Multi-component Similarity Method for Web Product Duplicate Detection

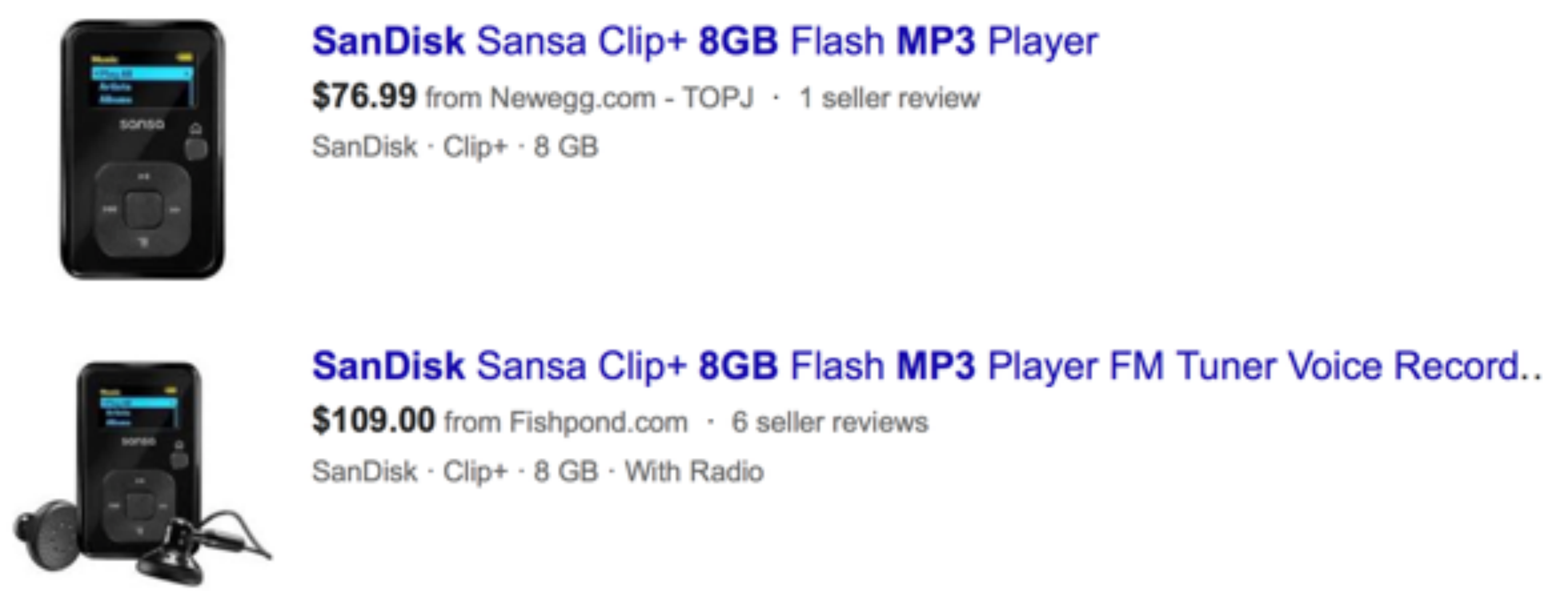
Ronald van Bezu, Sjoerd Borst, Rick Rijkse,
Jim Verhagen, Damir Vandic, Flavius Frasincar

vandic@ese.eur.nl
Erasmus University Rotterdam

ERASMUS UNIVERSITEIT ROTTERDAM

Introduction

- E-commerce is growing at a fast pace
  - Estimated $414 billion in 2018 in US
  - Entity Resolution (ER)



Introduction

- Product descriptions
  - Title
  - Key-Value pairs (basically a Map[String,String])
- Multiple Web shops
- Clean-Clean ER

Baseline approaches

TF-IDF approach for ER

- well-known in database ER literature
- Computes the TF-IDF value for each unique term that occurs in the product attribute values
- Each product description is represented by the TF-IDF vector
- Cosine similarity between pairs of vectors -> distance between product descriptions

Baseline approaches

Title Model Words Method (TMWM)

(1) compute cosine similarity between titles;

(2) if similarity is not high enough, the algorithm extracts *Model Words* from the titles;

(3) if the non-numeric part is *approximately* the same while the numeric part is not, the two products are classified different;

(4) else: compute aggregated weighted similarity

Baseline approaches

- Model words example
  - Samsung - **46”** Class/ LED / **1080p** / **120Hz** / HDTV
  - Samsung - **46”** Class/ LED / **1080p** / **200Hz** / HDTV

Baseline approaches

Hybrid Similarity Method (HSM)

(1) first TMWM is used

(2) for matching Key-Value Pairs (KVP), similarity between values is computed

(3) for non-matching KVP's, model words are extracted and similarity based on % matching is computed

(4) a final weighted similarity is computed

Our Approach
Similarity

- Multi-component Similarity Method
- All-pair similarity computation
    - scalability addressed in other work based on blocking schemes in a distributed environment

```
method KSIM(prod a, prod b)                                   ▷ similarity based on keys
    m := 0                                                    ▷ number of matches
    w := 0                                                    ▷ weight of matches
    I := a.keys and J := b.keys                               ▷ keys without a match
    sim := 0
    for all k_{i,a} ∈ a.keys do
        k_{i,a} := clean(k_{i,a})
        for all k_{j,b} ∈ b.keys do
            k_{j,b} := clean(k_{j,b})
            if keysMatch(k_{i,a}, k_{j,b}) then
                I := I \ k_{i,a} and J := J \ k_{j,b}
                keySim := keySim(k_{i,a}, k_{j,b})
                valueSim := valSim(value(k_{i,a}), value(k_{j,b}))
                sim := sim + keySim * valueSim
                m := m + 1
                w := w + keySim
            end if
        end for
    end for
    keySim* := 0
    if w > 0 then
        keySim* := sim / w
    end if
    return (m, I, J, keySim*)
end method
```

method $\textsc{sim}(\text{prod } a, \text{prod } b)$
    $(m, I, J, keySim^*) := \textsc{ksim}(a, b)$
    $I_{mw} := \text{mw}(I)$ and $J_{mw} := \text{mw}(J)$
    $mwSim := \text{mwSim}(I_{mw}, J_{mw})$
    $titleSim := \text{titleSim}(a.title, b.title, \alpha, \beta)$
    if $titleSim = 0$ **then**
        $\theta_1 := m/\min(|a.keys|, |b.keys|)$
        $\theta_2 := 1 - \theta_1$
    else
        $\theta_1 := (1 - \mu) \cdot \dfrac{m}{\min(|a.keys|, |b.keys|)}$
        $\theta_2 := 1 - \mu - \theta_1$
    end if
    $sim^* := \theta_1 \cdot keySim^* + \theta_2 \cdot mwSim + \mu * titleSim$
    return sim^*
end method

Our Approach
Clustering

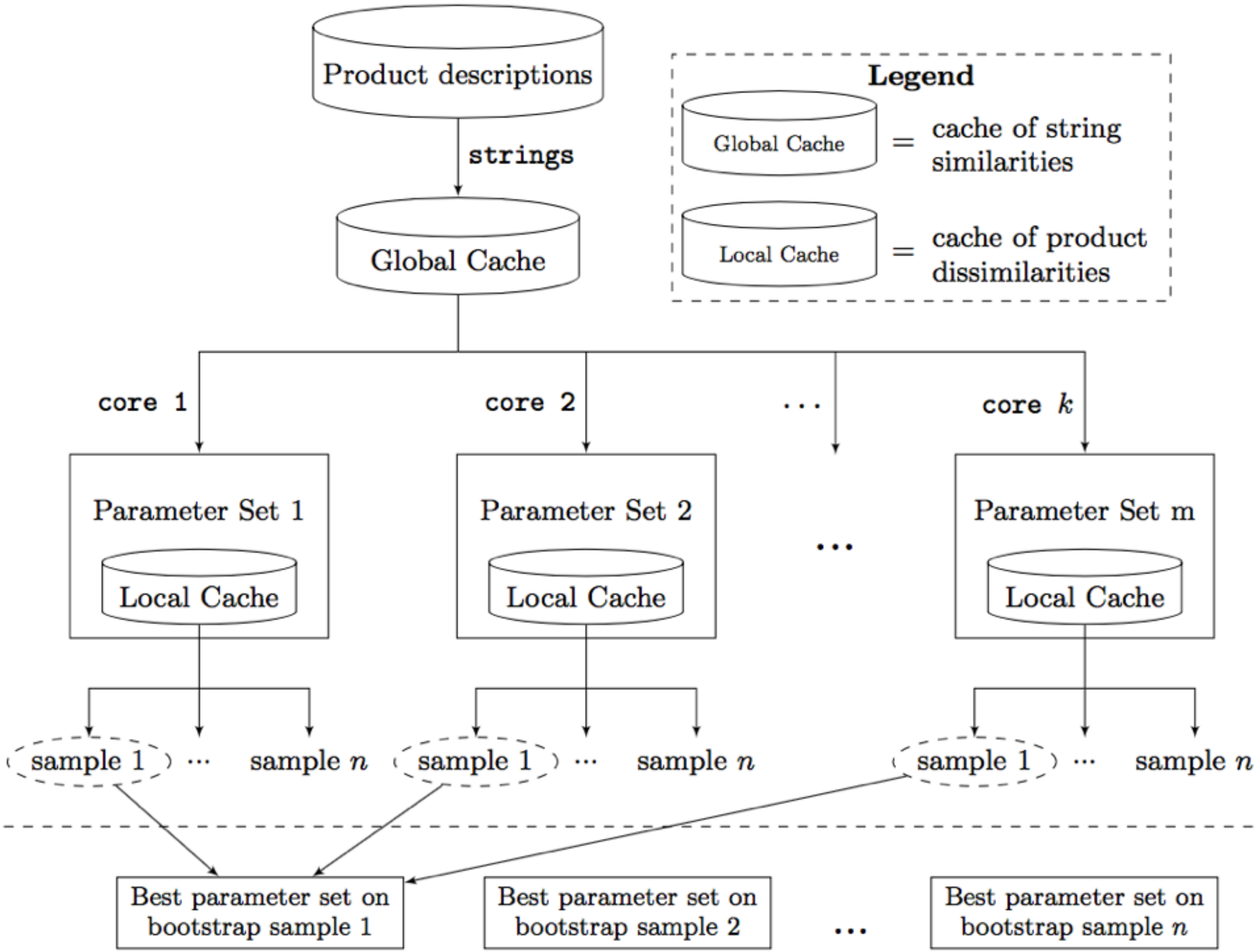
- Hierarchical Clustering
- Handle multiple Web shops at once
- Single linkage with exception to infinite distances to properly handle exclusion cases (e.g., two product descriptions from the same Web shop)

Evaluation

- Compare TF-IDF, TMWM, HSM, and MSM
- Data set contains 1629 TV product descriptions
- 1262 of these are unique
- 4 Web shops: Amazon.com, Newegg.com, Best- buy.com, and TheNerds.net
- On average, 29 key-value pairs per product description

Evaluation

- Interpretation of TP, FP, TN, FN (counting all pairs in produced clusters)
- Baseline algorithms are reimplemented to be used with the hierarchical clustering approach
- Bootstrap samples used as train/test sets
  - trained on F1
- Wilcoxon signed rank test


Product descriptions
strings
Global Cache
Legend
Global Cache
= cache of string similarities
Local Cache
= cache of product dissimilarities
core 1
core 2
...
core k
Parameter Set 1
Local Cache
Parameter Set 2
Local Cache
...
Parameter Set m
Local Cache
sample 1
...
sample n
sample 1
...
sample n
sample 1
...
sample n
Best parameter set on bootstrap sample 1
Best parameter set on bootstrap sample 2
...
Best parameter set on bootstrap sample n

Evaluation

- Results

Method	F_1-measure	precision	recall
TF-IDF	0.335	0.337	0.334
TMWM	0.298	0.349	0.309
HSM	0.287	0.237	0.381
MSM	0.475	0.445	0.512

p-value	TF-IDF	TMWM	HSM	MSM
TF-IDF	x	1.000	1.000	0.000
TMWM	0.000	x	0.999	0.000
HSM	0.000	0.001	x	0.000
MSM	1.000	1.000	1.000	x

Questions?