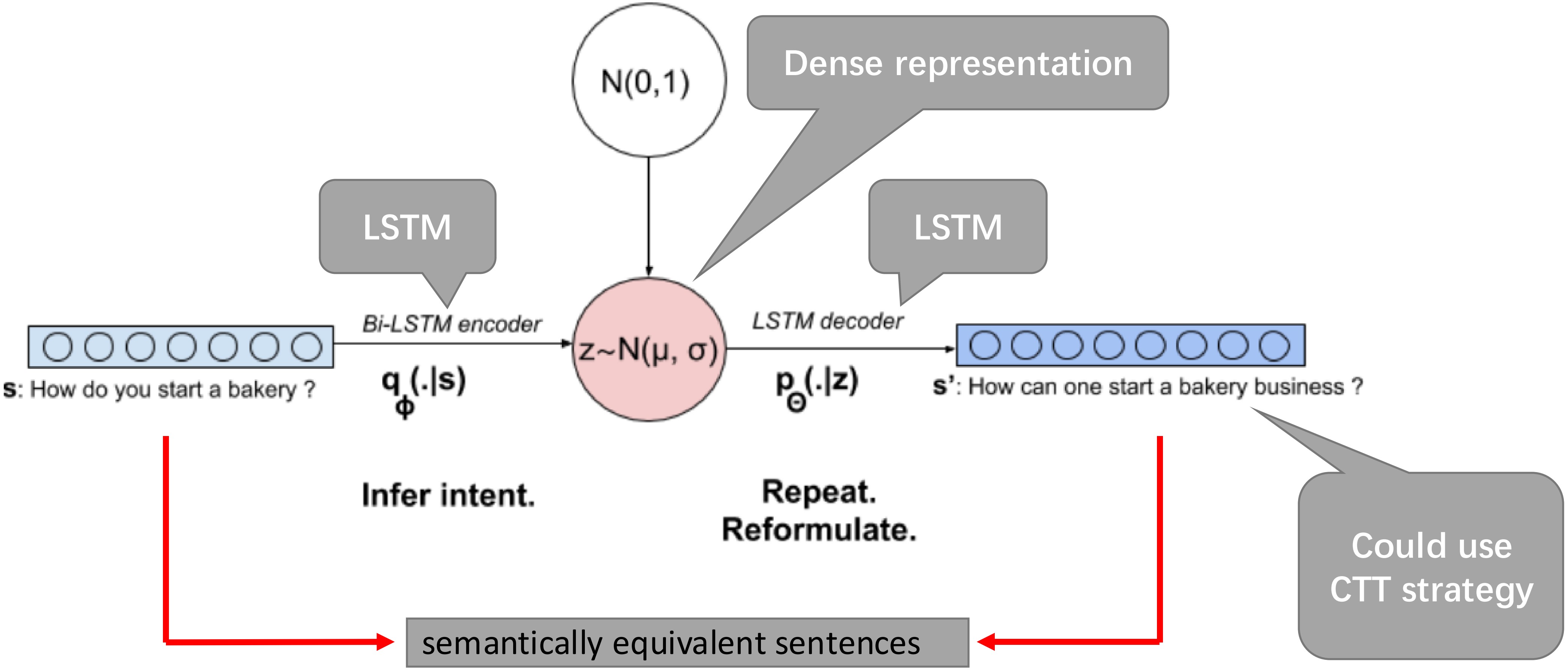
A	Name	City	Age
a_1	Daniel Smith	LA	18
a_2	Joe Welson	New York	25
a_3	Chuck Williams	Chicago	45

B	Name	City	Age
b_1	Joe Wilson	NY	25
b_2	Dan Smith	LA	30



Entity Blocking – VAR-Siamese [Michel Deudon., NeurIPS'18]

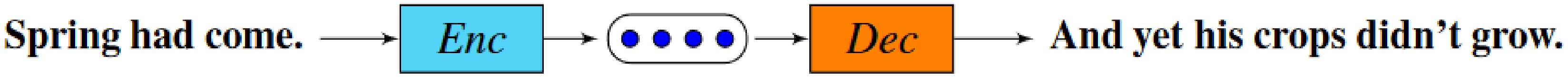
- **Variational autoencoder** $-L_{\theta;\phi}(s,s') = -E_{q_\phi(z|s)}[\log p_\theta(s'|z)] + \kappa KL(q_\phi(z|s)||N(0,I))$



Entity Blocking – QT [Logeswaran et al., ICLR'18]

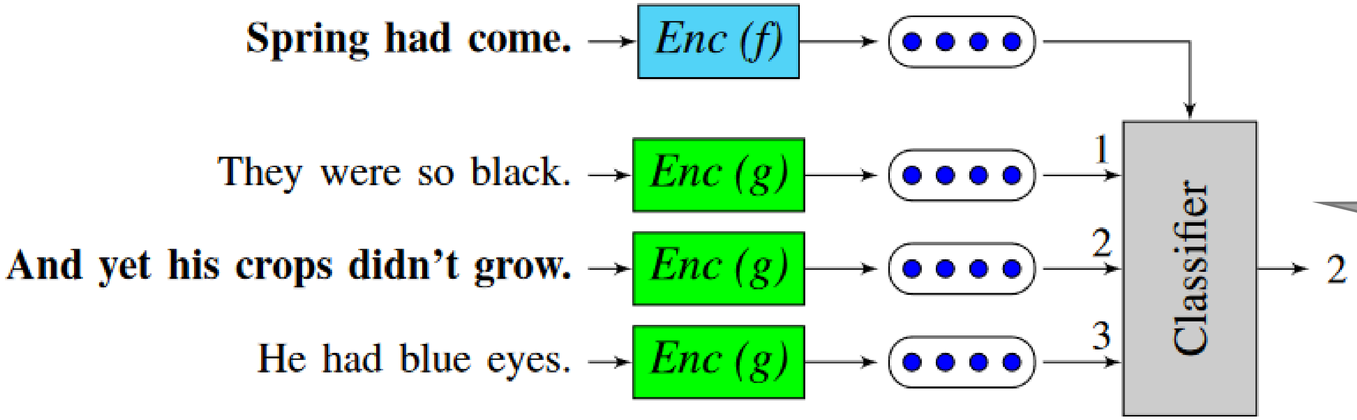
- Learning sentence representations
- Replace the decoder with a **classifier**

$$p(s_{\text{cand}}|s, S_{\text{cand}}) = \frac{\exp[c(f(s), g(s_{\text{cand}}))]}{\sum_{s' \in S_{\text{cand}}} \exp[c(f(s), g(s'))]}$$



$$\sum_{s \in D} \sum_{s_{\text{ctxt}} \in S_{\text{ctxt}}} \log p(s_{\text{ctxt}}|s, S_{\text{cand}})$$

(a) Conventional approach



(b) Proposed approach

Predict the next sentences OR the similar ones using CTT

Entity Blocking – Fast Query Processing

- **Problem**: Given two large relational tables A and B, and **Representation model Repr()**, how to efficiently find all satisfied tuple pairs ?
- **Algorithm**：

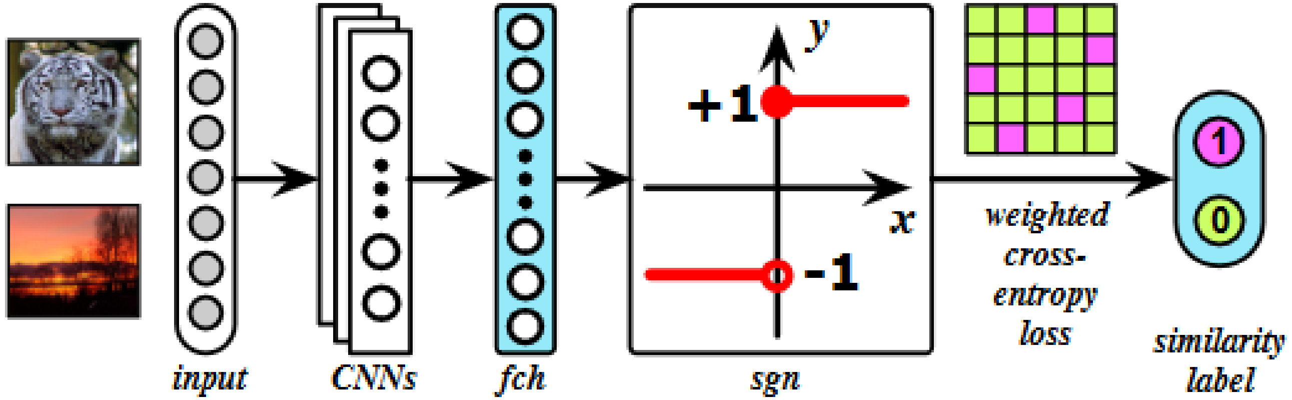
  Step 1. Transform each tuple to the embedding use Repr().

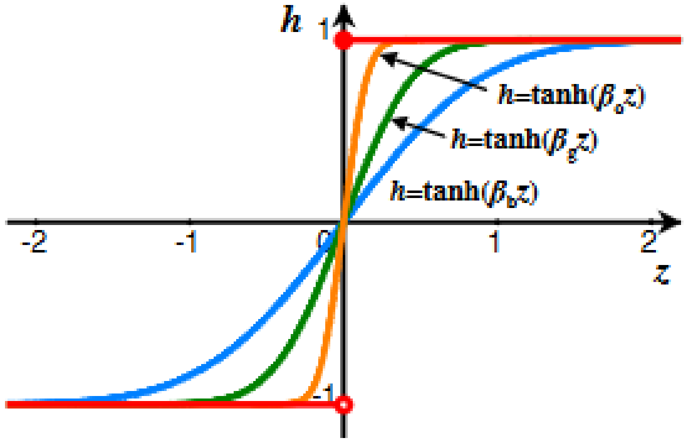
  Step 2. Cosine similarity Join between A and B

    - Locality-Sensitive-Hashing (LSH)
    - Product Quantization (PQ)
    - Faiss, Annoy, Hnswlib, etc.

Entity Blocking – Learn to hash

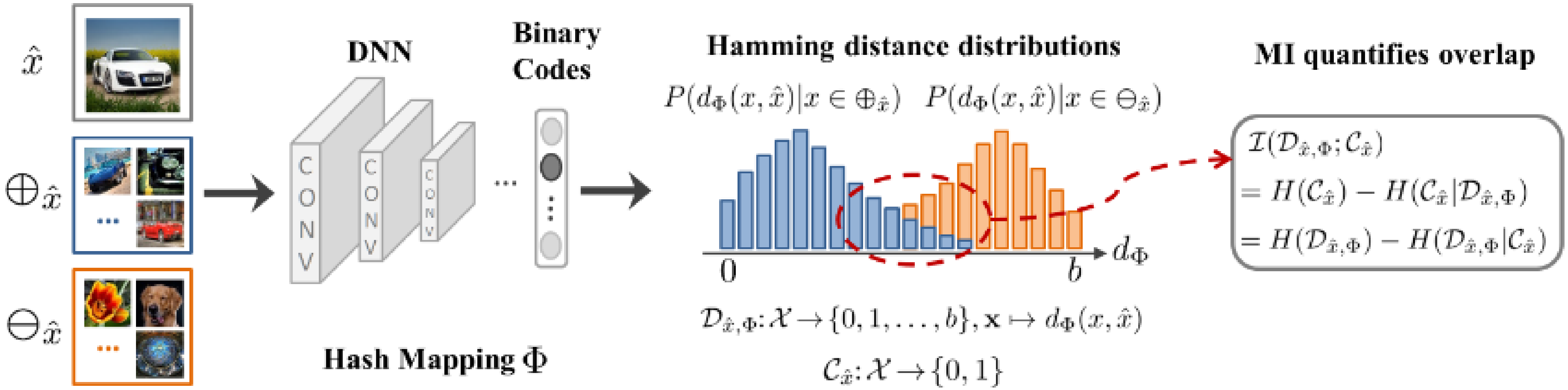
- Instead of tuple embedding, learn a high-dimensional binary vector
- Widely adopted in **CV**
- **Case I:** HashNet [Cao et al. CVPR17']
  - TanH activation function
  - Learning with Continuation





Entity Blocking – Learn to hash

- Instead of tuple embedding, learn a high-dimensional binary vector
- Widely adopted in **CV**
- **Case II**: MIHash [Cakir et al. PAMI18']


$\hat{x}$
$\oplus_{\hat{x}}$
$\ominus_{\hat{x}}$
DNN
CONV
CONV
CONV
Binary Codes
Hash Mapping Φ
Hamming distance distributions
$P(d_\Phi(x,\hat{x})|x \in \oplus_{\hat{x}})$
$P(d_\Phi(x,\hat{x})|x \in \ominus_{\hat{x}})$
0
b
d_Φ
$\mathcal{D}_{\hat{x},\Phi}: \mathcal{X} \to \{0,1,\ldots,b\}, \mathbf{x} \mapsto d_\Phi(x,\hat{x})$
$\mathcal{C}_{\hat{x}}: \mathcal{X} \to \{0,1\}$
MI quantifies overlap
$\mathcal{I}(\mathcal{D}_{\hat{x},\Phi};\mathcal{C}_{\hat{x}})$
$= H(\mathcal{C}_{\hat{x}}) - H(\mathcal{C}_{\hat{x}}|\mathcal{D}_{\hat{x},\Phi})$
$= H(\mathcal{D}_{\hat{x},\Phi}) - H(\mathcal{D}_{\hat{x},\Phi}|\mathcal{C}_{\hat{x}})$

Entity Blocking – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI’21]

Matching-aware Blocking

❑ Learnable hashing: effective than key-based methods and LSH

$$\mathrm{H}(t) = \mathrm{sign}(t\mathrm{X})$$

X: learnable hyperplanes

❑ Signum function is not differentiable → L2 Relaxation: replace the binary constraint with a regularizer

$$\mathrm{H}(t) = \mathrm{sign}(t\mathrm{X}) \Rightarrow \mathrm{H}^r(t) = t\mathrm{X}$$

❑ Loss function for blocking

$$\mathrm{L}_B^r = \frac{1}{2} y \,\|\, \mathrm{H}^r(t_i), \mathrm{H}^r(t_j) \,\|_2 + \frac{1}{2}(1-y)\max(m - \|\, \mathrm{H}^r(t_i), \mathrm{H}^r(t_j) \,\|_2, 0) + \gamma(\|\,|\mathrm{H}(t_i)| - 1\,\|_1 + \|\,|\mathrm{H}(t_j)| - 1\,\|_1),$$

L2 distance

Contrastive loss: prevent very dissimilar pairs from the computation

Regularizer for binary constraint

Entity Blocking – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI’21]

Matching-aware Blocking

❑ Hyperplanes Orthogonalization: ensure independency of hash bits and being isometry

➢ Regularization-based approach

$$R_o = \| \mathrm{X}\mathrm{X}^{\top} - \mathbf{I} \|_F$$

➢ SVD-based approach: decompose X using SVD, and replace X with orthogonal matrix US

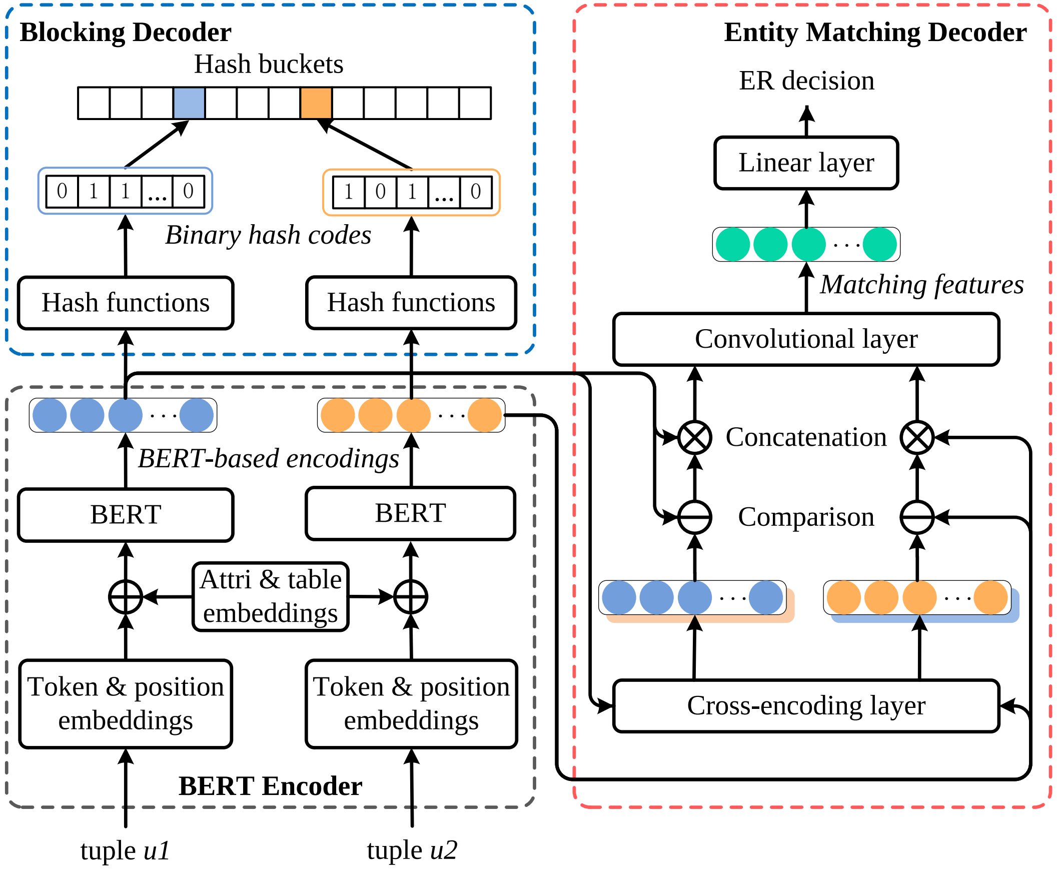
$$\mathrm{SVD}(\mathrm{X}) = USV^{\top}$$

$$\mathrm{X}' \leftarrow US$$

Entity Blocking – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

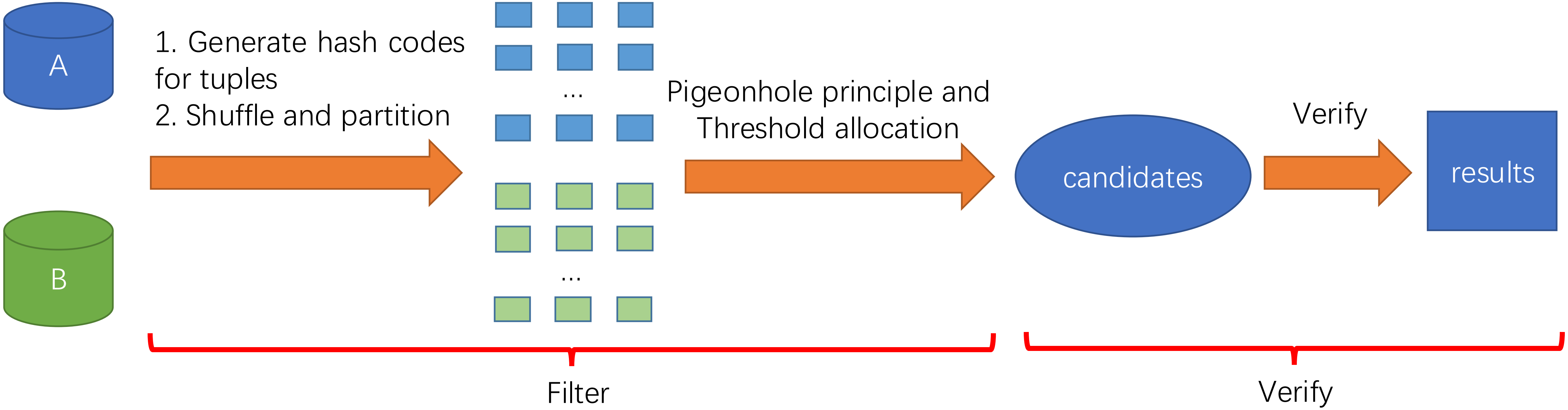
Final framework

- ❑ The base is the BERT encoder, shared by two task-specific decoders -- blocking and entity matching.



Entity Blocking – Fast Query Processing

- **Problem**: Given two large relational tables A and B, and ***learn to hash* model Repr()**, how to efficiently find all satisfied tuple pairs ?
- **Algorithm**: GPH [Qin et al. ICDE18', TKDE20']


A
B
1. Generate hash codes for tuples
2. Shuffle and partition
...
...
Pigeonhole principle and Threshold allocation
candidates
Verify
results
Filter
Verify

Entity Blocking

		Learning strategy	Schema-aware	# of instances	Accuracy	Running Time
Rule-based	ApproxDNF [ICDM06']	Supervised	Yes	Few	Not high	Moderate
	BSL/BSL+[AAAI06', IJCAI11]	Supervised	Yes	Few	Not high	Moderate
	Fisher [ICDM13']	Unsupervised	Yes	None	Not high	Moderate
	EM-GBF [VLDB17']	Supervised	Yes	A Few	Moderate	Moderate
	DNF-BSL [2015]	Unsupervised	No	None	Not high	Moderate
	HyMD [TODS20']	Mining	Yes	A few	Moderate	Moderate
ML-based	Smurf [VLDB19']	Supervised	Yes	A few	High	Not fast
	Meta-Blocking [VLDB14']	Supervised	No	A few	Moderate	Moderate

Entity Blocking

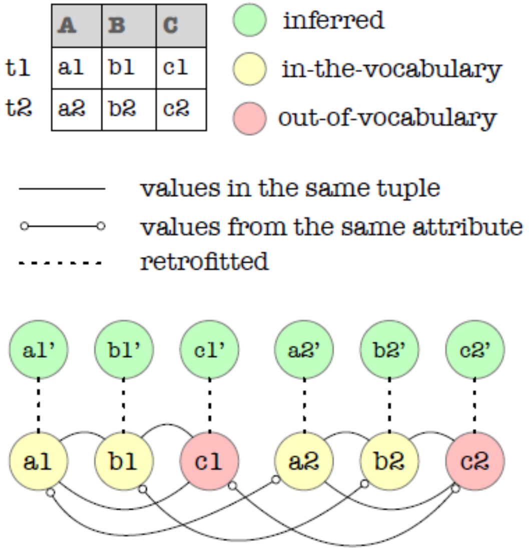
		Learning strategy	# of instances	Accuracy	Pairwise	Running Time
DL	DeepER [VLAB18']	Supervised	A lot	High	LSH	Fast (k-NN)
	Sbert [EMNLP19']	Supervised	A lot	High	Cosine	Fast (k-NN)
	Autoencoder / Trans-encoder [VLDB21']	Unsupervised	None	Moderate	Cosine	Fast (k-NN)
	CSAE [ACL16'] + cosine	Semi-supervised	A few	Moderate	Cosine	Fast (k-NN)
	SIF, CTT(-Cosine), Hybrid [VLDB21]	Supervised	A lot	High	Cosine	Fast (k-NN)
	QT [ICLR18'] + CTT	Supervised	A lot	High	Cosine	Fast (k-NN)
	VAR-Siamese [NeurIPS18'] + CTT	Semi-supervised	A few	Moderate	Cosine	Fast (k-NN)
	Bert-ER [AAAI21']	Supervised	A lot	High	Hamming	Very fast (k-NN, threshold)

Entity Matching – Problem Definition

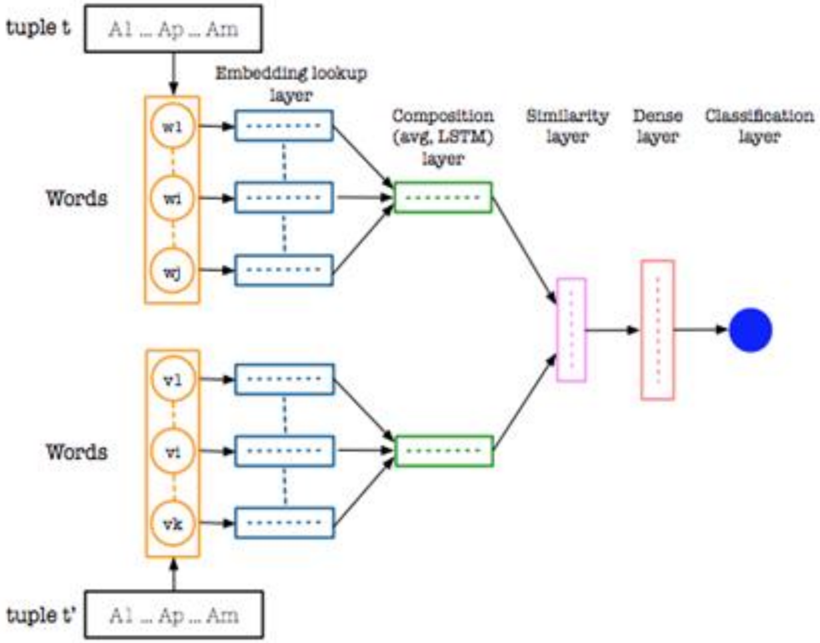
- **Problem Definition**: Fine-comparing (after blocking) tuple pairs to find co-references, i.e., binary classification problem.
- **Evaluation**
  - Precision $P=TP/(TP+FP)$
  - Recall $R=TP/(TP+FN)$
  - F1 $F1=2*P*R/(P+R)$

Entity Matching – DeepER [Ebraheem et al., VLDB'18]

- First DL-based ER model
- **Interaction:** Attribute comparison
- **Comparator:** Cosine
- **Encoder:** LSTM
- **Embedding:** GloVe
- For OOV word --- Vocabulary Retrofitting



	A	B	C
t1	a1	b1	c1
t2	a2	b2	c2

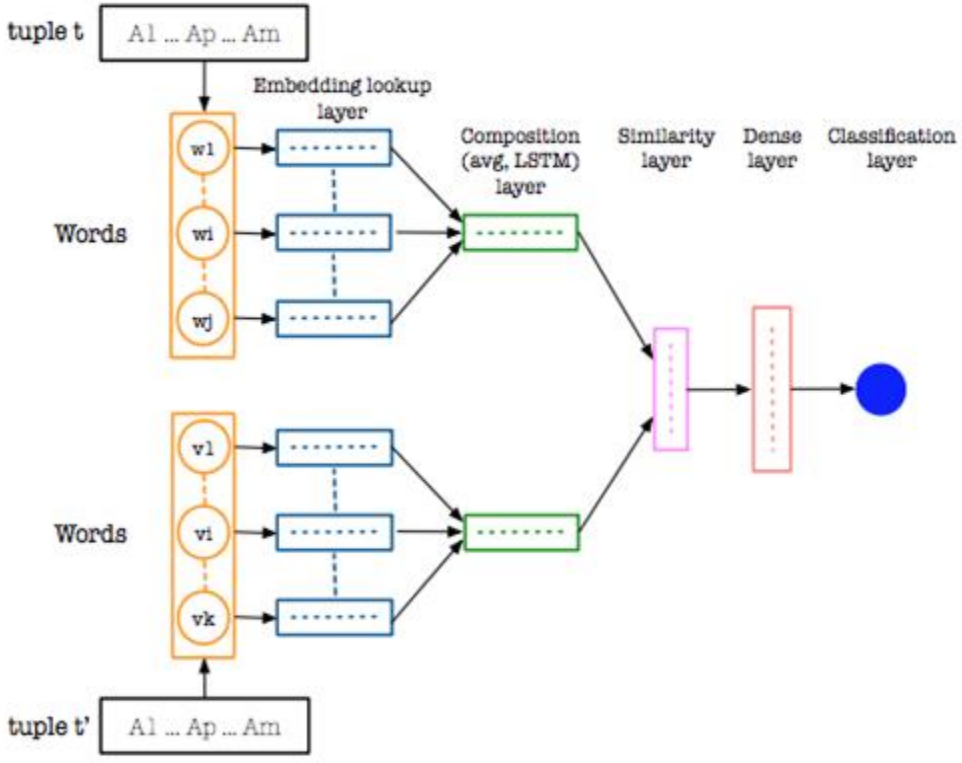


Entity Matching – DeepER [Ebraheem et al., VLDB'18]

- First DL-based ER model
- **Interaction:** Attribute comparison
- **Comparator:** Cosine
- **Encoder:** LSTM
- **Embedding:** Glove
- Outperforming SOTA non-deep solution Magellan with a big margin

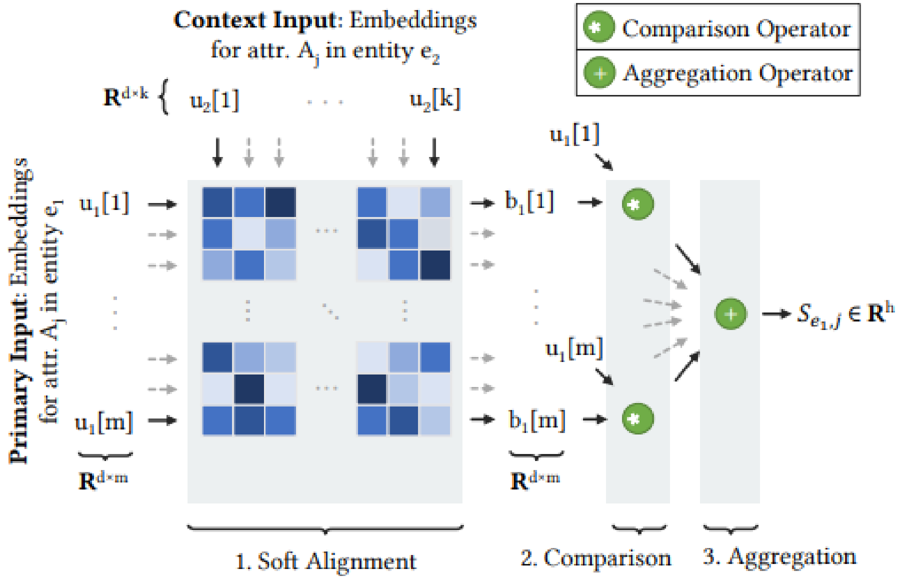
Performance

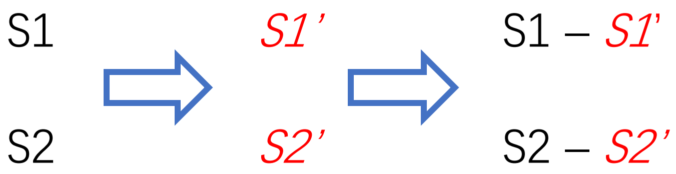
- **F-1: >96% on *Amazon-Google Dataset* w. 1,300 positive cases**
- **Magellan F-1:87.68% (~10 pts gap)**

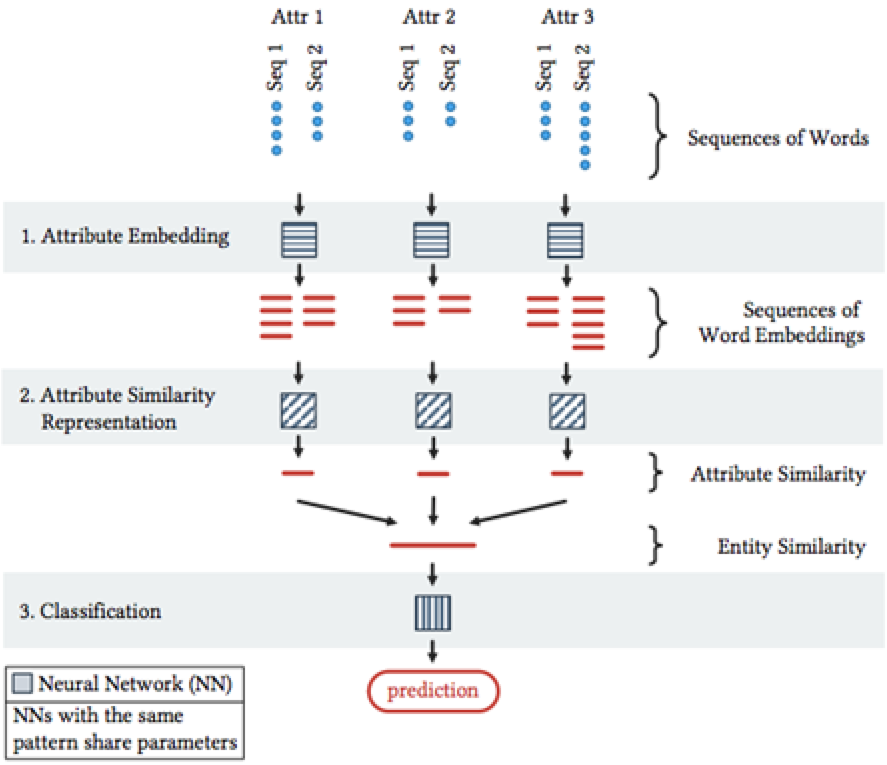


Entity Matching – DeepMatcher [Mudgal et al., SIGMOD’18]

- **Interaction:** Cross-encoded attribute comparison





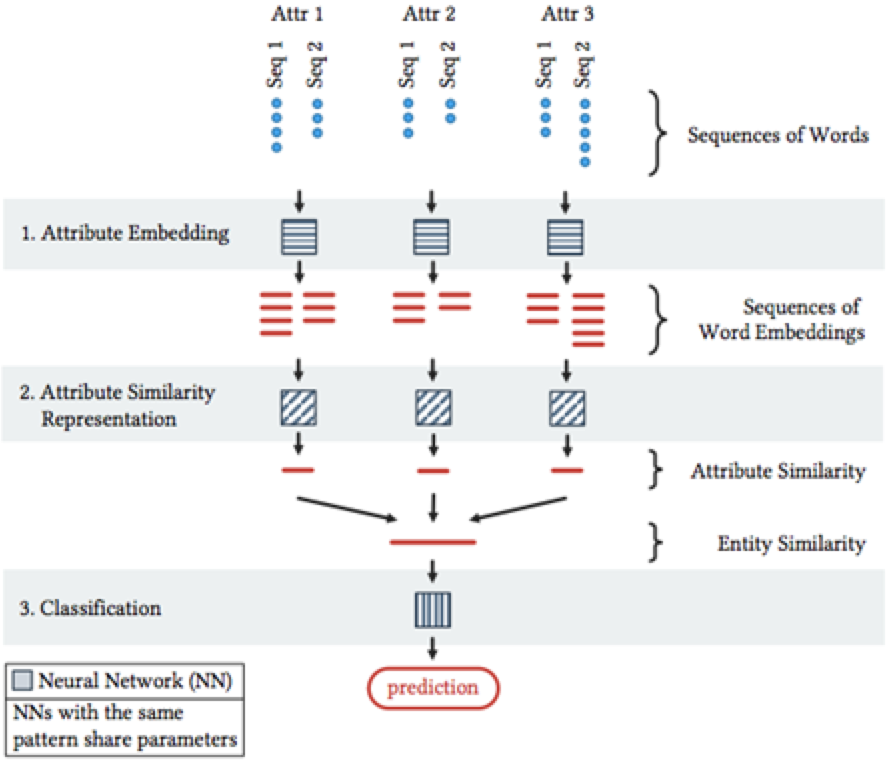


Entity Matching – DeepMatcher [Mudgal et al., SIGMOD'18]

- **Interaction:** Cross-encoded attribute comparison
- **Comparator:** Subtraction
- **Encoder:** RNN, LSTM
- **Embedding:** fastText (no big differences w. GloVe)
- Outperforming SOTA non-deep solution Magellan with a big margin

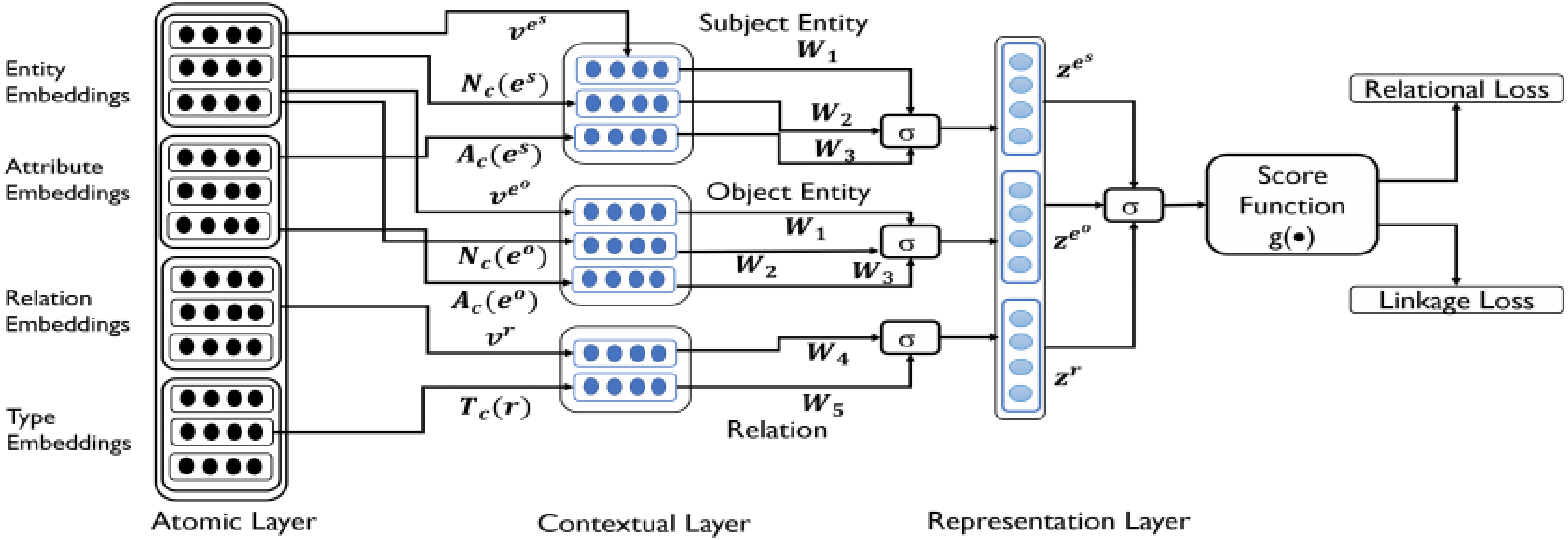
Performance

- **F-1: >69.3% on *Amazon-Google (refined)* w. 1,300 positive cases**
- **Magellan F-1:49.1% (~20 pts gap)**



Deep Learning Models [Trivedi et al., ACL’18]

- LinkNBed: Embeddings for entities as in knowledge embedding

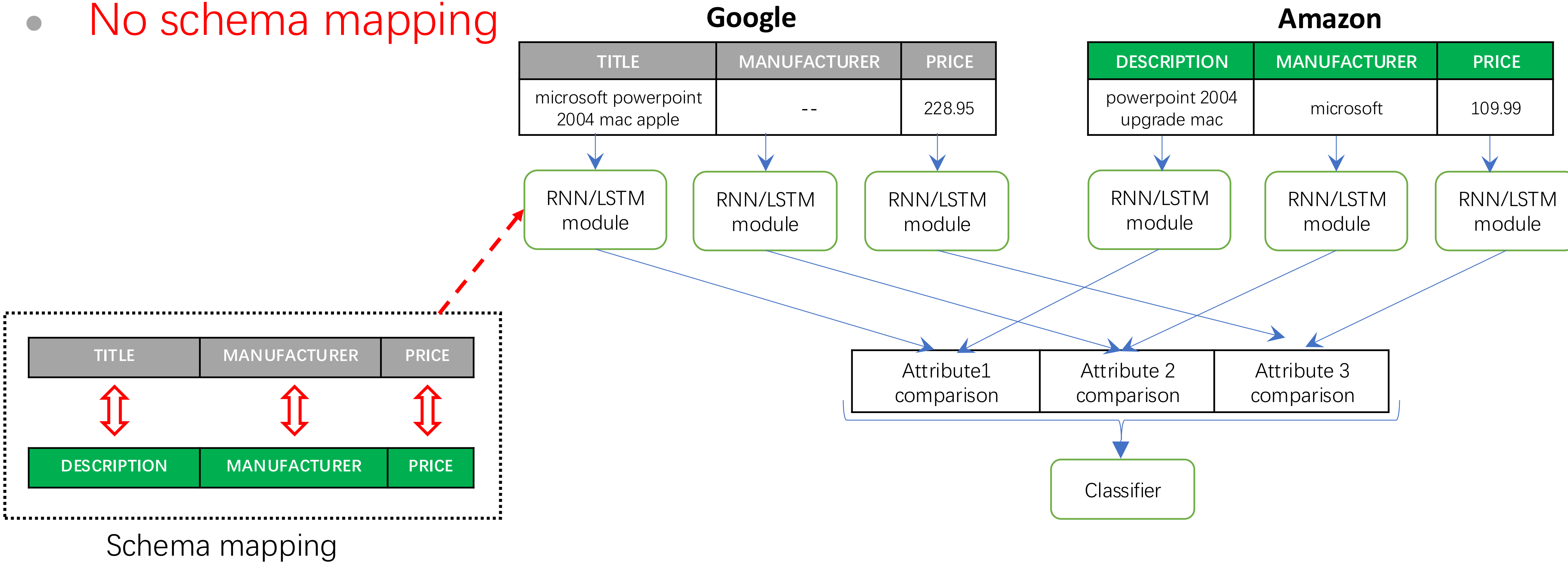


Deep Learning Models [Trivedi et al., ACL'18]

- LinkNBed: Embeddings for entities as in knowledge embedding
- Performance better than previous knowledge embedding methods, but not comparable to random forest
- Enable linking different types of entities

Entity Matching – GraphER [Bing Li, Wei Wang, et al, AAAI'20]

- **Interaction:** Graph-encoded token comparison
- No schema mapping



Google

TITLE	MANUFACTURER	PRICE
microsoft powerpoint 2004 mac apple	--	228.95

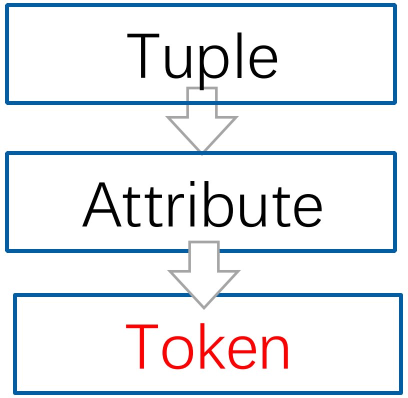
Amazon

DESCRIPTION	MANUFACTURER	PRICE
powerpoint 2004 upgrade mac	microsoft	109.99

Schema mapping

Entity Matching – GraphER [Bing Li, Wei Wang, et al, AAAI'20]

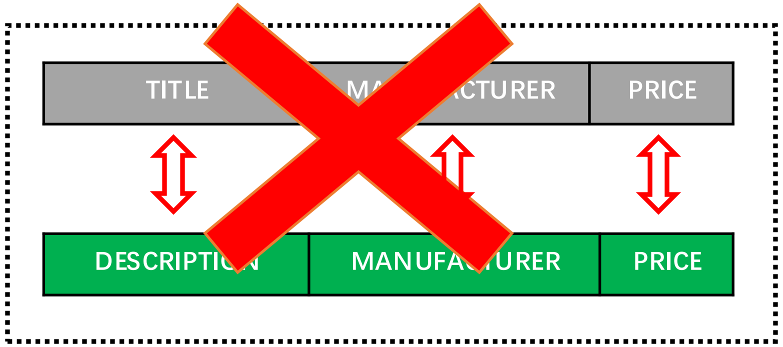
- **Interaction:** Graph-encoded token comparison
  - No schema mapping
  - Finer-grained
  - Share information between attributes



Tuple

Attribute

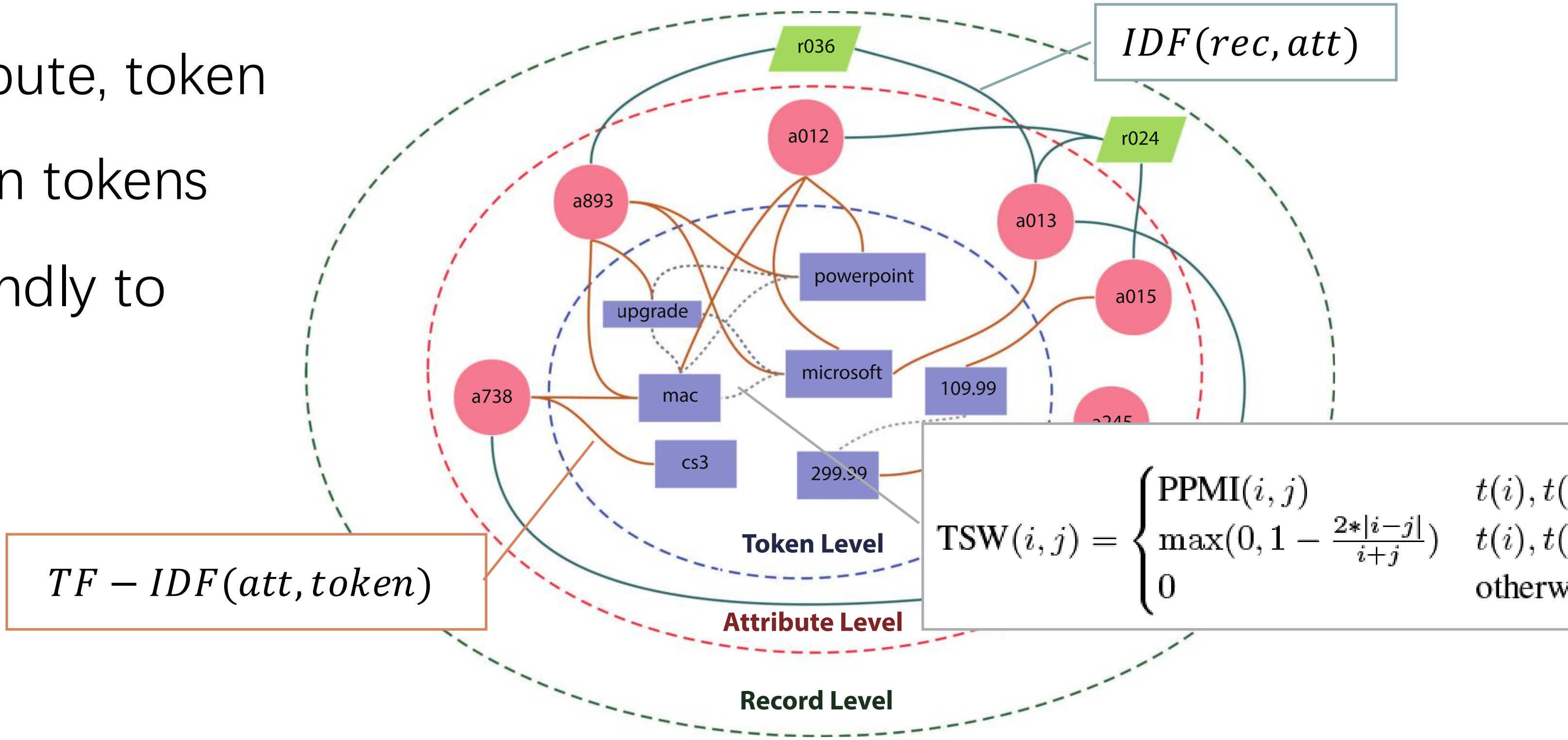
Token



Schema mapping

Entity Matching – GraphER [Bing Li, Wei Wang, et al, AAAI’20]

- ER-Graph
  - Inclusion of tuple, attribute, token
  - Co-occurrence between tokens
  - Type sensitive – be friendly to numerical values
- Two-layer GCN



$IDF(rec, att)$

$TF - IDF(att, token)$

$$\mathrm{TSW}(i,j) = \begin{cases} \mathrm{PPMI}(i,j) & t(i), t(\\ \max(0, 1 - \frac{2*|i-j|}{i+j}) & t(i), t(\\ 0 & \text{otherw} \end{cases}$$

$$E = \mathrm{ReLU}(\widetilde{A}\,\mathrm{ReLU}(\widetilde{A} I \Theta^{(1)})\Theta^{(2)})$$

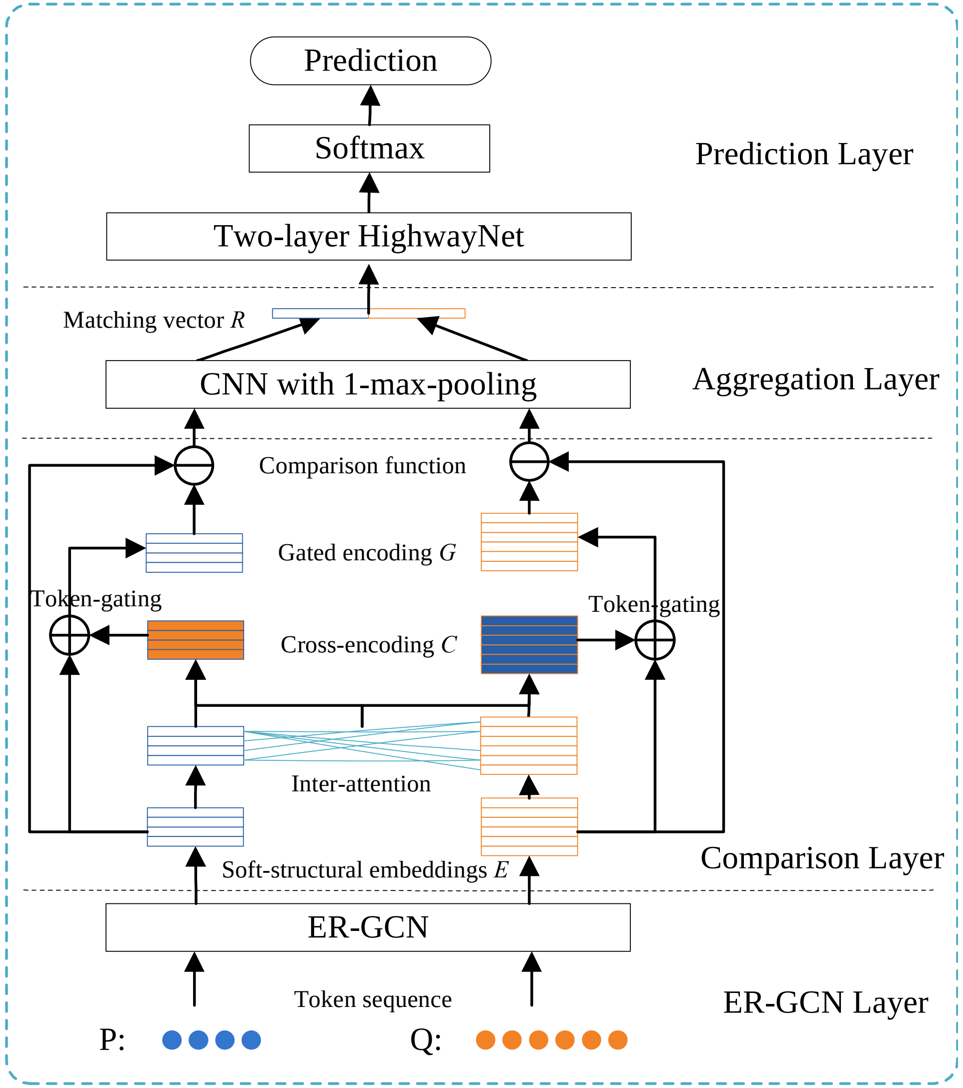
Entity Matching – GraphER [Bing Li, Wei Wang, et al, AAAI’20]

- **Interaction:** Graph-encoded Token
- **Comparator:** Subtraction
- **Encoder:** GCN
- **Embedding:** Glove or learn from scratch
- Aggregation Layer
  - bilateral matching [Wang et al, ICLR 2017]

$$r^{(P\to Q)} = \mathrm{CNN}(M^{(P\to Q)})$$

$$R = [r^{(P\to Q)}; r^{(Q\to P)}]$$

- Prediction layer
  - two-layer dense HighwayNet

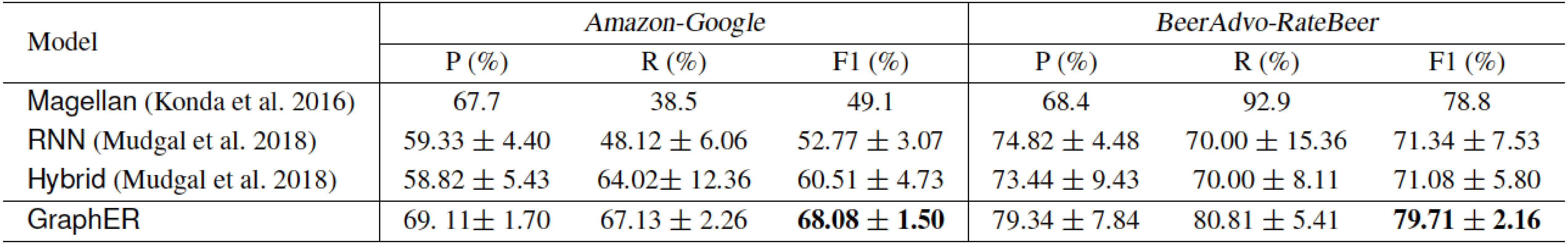


Entity Matching – GraphER [Bing Li, Wei Wang, et al, AAAI'20]

- **Interaction:** Graph-encoded token comparison
- **Encoder:** GCN
- **Embedding:** Glove or learn from scratch

Performance

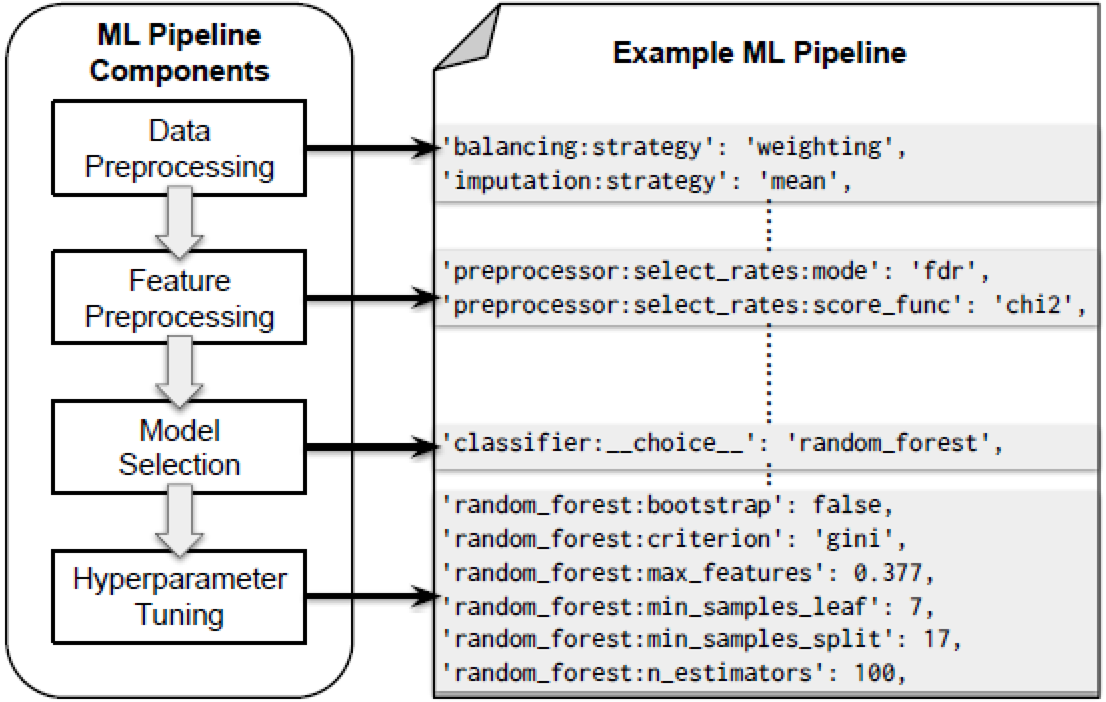
- **F-1: >68% avg on *Amazon-Google (refined)* w. 1,300 positive cases**
- **DeepMatcher F-1:60% avg (~8 pts gap)**



Model	*Amazon-Google*			*BeerAdvo-RateBeer*		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Magellan (Konda et al. 2016)	67.7	38.5	49.1	68.4	92.9	78.8
RNN (Mudgal et al. 2018)	59.33 ± 4.40	48.12 ± 6.06	52.77 ± 3.07	74.82 ± 4.48	70.00 ± 15.36	71.34 ± 7.53
Hybrid (Mudgal et al. 2018)	58.82 ± 5.43	64.02± 12.36	60.51 ± 4.73	73.44 ± 9.43	70.00 ± 8.11	71.08 ± 5.80
GraphER	69. 11± 1.70	67.13 ± 2.26	**68.08 ± 1.50**	79.34 ± 7.84	80.81 ± 5.41	**79.71 ± 2.16**

Entity Matching – AutoML-EM [Wang, Pei, et al., ICDE'21]

- **Main idea:** hand-off EM
  - Treat EM pipeline development as a solvable search problem with AutoML
- **Interaction:** Tuple features comparison
- **Backbone:** AutoML
- **Searching Algorithm**
  - Input: search space (e.g., a set of components); evaluation metric (e.g., F1); a time budget
  - Output: the best pipeline



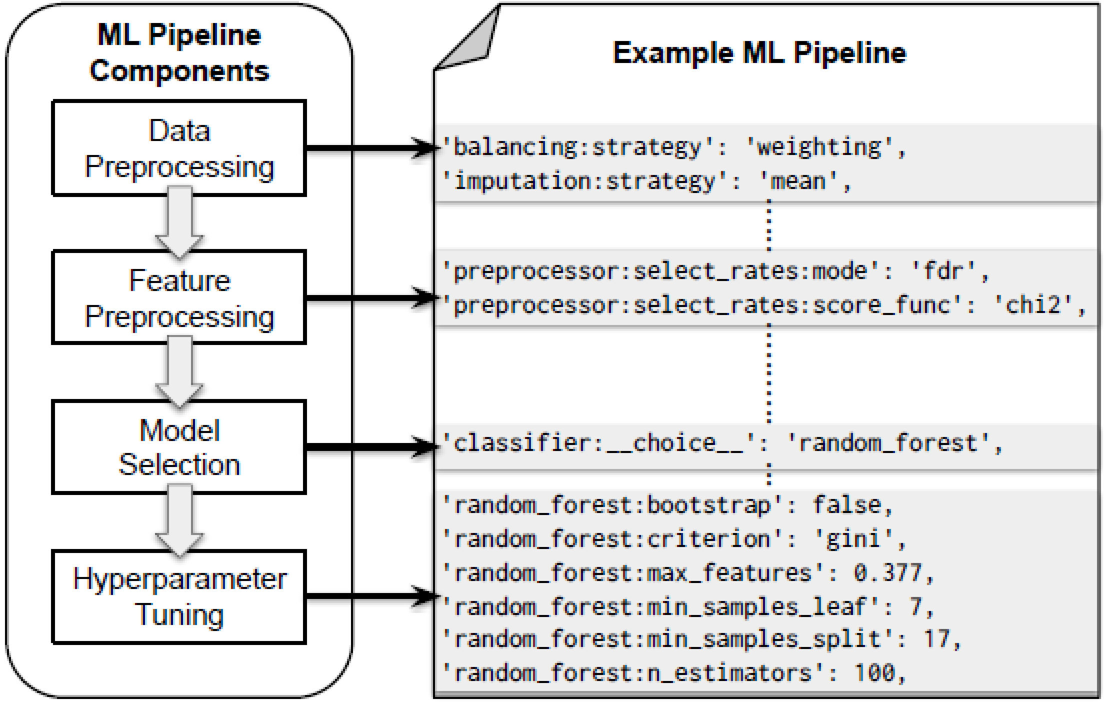
```
'balancing:strategy': 'weighting',
'imputation:strategy': 'mean',
                    ⋮
'preprocessor:select_rates:mode': 'fdr',
'preprocessor:select_rates:score_func': 'chi2',
                    ⋮
'classifier:__choice__': 'random_forest',
                    ⋮
'random_forest:bootstrap': false,
'random_forest:criterion': 'gini',
'random_forest:max_features': 0.377,
'random_forest:min_samples_leaf': 7,
'random_forest:min_samples_split': 17,
'random_forest:n_estimators': 100,
```

Entity Matching – AutoML-EM [Wang, Pei, et al., ICDE'21]

- **Active Labelling**
  - Human-in-the-loop
  - In each round, selects a set of unlabeled pairs with lowest confidence scores and asks humans to label them

Performance

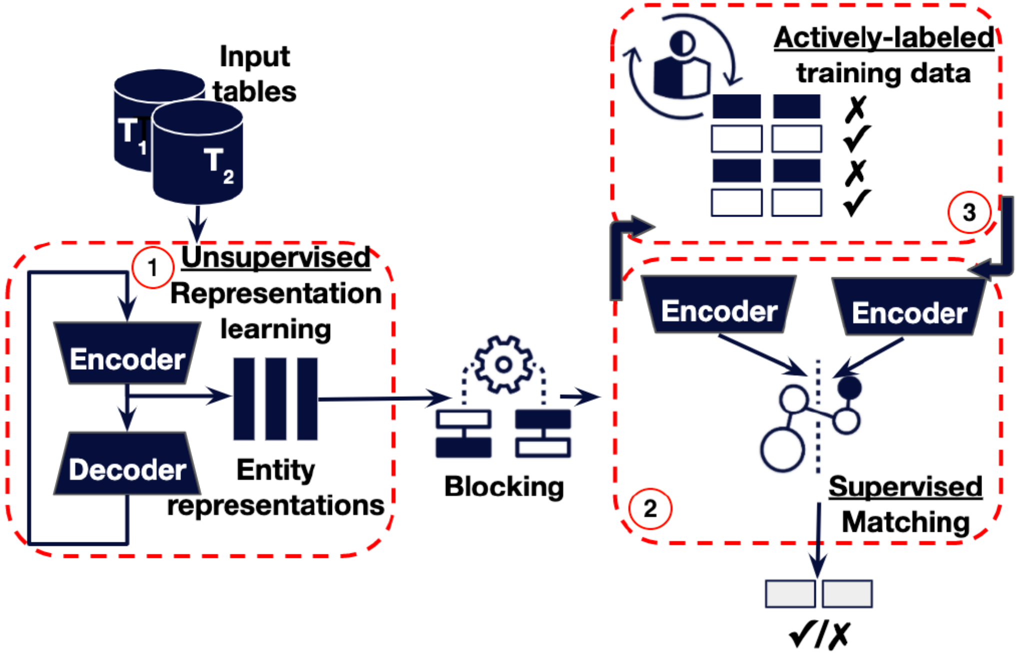
- **F-1: 66.4% on *Amazon-Google (refined)* w. 1,300 positive cases**
- **DeepMatcher F-1:69.3%**



```
'balancing:strategy': 'weighting',
'imputation:strategy': 'mean',

'preprocessor:select_rates:mode': 'fdr',
'preprocessor:select_rates:score_func': 'chi2',

'classifier:__choice__': 'random_forest',

'random_forest:bootstrap': false,
'random_forest:criterion': 'gini',
'random_forest:max_features': 0.377,
'random_forest:min_samples_leaf': 7,
'random_forest:min_samples_split': 17,
'random_forest:n_estimators': 100,
```

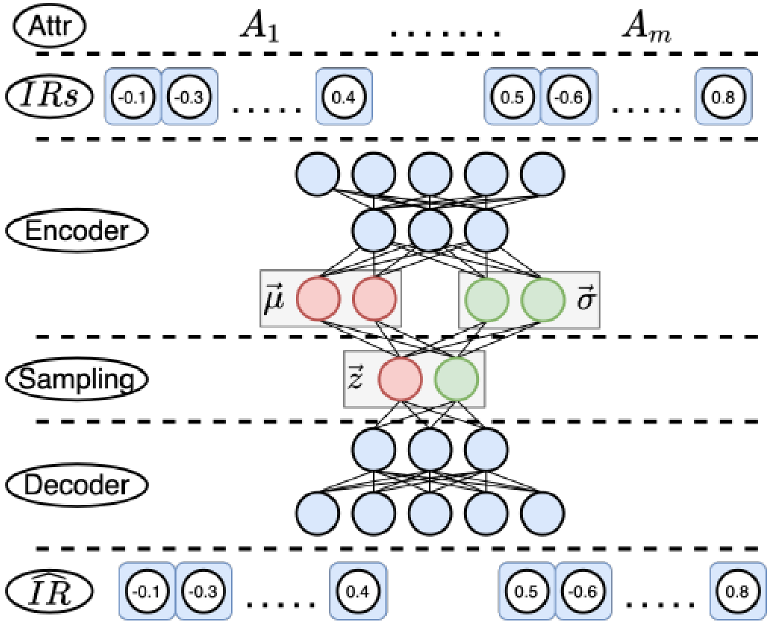
Entity Matching – VAER [Bogatu, Alex, et al, ICDE'21]

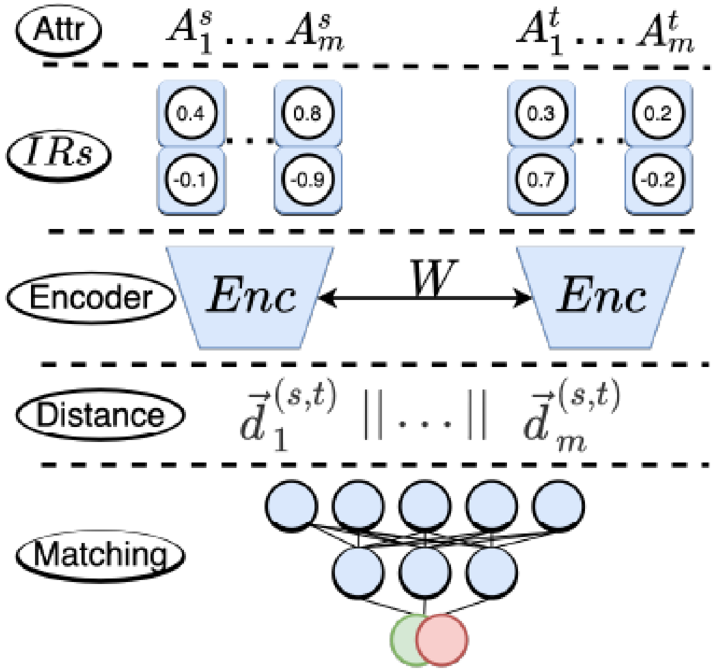
- **Interaction:** Tuple comparison
- **Comparator:** 2–Wasserstein distance
- **Encoder:** Variational Auto-Encoders (VAE)
- **Embedding:** LSA (Latent semantic analysis)



Entity Matching – VAER [Bogatu, Alex, et al, ICDE’21]

- **Interaction:** Tuple comparison
- **Comparator:** 2–Wasserstein distance
- **Encoder:** Variational Auto-Encoders (VAE)
  - Unsupervised Representation – learn compressed encoding using VAE



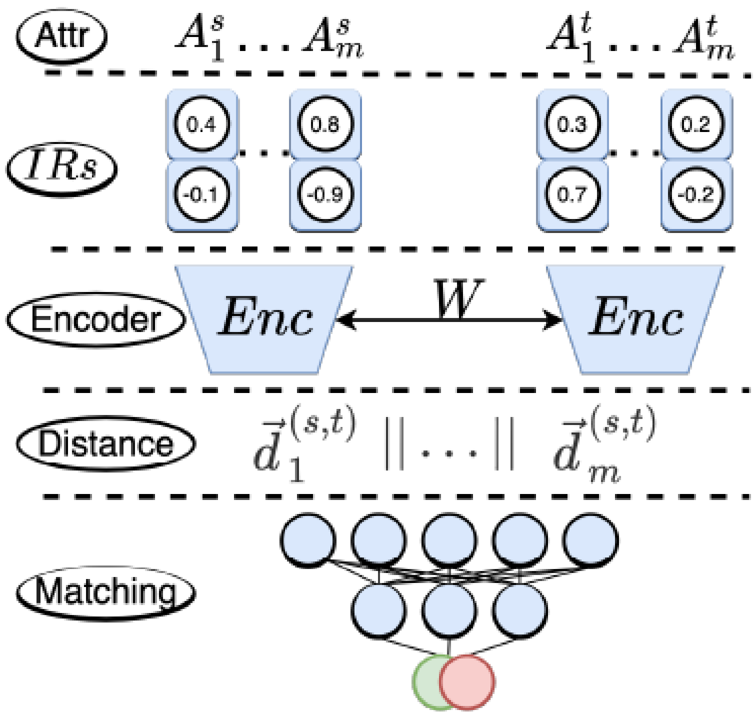


Entity Matching – VAER [Bogatu, Alex, et al, ICDE'21]

- **Interaction:** Tuple comparison
- **Comparator:** 2–Wasserstein distance
- **Encoder:** Variational Auto-Encoders (VAE)
  - Unsupervised Representation – learn compressed encoding using VAE

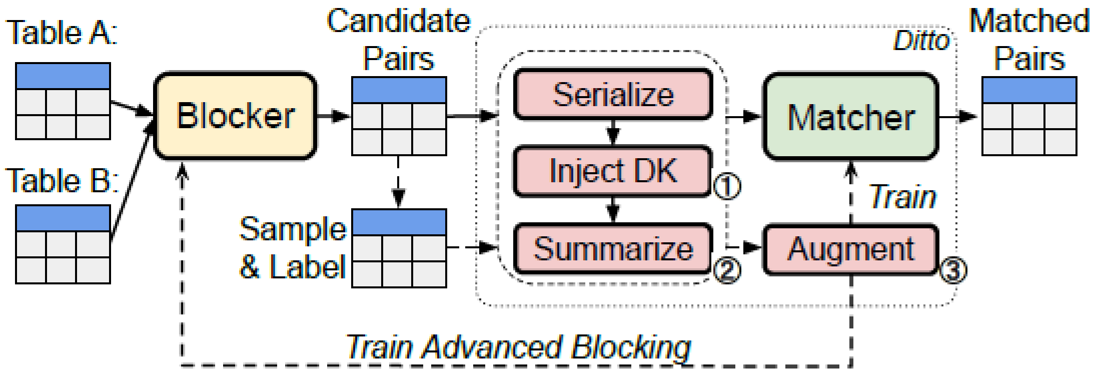
Performance

- **Reduce data labeling**
- **Achieving 90% or more F1 score with less actively labeled samples**



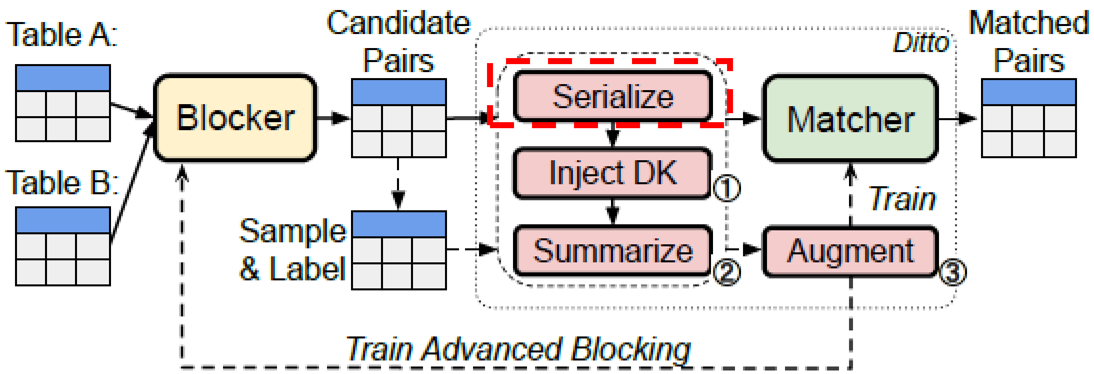
Entity Matching – DITTO [Li, Yuliang, et al., VLDB'21]

- **Interaction:** Synchronous deep interaction
- **Encoder:** Pre-trained LMs
- **Embedding:** Deeply contextualized embedding



Entity Matching – DITTO [Li, Yuliang, et al., VLDB’21]

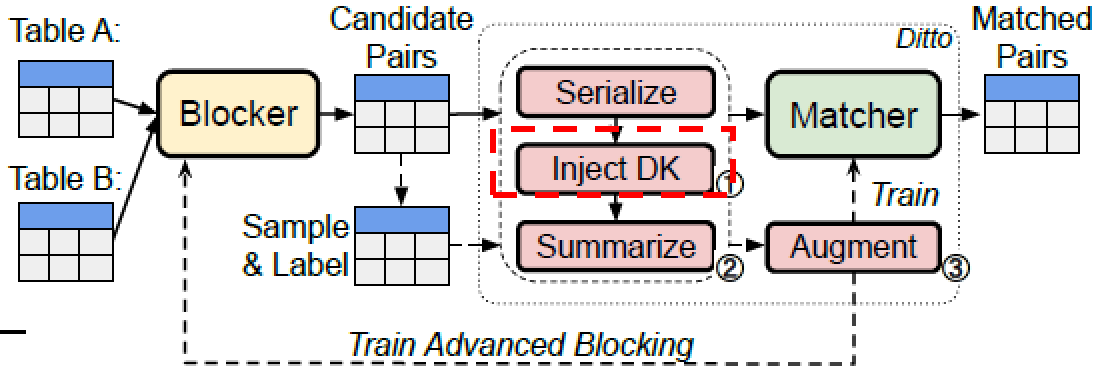
- **Serialize**
  - Special token [COL]: attribute’s name [VAL]: values
  - Pack tuple pair



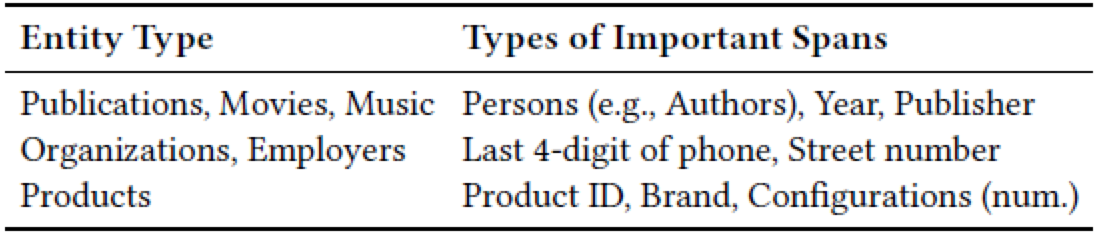
$$\text{serialize}(e) ::= [\text{COL}]\ \text{attr}_1\ [\text{VAL}]\ \text{val}_1 \ldots [\text{COL}]\ \text{attr}_k\ [\text{VAL}]\ \text{val}_k,$$

$$\text{serialize}(e, e') ::= [\text{CLS}]\ \text{serialize}(e)\ [\text{SEP}]\ \text{serialize}(e')\ [\text{SEP}],$$

Entity Matching – DITTO [Li, Yuliang, et al., VLDB'21]



- Inject Domain knowledge
  - Entity Span

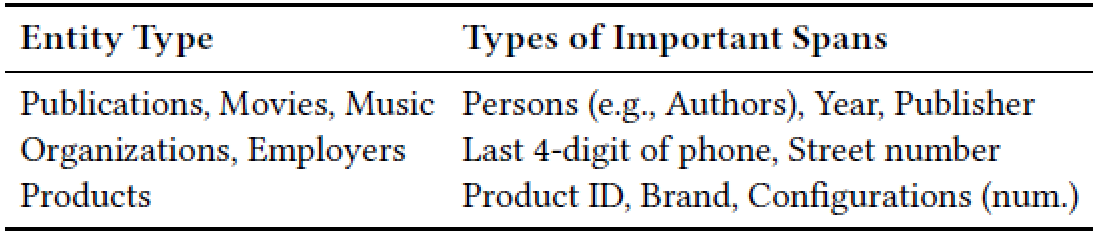


Entity Type	Types of Important Spans
Publications, Movies, Music	Persons (e.g., Authors), Year, Publisher
Organizations, Employers	Last 4-digit of phone, Street number
Products	Product ID, Brand, Configurations (num.)

  - Span Normalization
    - E.g., VLDB journal = VLDBJ

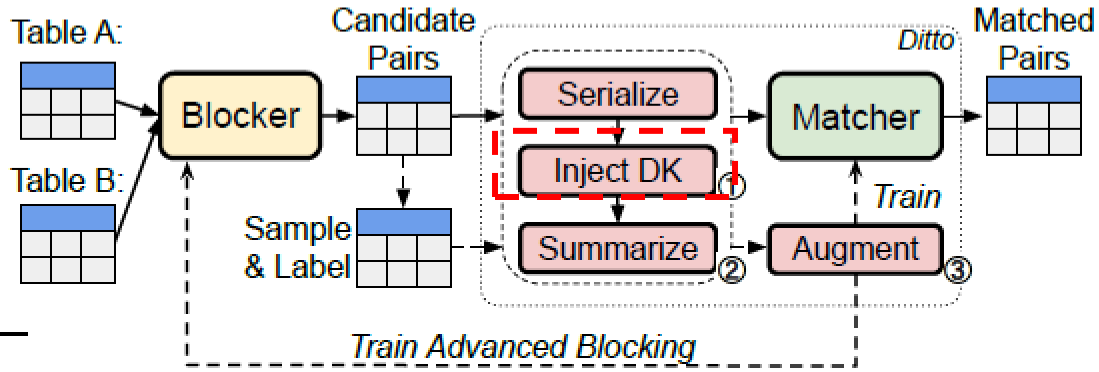
Entity Matching – DITTO [Li, Yuliang, et al., VLDB'21]

- **Inject Domain knowledge**
  - Entity Span



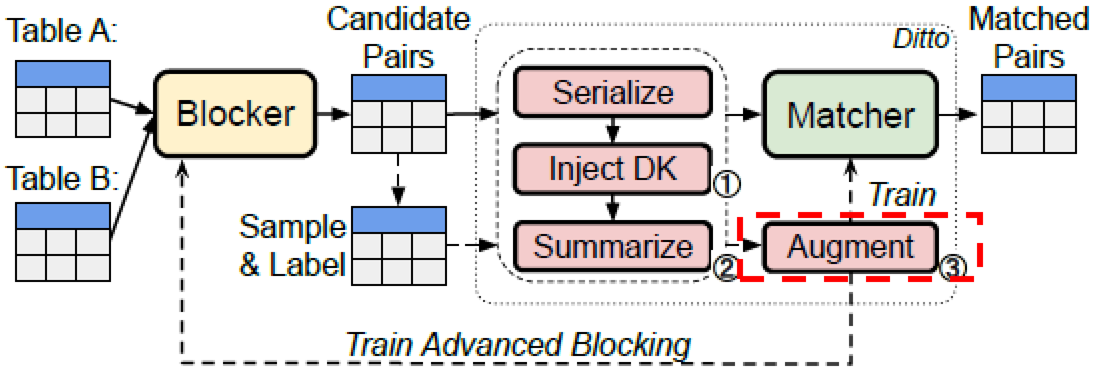
Entity Type	Types of Important Spans
Publications, Movies, Music	Persons (e.g., Authors), Year, Publisher
Organizations, Employers	Last 4-digit of phone, Street number
Products	Product ID, Brand, Configurations (num.)

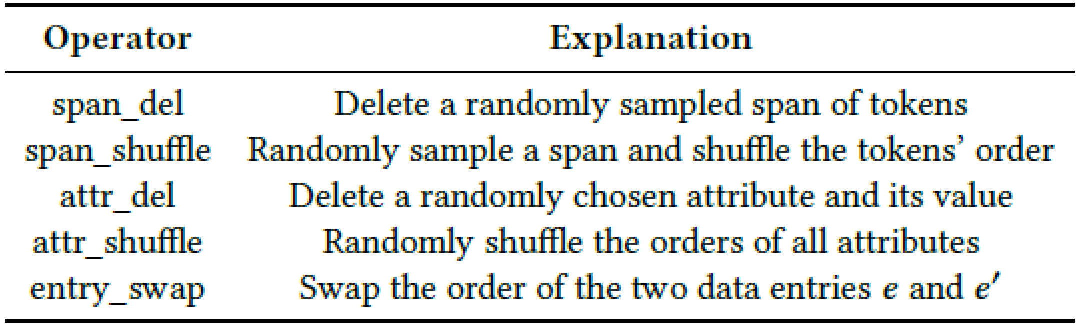
  - Span Normalization
    - E.g., VLDB journal = VLDBJ
- **Summarize**
  - Pick top-512 tokens w.r.t. TF-IDF



Entity Matching – DITTO [Li, Yuliang, et al., VLDB’21]

- **Data Augmentation (DA)**
  - More training data, more robust model

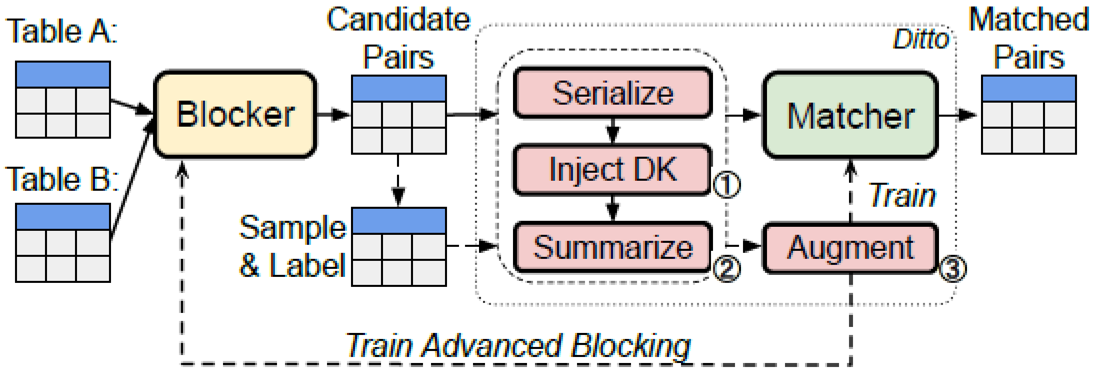




Operator	Explanation
span_del	Delete a randomly sampled span of tokens
span_shuffle	Randomly sample a span and shuffle the tokens’ order
attr_del	Delete a randomly chosen attribute and its value
attr_shuffle	Randomly shuffle the orders of all attributes
entry_swap	Swap the order of the two data entries e and e'

Entity Matching – DITTO [Li, Yuliang, et al., VLDB'21]

- **Interaction:** Synchronous deep interaction
- **Encoder:** Pre-trained LMs
- **Embedding:** Deeply contextualized embedding
- With RoBERTa as the back-bone

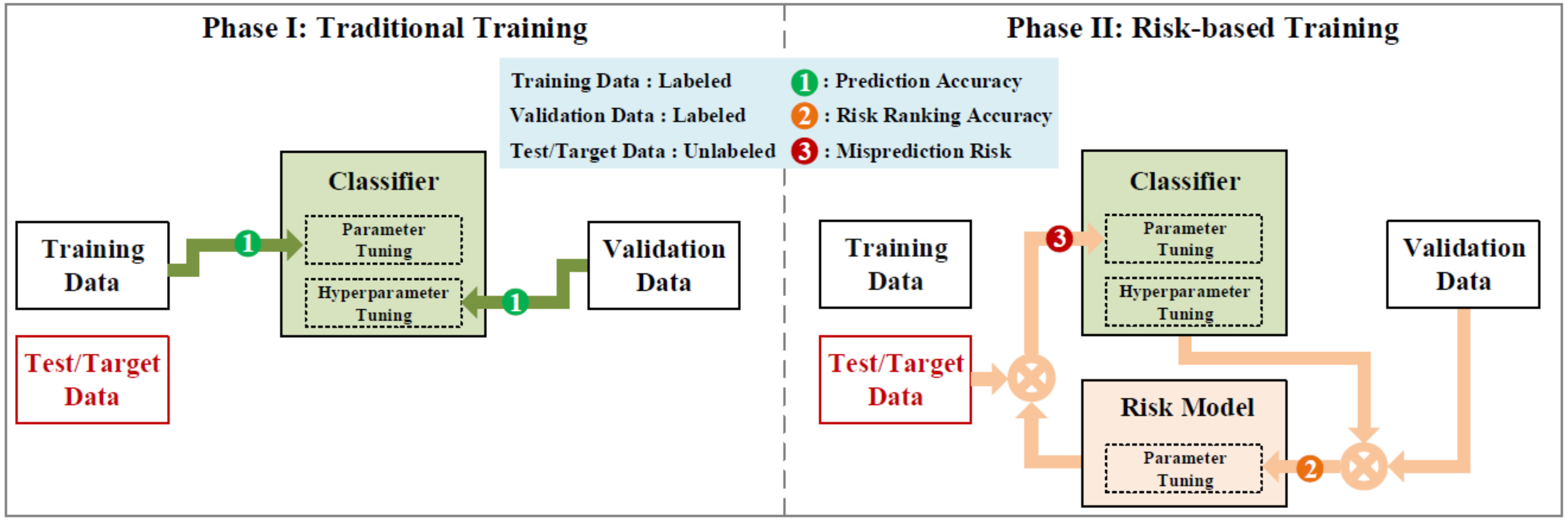


Performance

- F-1: 75.58% avg on *Amazon-Google (refined)* w. 1,300 positive cases
- DeepMatcher+ F-1:70.7% avg (~5 pts gap)

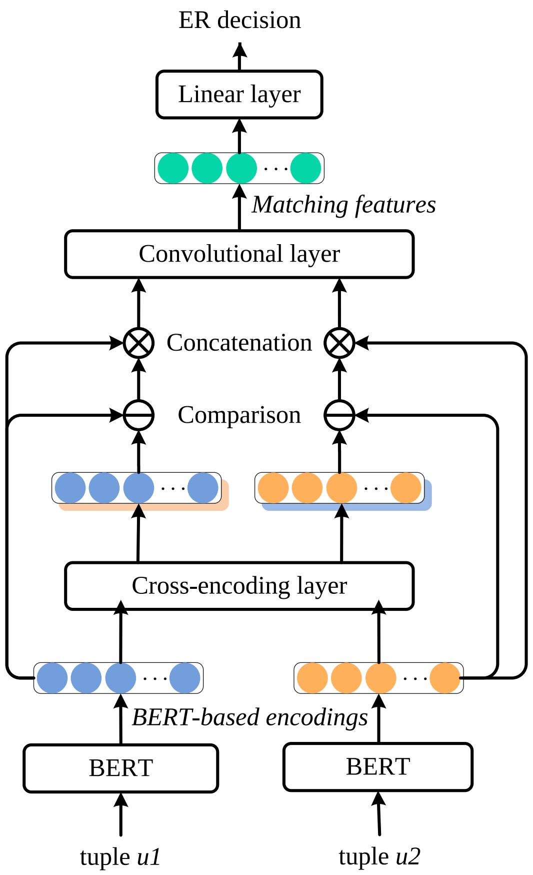
Entity Matching – Risk [Chen, Q el al, JMLR'21]

- **Main idea:** Learning classification risk (residual)
  - Similar idea for gradient boosting
- **Encoder:** DeepMatcher or DITTO (base learner)
  - Risk leaner: a simple linear layer with manual risk features (e.g, r1[year] = r2[year])
  - Outperforming base learner with only 10% to 30% training data



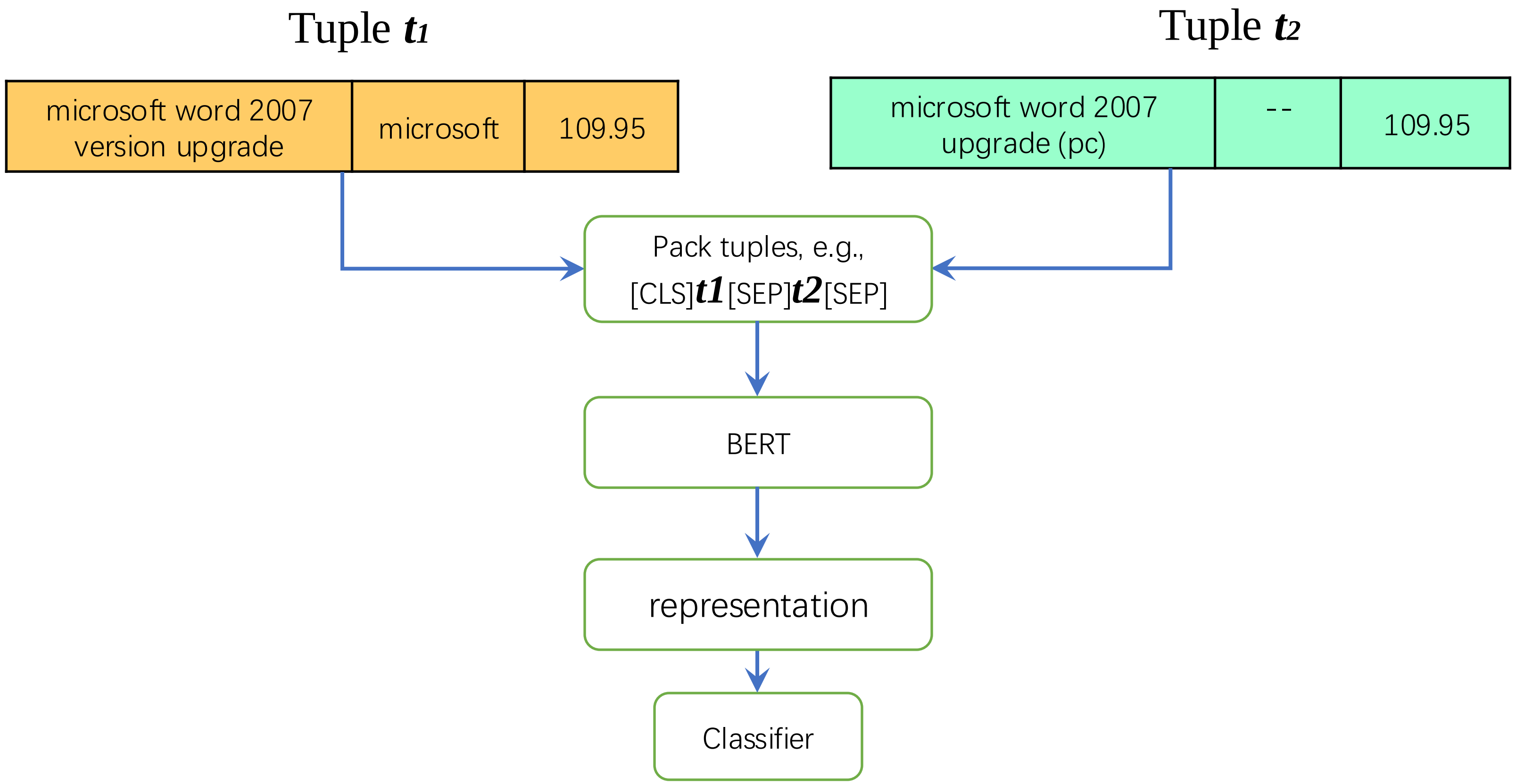
Entity Matching – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

- Current SOTA
- **Interaction:** Asynchronous deep interaction
- **Encoder:** BERT
- **Embedding:** Deeply contextualized embedding



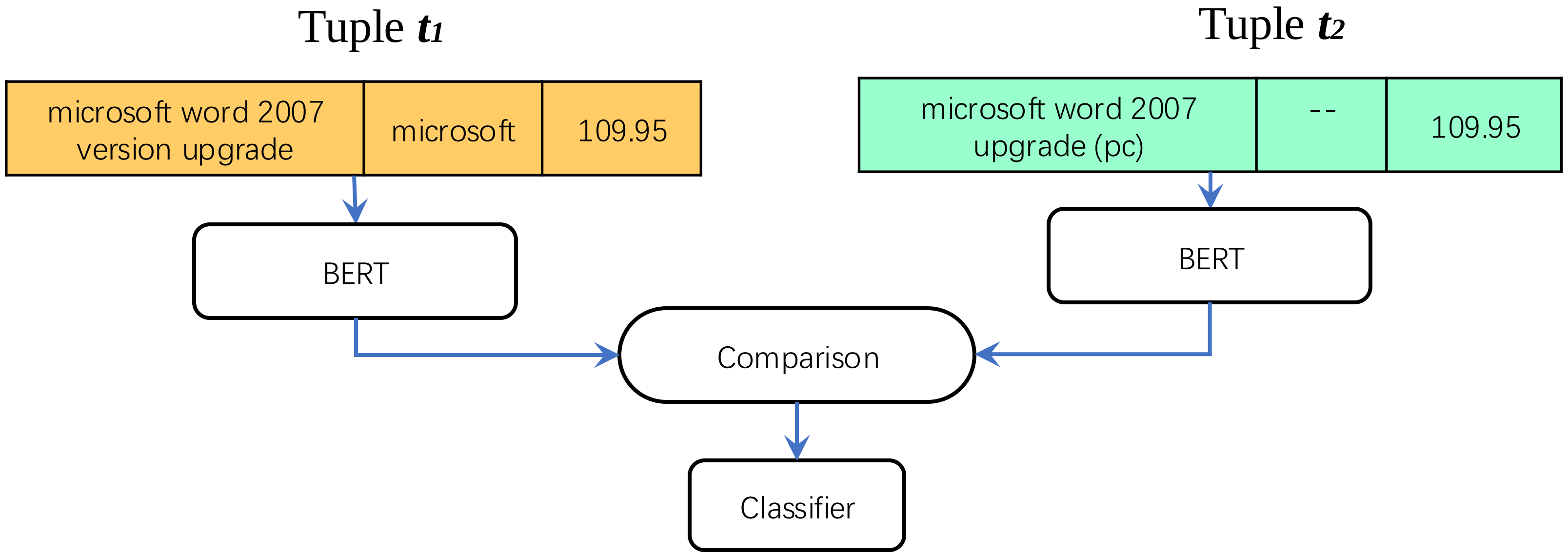
Entity Matching – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

- **Interaction:** Asynchronous deep interaction
  - DITTO embeds **pair** *not* **tuple**



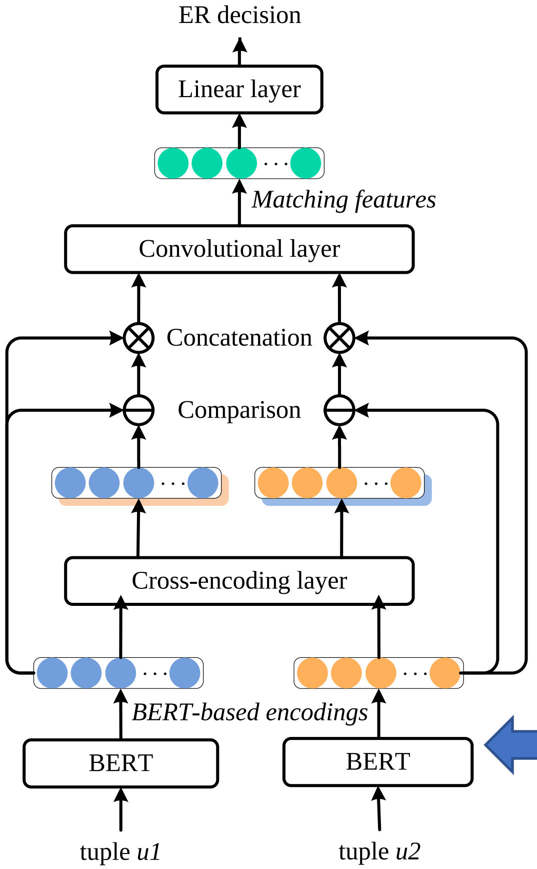
Entity Matching – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

- **Interaction:** Asynchronous deep interaction
  - DITTO embeds **pair** *not* **tuple** – end-to-end blocking unable
  - BERT-ER make it **Siamese** – ready for blocking



Entity Matching – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

- **Interaction:** Asynchronous deep interaction



With individual encodings, we can integrate blocking module

Entity Matching – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

- **Core component** Delayed and Enhanced Alignment

$$e_i = \mathrm{PFFN}(s_i^I + s_i^C) \approx \underbrace{\mathrm{PFFN}(s_i^I)}_{\text{(a) representation}} + \boxed{\underbrace{\mathrm{PFFN}(s_i^C)}_{\text{(b) interaction}}}$$

- Implicit cross-encoding features -> Explicit comparison features

$$\begin{aligned} E_{u1}^C &= \mathrm{softmax}(Q_{u1}K_{u2}^\top)E_{u2}^I \\ E_{u2}^C &= \mathrm{softmax}(Q_{u2}K_{u1}^\top)E_{u1}^I \end{aligned} \quad \Rightarrow \quad \begin{aligned} \mathrm{f}_{\mathrm{sub}}(E^I, E^C) &= (E^I - E^C) \odot (E^I - E^C) \\ \mathrm{f}_{\mathrm{mul}}(E^I, E^C) &= E^I \odot E^C \end{aligned}$$

- Add representation and alignment features -> Concatenate (separating parameters)

$$E_{u1} = E_{u1}^I + E^{u1 \to u2} \quad \Rightarrow \quad E_{u1} = [E_{u1}^I ; E^{u1 \to u2}]$$

- Single-gram features -> Multi-gram features

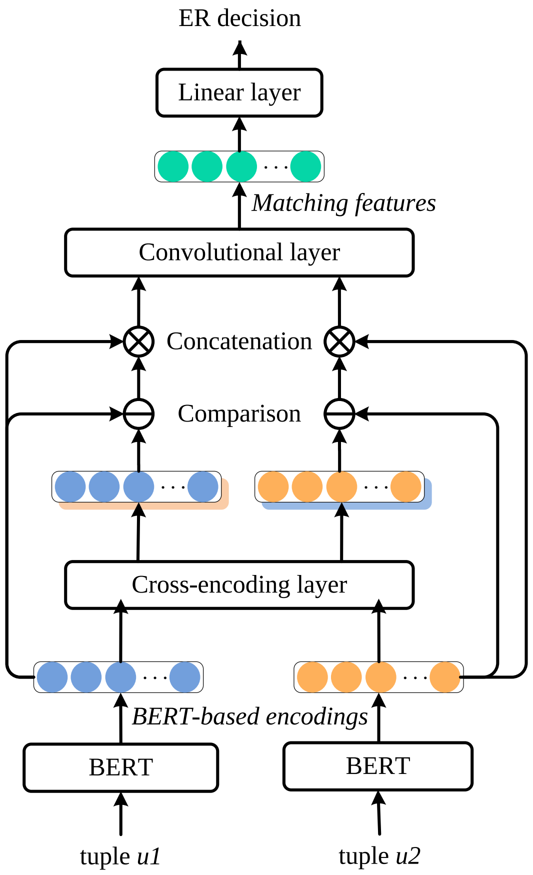
$$M_{u1} = \mathrm{Conv}(E_{u1})$$

Entity Matching – BERT-ER [B Li, Y Wang, W Wang, et al, AAAI'21]

- **Interaction:** Asynchronous deep interaction
- **Encoder:** BERT
- **Embedding:** Deeply contextualized embedding

Performance

- **F-1: 75.3% on *Amazon-Google (refined)* w. 1,300 positive cases**
- **BERT F-1:73.1 % (~2 pts gap)**
- **With Fast blocking ~300X speed-up**



Entity Matching

			On Which Level Tuples Interact?				
			Tuple	*Attribute*	*Token*	*Cross-Encoding*	*Siamese*
Encoder	*Supervised*	*LSTM*		DeepER [VLDB’18]		×	√
				DeepMatcher [SIGMOD’18]		√	√
		GCN			GraphER [AAAI’20]	√	√
		Pretrained LMs			BERT-ER [AAAI’21]	√	√
					DITTO [VLDB’21]	√	×
	Unsupervised	*VAE*	VAER [ICDE’21]			×	√
	Hand-off		AutoML-EM [ICDE’21]			×	×
	Ensemble			RISK [JMLR’21]		√	√

THANK YOU!