A Heterogeneous Field Matching Method for Record Linkage

Steven Minton and Claude Nanjo
Fetch Technologies
{sminton, cnanjo}@fetch.com

Craig A. Knoblock, Martin Michalowski, and Matthew Michelson
USC / ISI
{knoblock,martinm,michelso}@isi.edu

Introduction

- Record linkage is the process of recognizing when two database records are referring to the same entity.
  - Employs similarity metrics that compare pairs of field values.
  - Given field-level similarity, an overall record-level judgment is made.

Record Linkage

An example

Union Switch and Signal	2022 Hampton Ave	Manufacturing
JPM	115 Main St	Manufacturing
McDonald’s	Corner of 5th and Main	Food Retail

Joint Pipe Manufacturers	115 Main Street	Plumbing Manufacturer
Union Sign	300 Hampton Ave	Signage
McDonald’s Restaurant	532 West Main St.	Restaurant

Traditional Approaches to Field Matching

Rule Based Approach:

- Pros:
  - Highly tailored domain-specific rules for each fields
    - E.g., last_name > first_name
  - Leverages domain-specific information.
- Cons:
  - Not Scalable
  - Rarely reusable on other domains

Fetch TECHNOLOGIES

Traditional Approaches to Field Matching

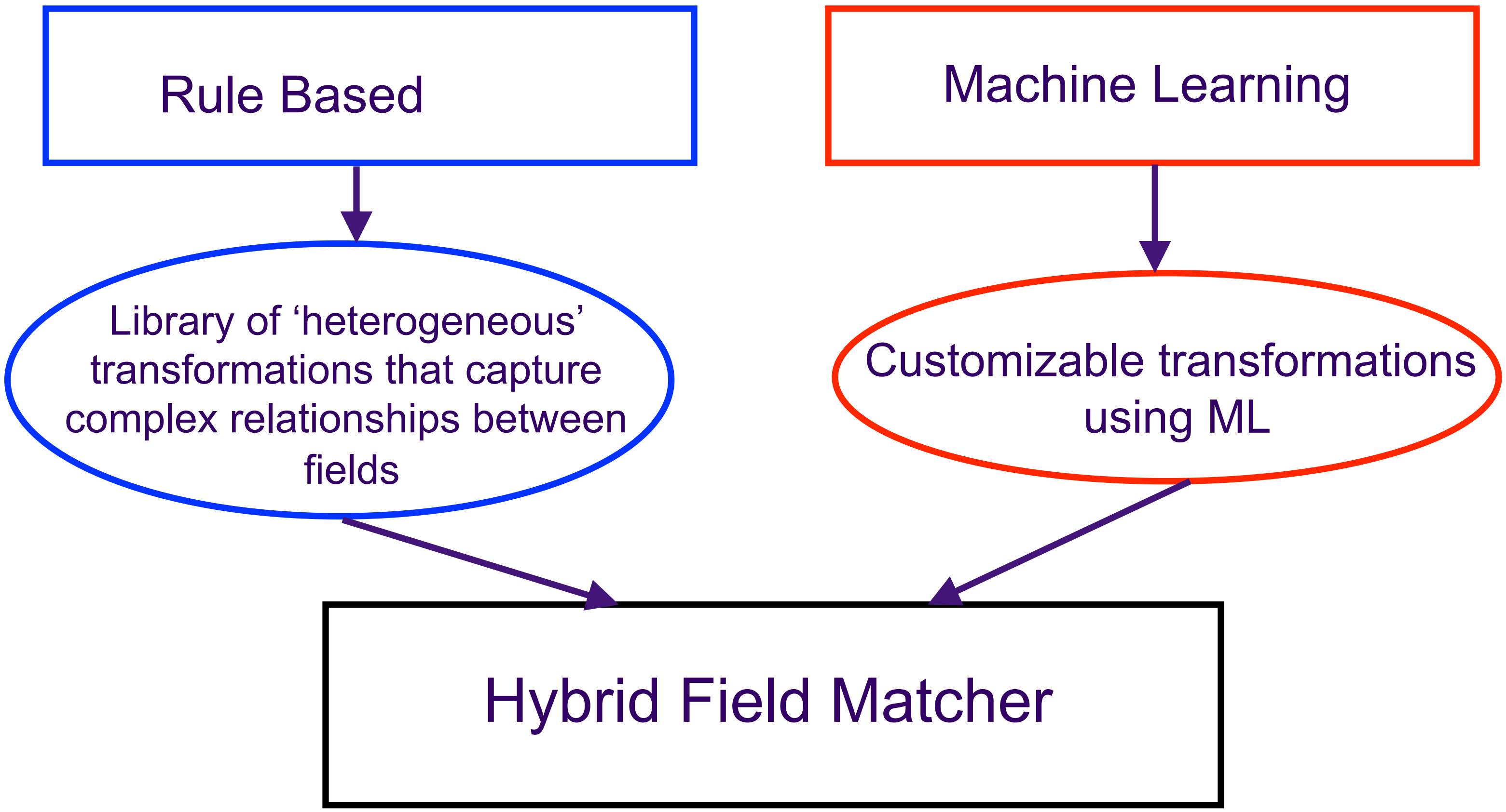
Previous Machine Learning Approaches:

- Pros
  - Sophisticated decision-making methods at record level (e.g. DT, SVM, etc…)
  - Field matching often generic (TFIDF, Levenshtein)
  - Hence, more scalable
- Cons
  - Often used only one such homogeneous field matching approach
    - Thus, unable to detect heterogeneous relationships within fields (e.g. acronyms and abbreviations)
  - Failed to capture some important domain-specific fine-grained phenomena

Fetch TECHNOLOGIES

Introducing the Hybrid Field Matcher (HFM)

(Based on Sheila Tejada's Active Atlas platform)



Better field matching results in better record linkage

Field Matching: Our Goals

- To identify important relationships between tokens
- To capture these relationships using an expressive library of ‘transformations’.
- To make these transformations generalizable across domain types.
- To translate the knowledge imparted from their application into a field score.

Field Matching

“JPM” ~ “Joint Pipe Manufacturers” → Acronym

“Hatchback” ~ “Liftback” → Synonym

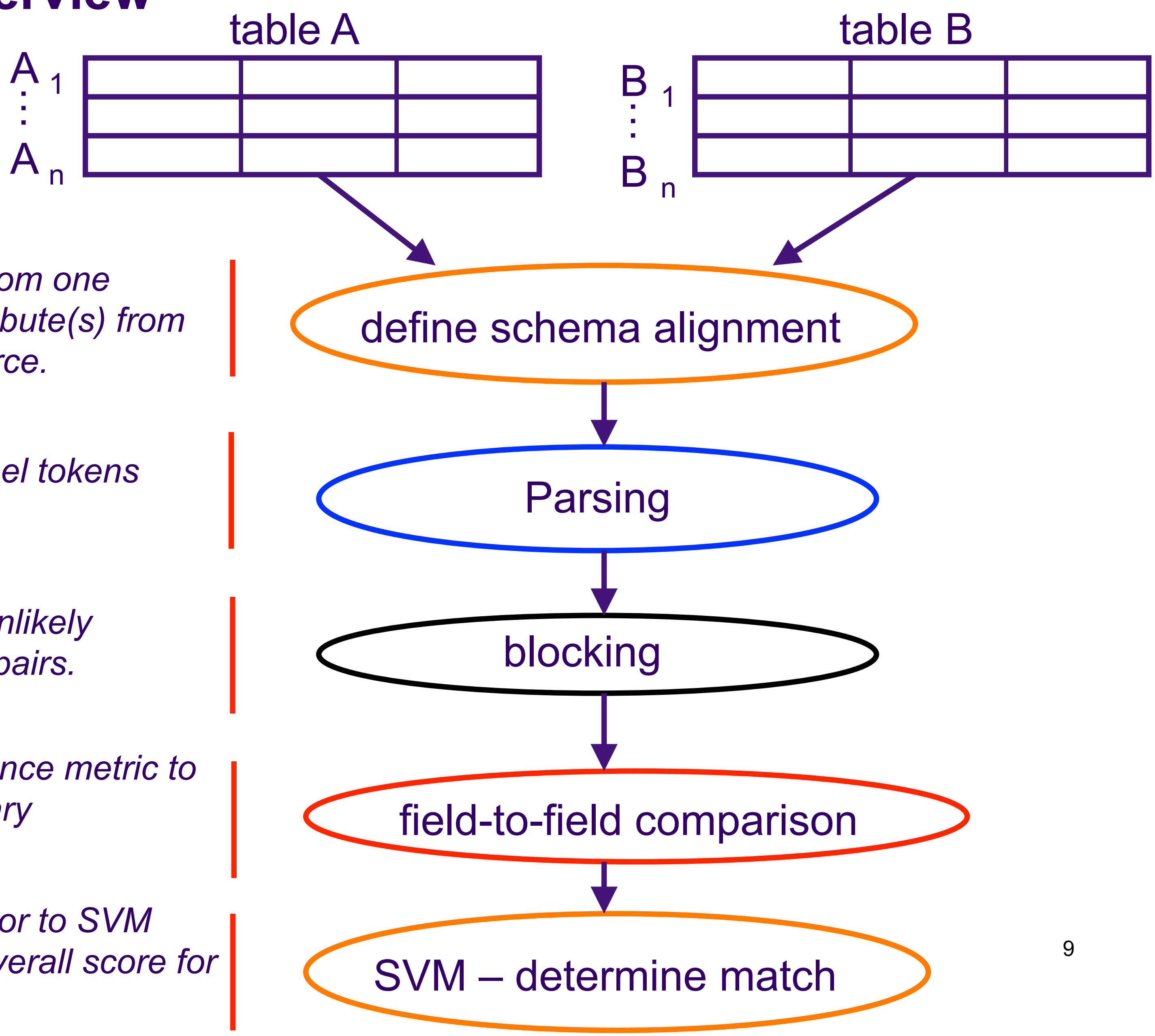
“Miinton” ~ “Minton” → Spelling mistake

“S. Minton” ~ “Steven Minton” → Initials

“Blvd” ~ “Boulevard” → Abbreviation

“200ZX” ~ “200 ZX” → Concatenation

HFM Overview



Map attribute(s) from one datasource to attribute(s) from the other datasource.

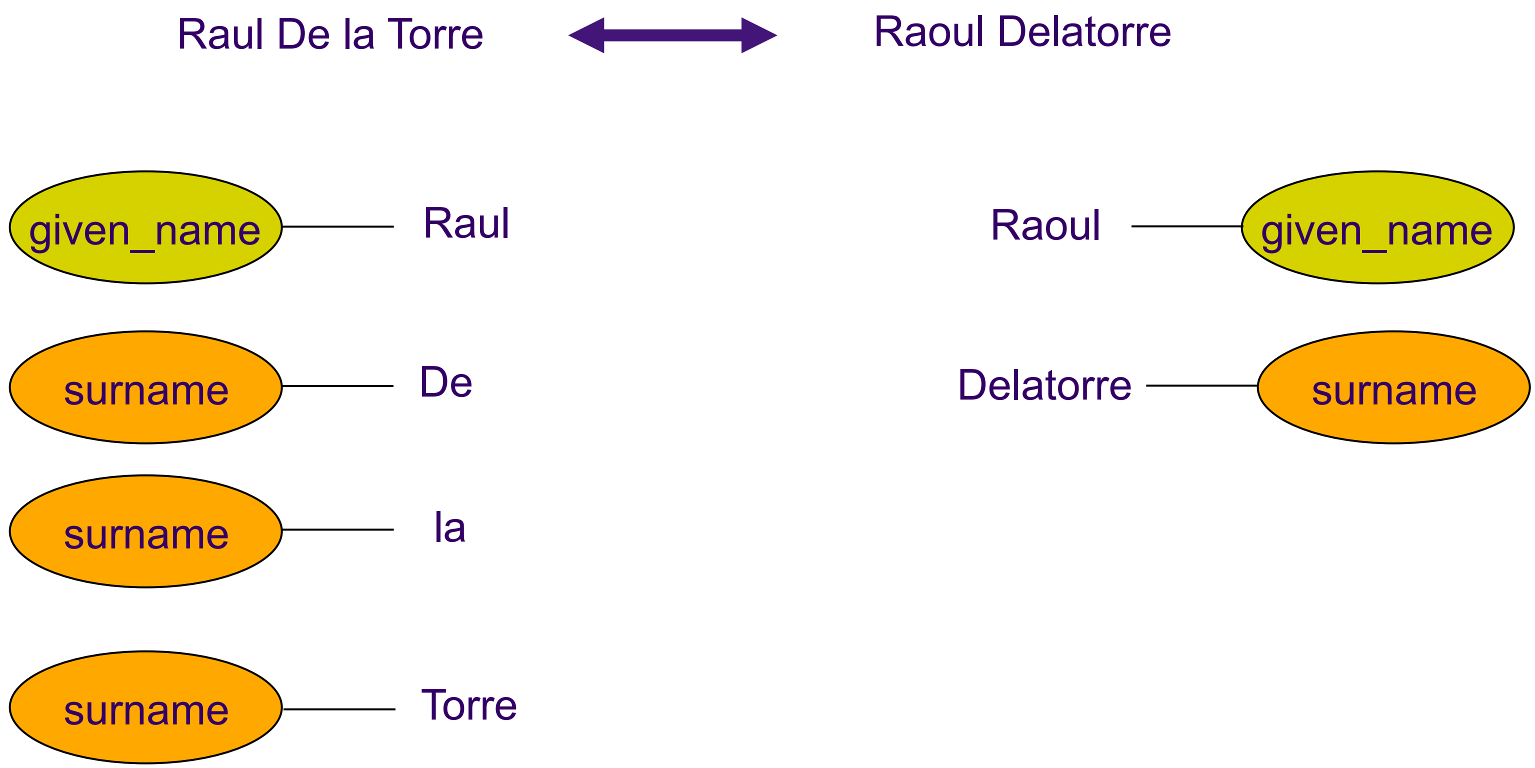
Tokenize, then label tokens

Eliminate highly unlikely candidate record pairs.

Use learned distance metric to score field– primary contribution

Pass feature vector to SVM classifier to get overall score for candidate pair.

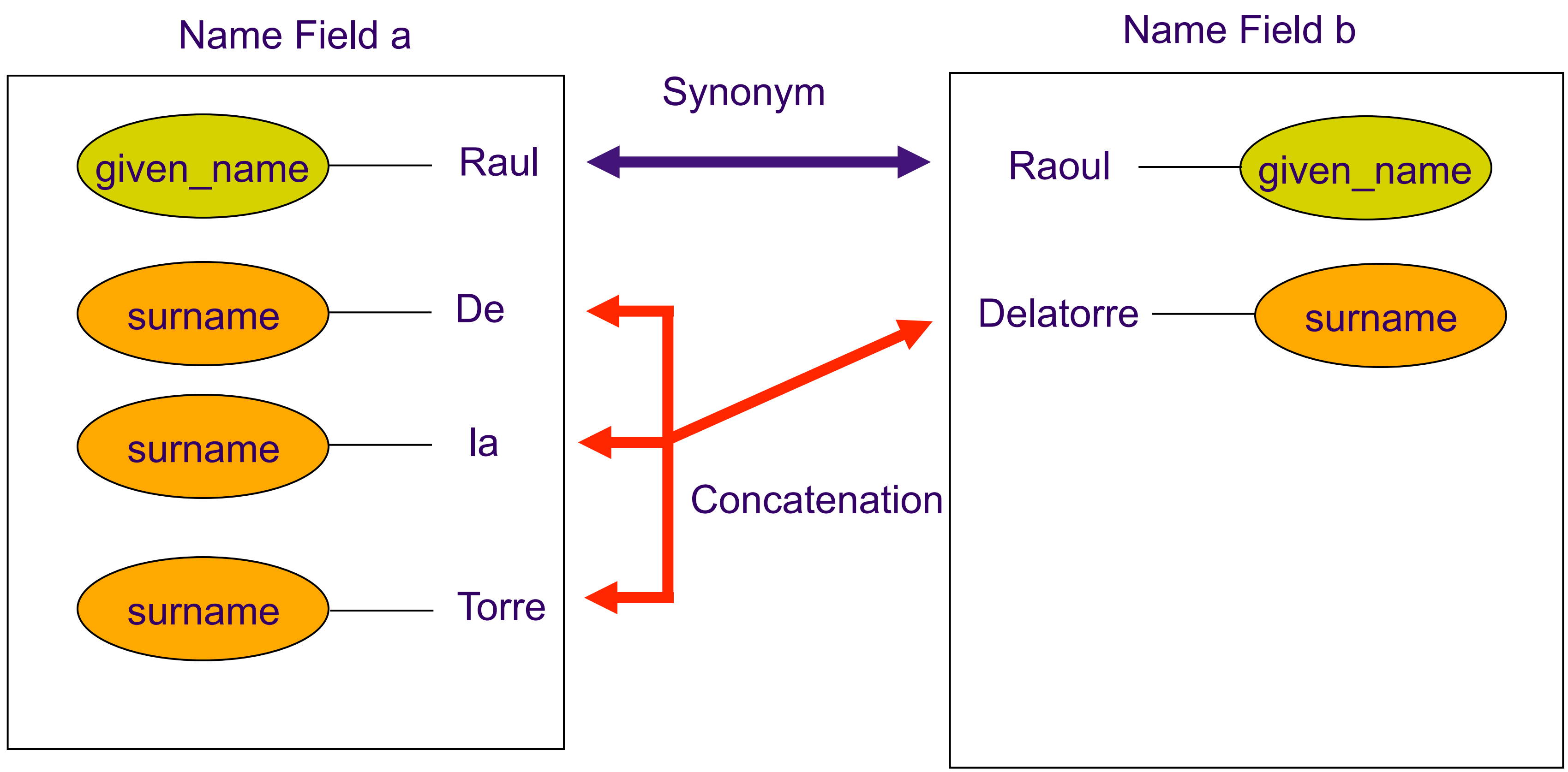
HFM Overview
Parsing and tagging



HFM Overview
Blocking

- Provide the best set of candidate record pairs to consider for record linkage
- Blocking step should not affect recall by eliminating good matches
- We used a reverse index
  - datasource 1 used to build index
  - datasource 2 used to do lookup

HFM Overview
Field to Field Comparison



Score = 0.98

HFM Overview
SVM Classification

	Record 1	Record 2	Score
Name	*Raoul DelaTorre*	*Raul De la Torre*	***0.98***
Gender	*Male*	*M*	***0.99***
Age	*35*	*36*	***0.79***

SVM Classifier

Score for candidate pair: 0.975

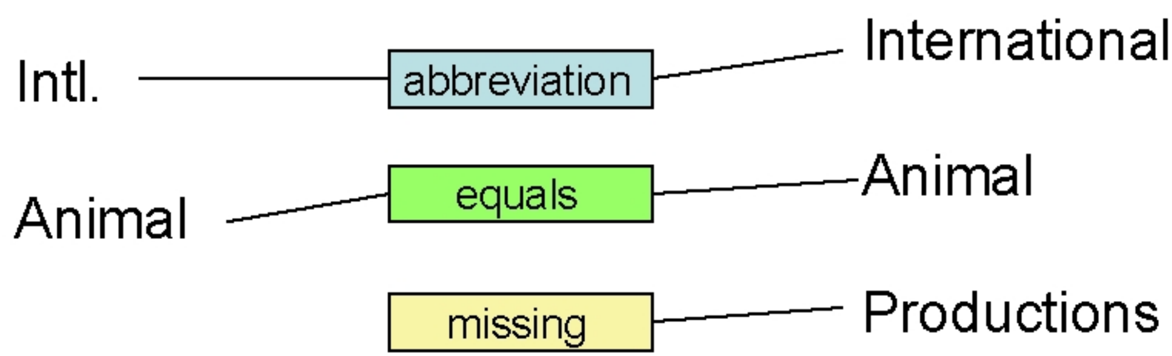
Training the Field Learner

Transformations =

{ Equal, Synonym, Misspelling, Abbreviation,   Prefix, Acronym, Concatenation, Suffix, Soundex, Missing… }

Transformation Graph

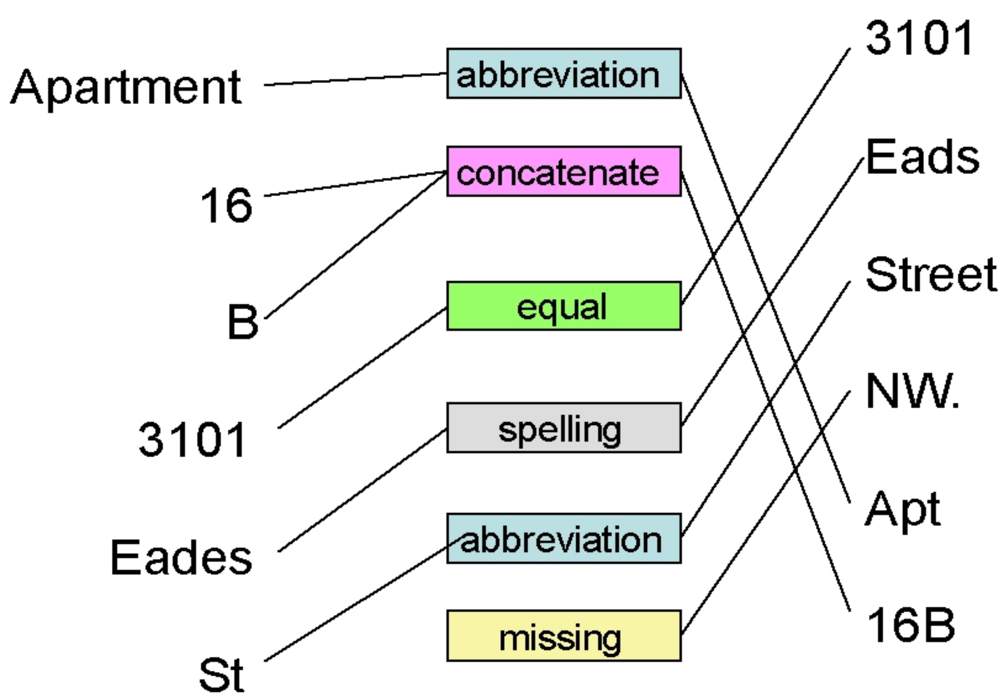
“Intl. Animal” ⟷ “International Animal Productions”



Training the Field Learner

Another Transformation Graph

"Apartment 16 B, 3101 Eades St" ⟷ "3101 Eads Street NW Apt 16B"



Training the Field Learner

Step 1: Tallying transformation frequencies

<u>Generic Preference Ordering</u>

Equal > Synonym > Misspelling > Missing …

Training Algorithm:

I. For each training record pair
    i. For each aligned field pair (a, b)
        i. build transformation graph T(a, b)
            - “complete / consistent”
            - Greedy approach: preference ordering over transformations

Training the Field Learner

Step 2: Calculating the probabilities

- For each transformation type v_i (e.g. Synonym), calculate the following two probabilities:

$p(v_i|\text{Match}) = p(v_i|M) = (\text{freq. of } v_i \text{ in } M) / (\text{size } M)$

$p(v_i|\text{Non-Match}) = p(v_i|\neg M) = (\text{freq. of } v_i \text{ in } \neg M) / (\text{size } \neg M)$

- Note: Here we make the Naïve Bayes assumption

Scoring unseen instances

Naïve Bayes assumption

$$p(M \mid v_1, v_2, \ldots, v_n) = \frac{p(M)\prod_{i=1}^{n} p(v_i \mid M)}{\prod_{i=1}^{n} p(v_i)}$$

$$Score_{HFM} = \frac{\boxed{p(M \mid \mathbf{V})}}{p(M \mid \mathbf{V}) + p(\neg M \mid \mathbf{V})}$$

$$= \frac{p(M)\prod_{i=1}^{n} p(v_i \mid M)}{p(M)\prod_{i=1}^{n} p(v_i \mid M) + p(\neg M)\prod_{i=1}^{n} p(v_i \mid \neg M)}$$

Scoring unseen instances

An Example

a = “Giovani Italian Cucina Int’l”

b = “Giovani Italian Kitchen International”

T(a,b) = {*Equal*(Giovani, Giovani), *Equal*(Italian, Italian),
Synonym(Cucina, Kitchen), *Abbreviation*(Int’l, International)}

Training:

p(M) = 0.31	p(¬ M) = 0.69
p(*Equal* | M) = 0.17	p(*Equal* | ¬ M) = 0.027
p(*Synonym* | M) = 0.29	p(*Synonym* | ¬ M) = 0.14
p(*Abbreviation* | M) = 0.11	p(*Abbreviation* | ¬ M) = 0.03

$$p(M)\prod p(v_i \mid M) = 2.86\text{E -4}$$

$$p(\neg M)\prod p(v_i \mid \neg M) = 2.11\text{E -6}$$

$\text{Score}_{\text{HFM}}$ = 0.993 → Good Match!

Consider the following case

Pizza Hut Restaurant ⟷ Pizza Hut Rstrnt

Sabon Gari Restaurant ⟷ Sabon Gari Rstrnt

Should these score equally well?

Introducing Fine-Grained Transformations

- Capture additional information about a relationship between tokens
  - Frequency information
    - Pizza Hut vs. Sabon Gari
  - Semantic category
    - Street Number vs. Apartment Number
- Parameterized transformations
  - *Equal[HighFreq] vs Equal[MedFreq]*
  - *Equal[FirstName] vs Equal[LastName]*

Fine-Grained Transformations
Frequency Considerations

Coarse Grained:

Pizza Hut Restaurant

2 Equal and 1
Abbreviation Transformation

Pizza Hut Rstrnt

Sabon Gari Restaurant

2 Equal and 1 Abbreviation
transformations

Sabon Gari Rstrnt

Both score equally well.

Fine-Grained Transformations
Frequency Considerations

Fine Grained:

Pizza Hut Restaurant

2 ***high-frequency*** Equal transformations and 1 Abbreviation transformation

Pizza Hut Rstrnt

Sabon Gari Restaurant

2 ***low-frequency*** Equal transformations and 1 Abbreviation transformation

Sabon Gari Rstrnt

Sabon Gari Restaurant scores higher since low frequency equals are much more indicative of a match

Fine-Grained Transformations
Semantic Categorization

Without Tagging:

123 Venice Boulevard, 405

Equal

Equal

Equal

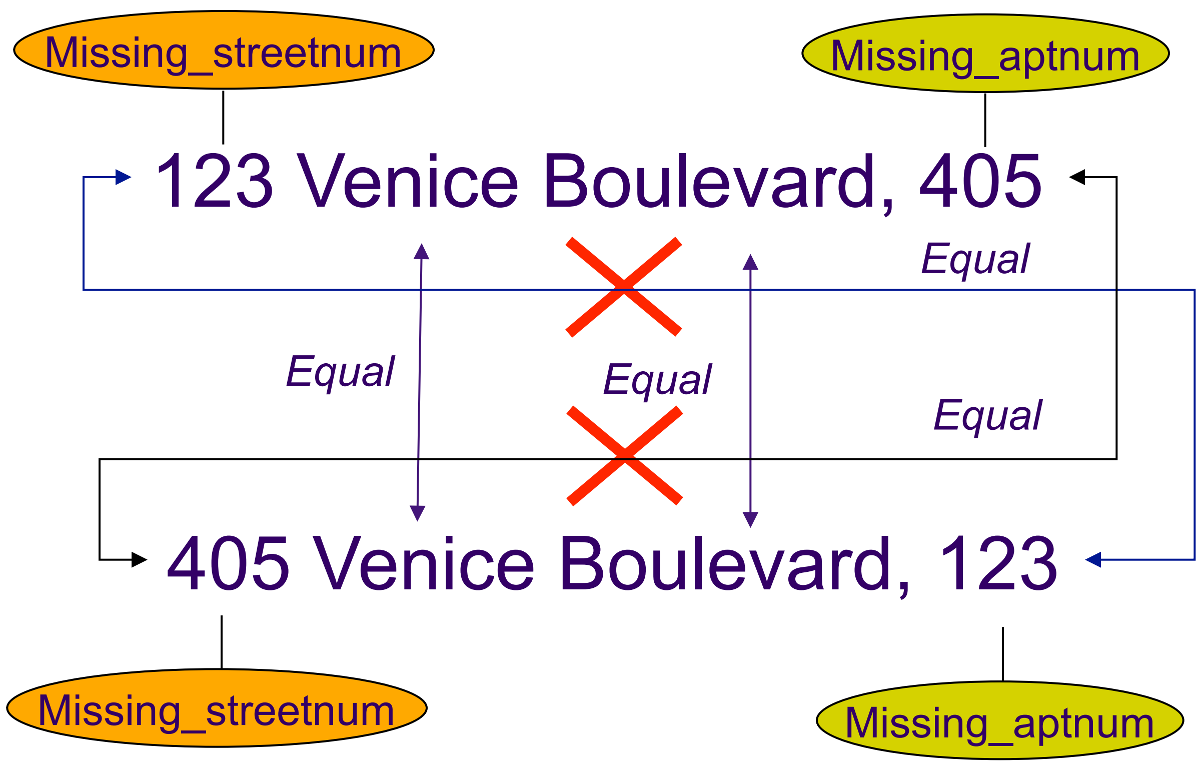
Equal

405 Venice Boulevard, 123

Scores well

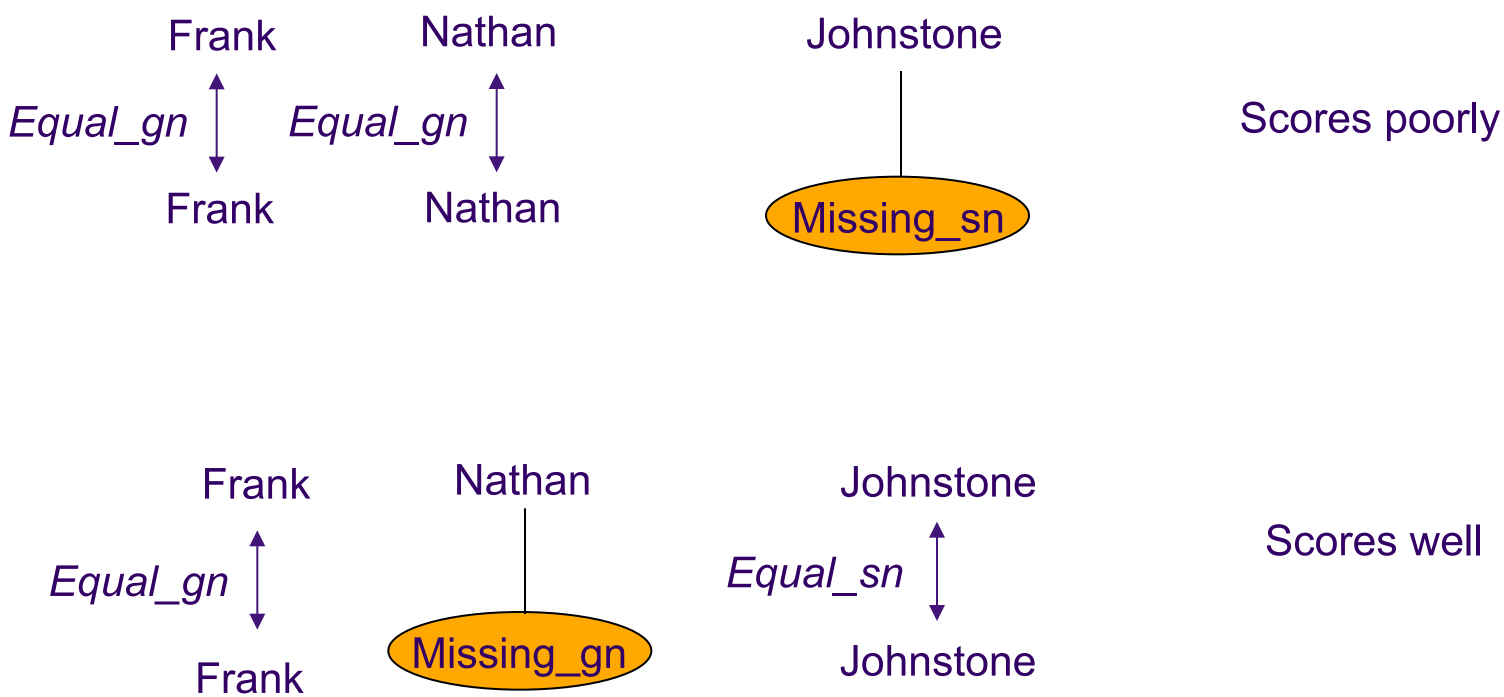
Fine-Grained Transformations
Semantic Categorization

With Tagging:



Scores poorly

Fine-Grained Transformations -
Differential Impact of Missings



A missing surname penalizes a score far more than a missing given name.

Global Transformations

- Applied to entire transformation graph
  - Reordering
    - “Steven N. Minton” vs. “Minton, Steven N.”
  - Subset
    - “Nissan 150 Pulsar wth AC” vs.
      “Nissan 150 Pulsar”

Experimental Results

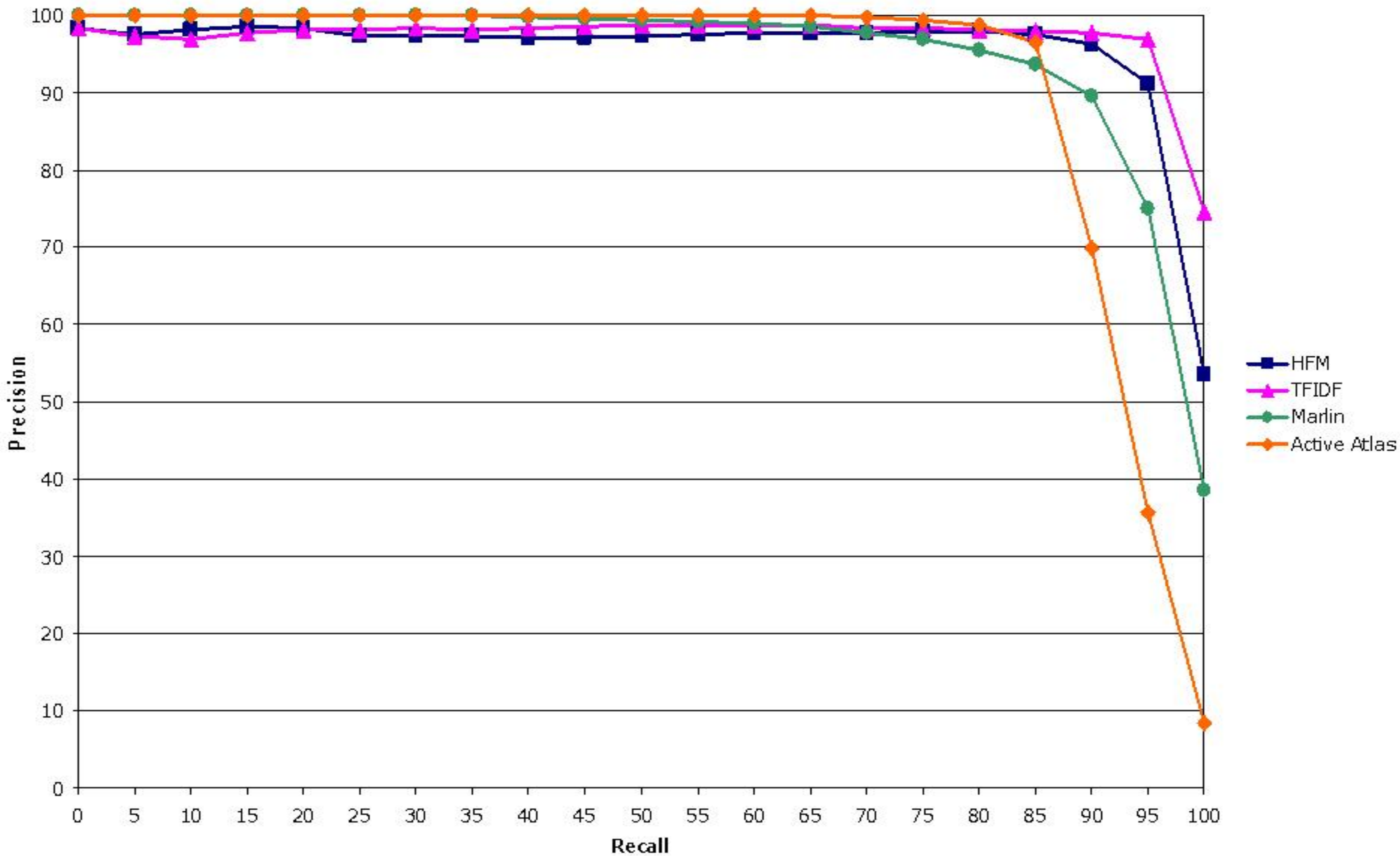
- We compared the following four systems:
  - **HFM**
  - **TF-IDF** (Vector-based cosine)
    - matches tokens
  - **MARLIN**
    - learned string edit distance
  - **Active Atlas** (older version)
- We made use of 4 datasets
  - Two restaurant datasets
  - One car dataset
  - One hotel dataset

Experimental Results

- Reproduced the experimental methodology described in the MARLIN paper (entitled "**Adaptive Duplicate Detection Using Learnable String Similarity Measures**" by M. Bilenko and R. Mooney, 2003)
  - All methods calculate vector of feature scores
    - Pass to SVM trained to label matches/non-matches
    - Radial Bias Function kernel, $\gamma = 10.0$
  - 20 trials, cross-validation
    - Dataset randomly split into two folds for cross validation
    - Precision interpolated at 20 standard recall levels.

“Marlin Restaurants” Dataset

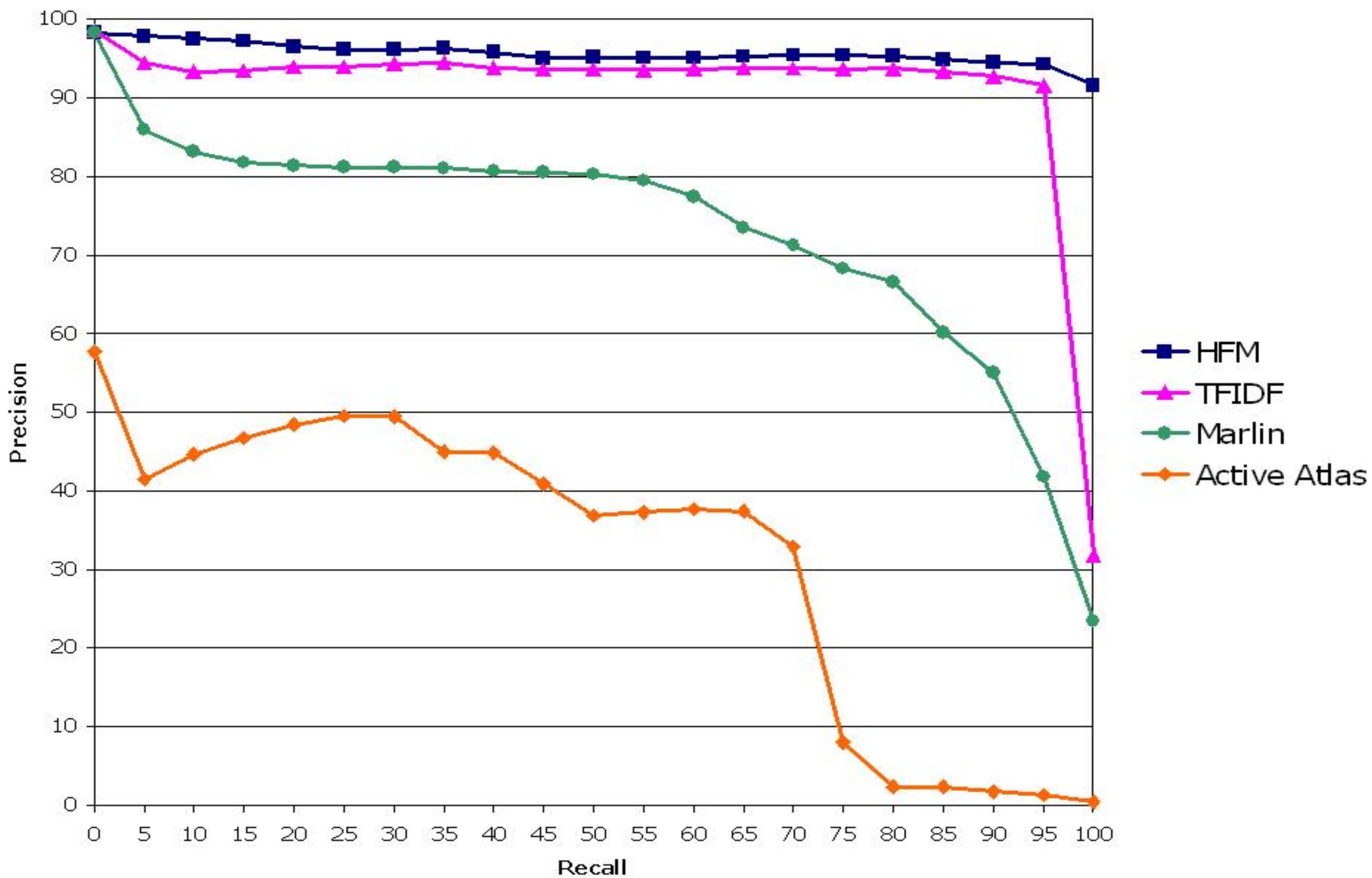
Fields: name, address, city, cuisine
Size: Fodors (534 records), Zagats (330 records),112 Matches



Larger Restaurant Set With Duplicates

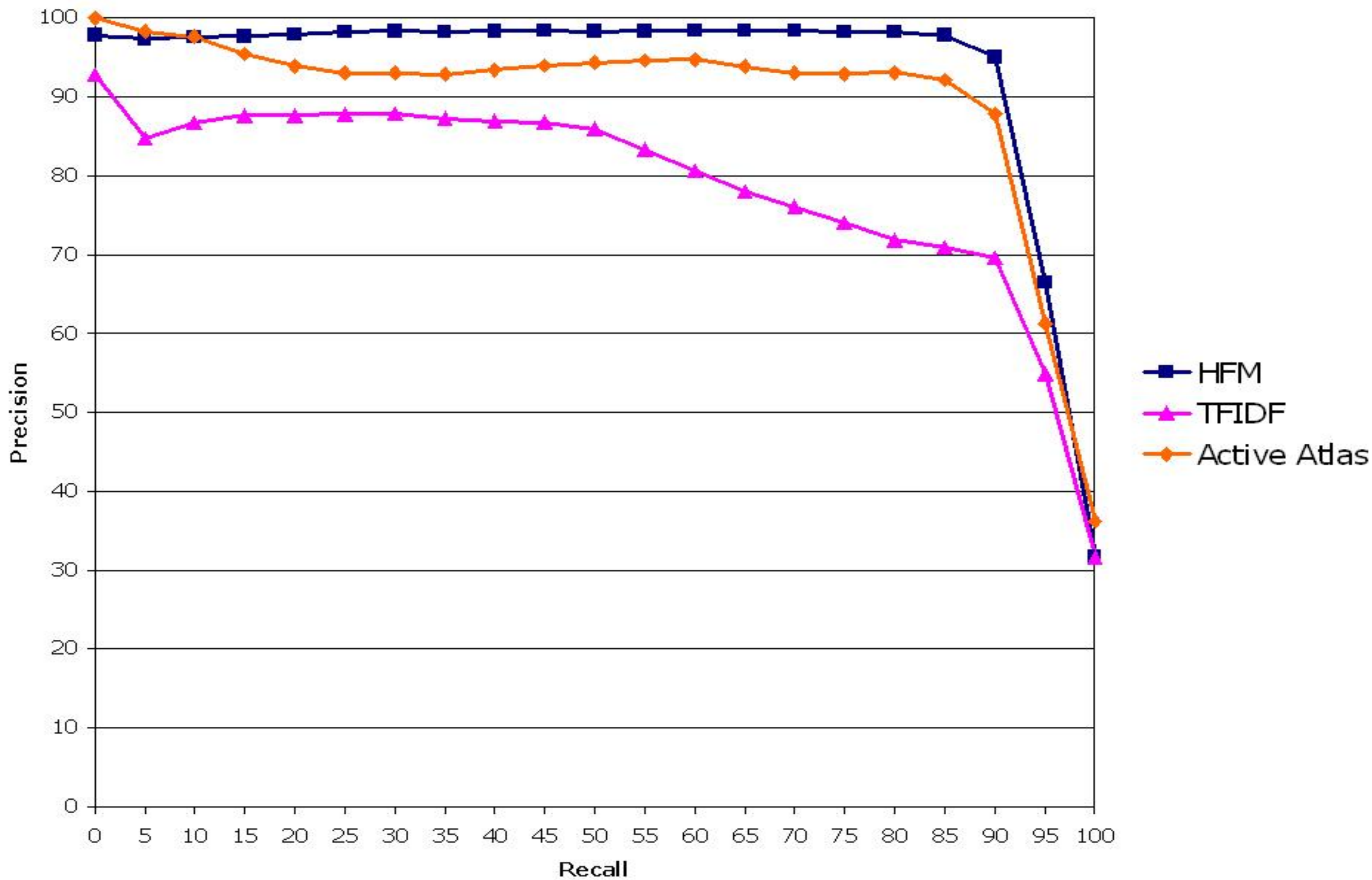
Fields: name, address

Size: LA County Health Dept. Website (3701), Yahoo LA Restaurants (438), 303 Matches



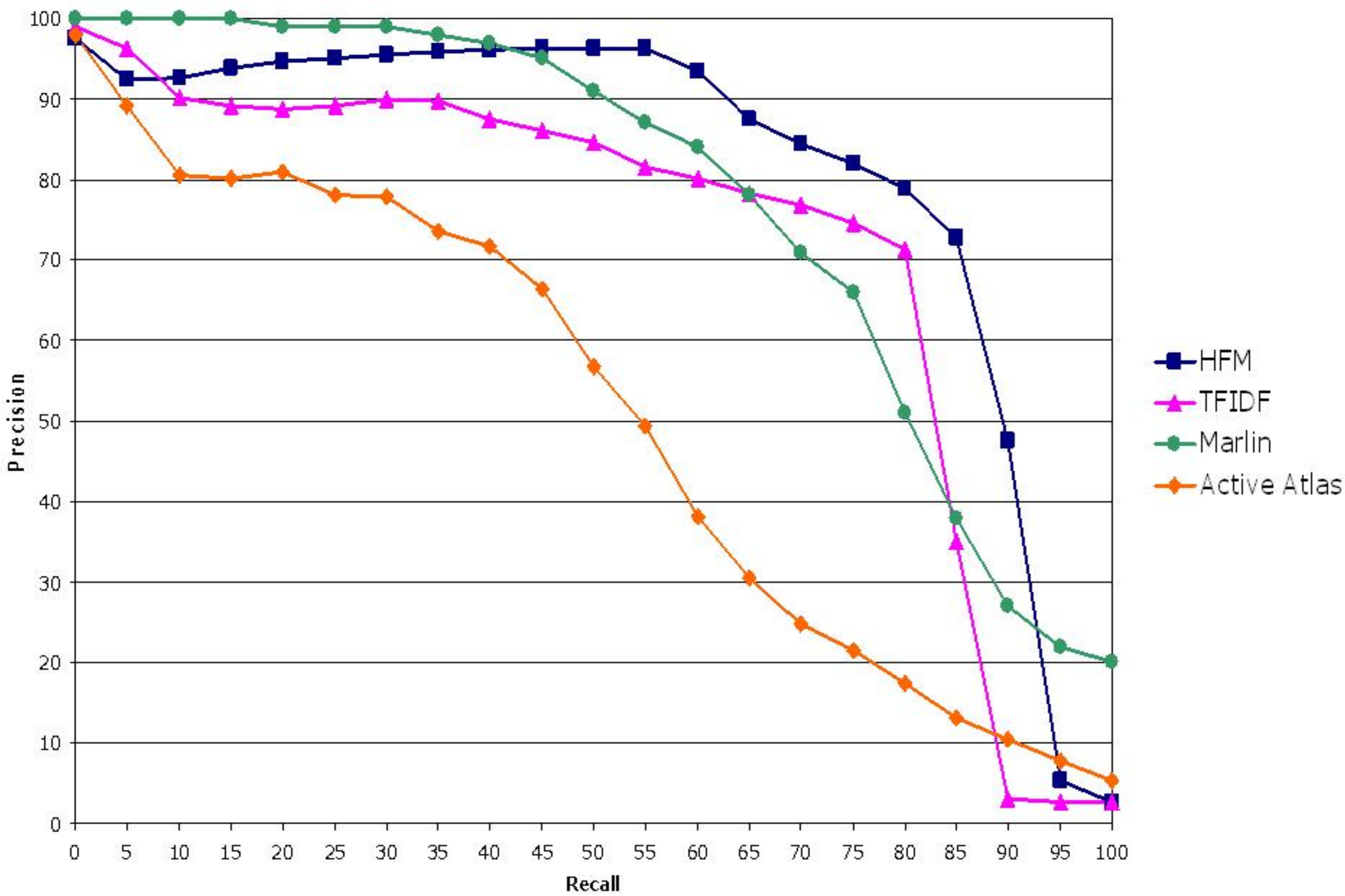
Car Dataset

Fields: make, model, trim, year
Attributes: Edmunds (3171), Kelly Blue Book (2777), 2909 Matches



Bidding for Travel

Fields: star rating, hotel name, hotel area
Size: Extracted posts (1125), “Clean” hotels (132), 1028 matches



Result Summary

Matching Technique	Domain			
	Marlin Res.	MD Res.	Cars	BFT
HFM	94.64	**95.77**	**92.48**	**79.52**
Active Atlas	*92.31*	45.09	88.97	56.95
TF-IDF	**96.86**	93.52	78.52	75.65
Marlin	*91.39*	76.29	N/A	75.54

Average maximum F-measure for detecting matching records. Note: *red* is not significant with respect to a 1-tailed paired t-test at confidence 0.05

Discussion of Results

- Comparison to TFIDF
  - HFM outperforms TFIDF by identifying complex relationships which improve matching
    - Restaurant Datasets:
      - Tokens related mostly by equality
      - Minor improvement over TFIDF
    - Car Dataset:
      - Transformations yield large improvements (in particular, synonym and ordered concatenation transformations)
- Comparison to Active Atlas
  - HFM introduces fine-grained & global transformations
  - HFM based on a better justified statistical approach. (Improved scoring of transformations based on Naïve Bayes)
- Comparison to Marlin
  - Can handle larger datasets
  - Captures important token-level relationships not accessible to Marlin
  - Token-based and not character-based

Discussion / Conclusion

- Alternative to transformations: normalize/preprocess data
  - No normal form
    - Caitlyn → {Catherine, Lynne}
- Scalability
  - HFM does well on large, complex datasets

Fetch TECHNOLOGIES

Acknowledgements

- We would like to thank:
  - Mikhail Bilenko for his kind help in helping us set up and run MARLIN on our datasets.
  - Sheila Tejada for her work on Active Atlas, the precursor to HFM

Questions / Comments

Thank you!

HFM Overview

Schema Alignment

- Field alignments are defined mappings between attribute(s) from one datasource to attribute(s) from another datasource.

First Name	Last Name	Age	Gender
Raoul	DelaTorre	35	Male

Name	SS#	Age	Gender
De la Torre, Raul	N/A	36	M