ACCURACY OF APPROXIMATE STRING JOINS USING GRAMS

Oktie Hassanzadeh

Mohammad Sadoghi

Renée J. Miller

University of Toronto

September 23, 2007
Vienna, Austria

5th International Workshop on Quality in Databases at VLDB

Outline

- Problem Definition
- Related Work
- Overview of Similarity Measures
- Accuracy Evaluation
- Conclusion

Problem Definition

- Input: Two relations of string records $R = \{r_i : 1 \leq i \leq N_1\}$ and $S = \{s_j : 1 \leq j \leq N_2\}$
- Output: pairs $(r_i, s_j) \in R \times S$ where r_i and s_j are *similar* records
- Two records are *similar* if $\mathrm{sim}(r_i, s_j) \geq \theta$ for some string similarity function sim() and a threshold θ

r1	Microsoft Corp.
r2	Macrosoft Corp.
r3	Microsoft Corporation
r4	AT&T Corp.
r5	AT&T Inc.

sim()

1.00
0.85
0.71
0.38
0.15

s1	Microsoft Corp.
s2	Macrosoft Corp.
s3	Microsoft Corporation
s4	AT&T Corp.
s5	AT&T Inc.

if θ=0.7 => (r1,s1), (r1,s2), (r1,s3) will be in the output

UofT:DB Group

Related Work

- A huge amount of work on Similarity Join / Record Linkage
  - [Tutorial-VLDB'05, Tutorial-SIGMOD'06]
- Many string similarity measures proposed
  - Survey for duplicate detection in [DDSurvey-TKDE'07]
  - A comparison for name-matching in [NameMatching-IJCAI'03] by Cohen et al.
  - Benchmarked for declarative approximate selection in [D.App.σ-SIGMOD'07]

Related Work - Efficiency

- Most of recent work address efficiency
- Many efficient algorithms are based on q-grams
  - treat each string as a set of q-grams (substrings of length q)
    - "string" => {'str', 'tri' , 'rin' , 'ing'}
- Using indexing techniques and algorithms for set-similarity joins

Related Work - Efficiency

- Techniques for set-similarity join (Signature-based techniques)
  - Locality Sensitive Hashing (LSH) [LSH-STOC'97, FMS-SIGMOD'03]
    - Derived from dimensionality reduction techniques for nearest neighbor problem in high-dimensional spaces
  - PartEnum and WtEnum [ExactSSJoin-VLDB'06]
  - Multi-Probe LSH [MP-LSH-VLDB'07]
- Indexing Techniques
  - Some derived from the indexing techniques in IR
    - Novel indexing and optimization strategies, without extensive parameter tuning [AllPairs-WWW'07]
  - Variable-length grams [VGRAM-VLDB'07] by Chen Li, et al.

Choice of the similarity measure in these techniques is limited

Their effectiveness depends on the value of the threshold

Related Work - Accuracy

- Very few works address accuracy
  - [FMS-SIGMOD'03] introduces fuzzy match similarity as a more accurate measure
    - Not compared with other measures
  - [NameMatching-IJCAI'03] provides an accuracy comparison of several measure for name matching
    - Efficiency not considered
  - [D.App.σ-SIGMOD'07] benchmarks accuracy of several measures for declarative approximate selection
    - Problem: Given a query, find similar records to that query
    - Extension to join and the effect of threshold values not considered

Overview of Similarity Measures

- Overlap
  - Jaccard and Weighted Jaccard
- Edit distance
- From IR
  - Cosine w/tf-idf
  - BM25
  - Language Modeling
  - Hidden Markov Models
- Hybrid

Why?

- High Scalability:
  Various techniques exist for enhancing the performance of these measures.

- High Accuracy:
  Previous work (on name-matching and approximate selection) has shown their high accuracy

Overlap

- Jaccard

r_1 = "Microsoft" $\mathbf{r_1}$ = {\$M, **Mi**, **ic**, cr, ro, os, so, of, ft, t\$}
r_2 = "Macrosoft" $\mathbf{r_2}$ = {\$M, **Ma**, **ac**, cr, ro, os, so, of, ft, t\$}

$$sim_{Jaccard}(r_1, r_2) = \frac{|\mathbf{r}_1 \cap \mathbf{r}_2|}{|\mathbf{r}_1 \cup \mathbf{r}_2|} = 8/12 = 0.67$$

- Weighted Jaccard
  - So that "AT&T Corp." is more similar to "AT&T Inc." *than* "IBM Corp."

$$sim_{WJaccard}(r_1, r_2) = \frac{\sum_{t \in \mathbf{r}_1 \cap \mathbf{r}_2} w_R(t)}{\sum_{t \in \mathbf{r}_1 \cup \mathbf{r}_2} w_R(t)}$$

$$w_R(t) = \log\left(\frac{N - n_t + 0.5}{n_t + 0.5}\right)$$

N : Total number of records in the relation
n_t : frequency of token t in the relation R

- Weights: Robertson-Sparck Jones (RSJ)
  - Similar to but more effective than the commonly used IDF (Inverse Document Frequency)

UofT:DB Group

Edit Similarity

- $tc(r_1,r_2)$: minimum cost of edit operations to transform r_1 to r_2
- Edit operations: character insert, delete and replace
- Levenshtein distance: unit cost for all operations

$$sim_{edit}(r_1, r_2) = 1 - \frac{tc(r_1, r_2)}{\max\{|\mathbf{r}_1|, |\mathbf{r}_2|\}}$$

sim_{edit} (“`Microsoft`”, “`Macrosft`”) = 1- (2/9) = 0.78

Efficient implementations use grams

UofT:DB Group

From IR

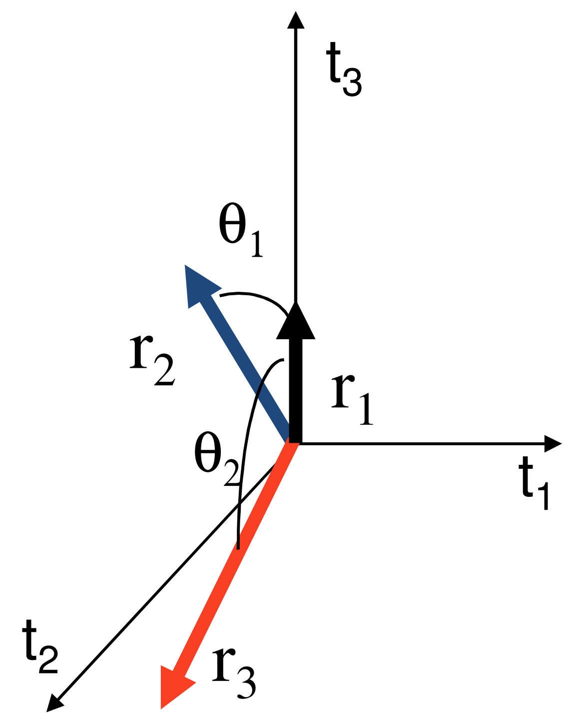
- In IR
  - Given: a query and a collection of documents
  - Return: the most *relevant* documents to the query.
  - Query and Documents: set of words tokens
- Here
  - Given: a query string and a collection of strings
  - Return: the most *similar* strings to the query
  - Query and Records: set of q-grams
- Same techniques can be used

Cosine w/tf-idf

- Well-established measure in the IR
- Strings: vectors of tf-idf weights of q-grams
- Similarity: cosine of the angle between the vectors

$$sim_{Cosine}(r_1, r_2) = \sum_{t \in \mathbf{r_1} \cap \mathbf{r_2}} w_{r_1}(t) \cdot w_{r_2}(t)$$

$$w_r(t) = \frac{w'_r(t)}{\sqrt{\sum_{t' \in \mathbf{r}} w'_r(t')^2}} \quad , \quad w'_r(t) = tf_r(t) \cdot idf(t)$$



BM25

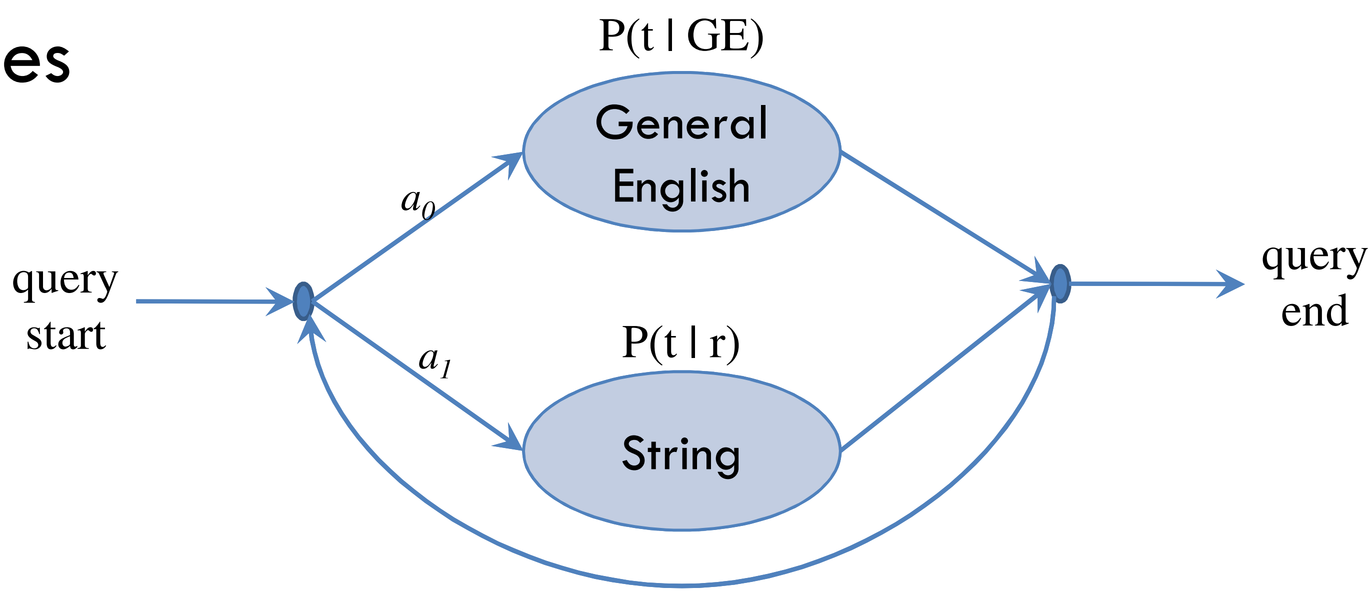
- Outperforms cosine w/tf-idf
- Score formula similar to cosine similarity
- More accurate model
  - Theoretical justification in [UnderstandingIDF-Jdoc'04] by Robertson

Language Modeling

- In IR, Based on [LM-SIGIR'98] by Ponte and Croft
  - Given a collection of documents, a language model is inferred for each
  - The probability of generating a given query according to each of these models is estimated and documents are ranked according to these probabilities
- For string matching
  - a model is inferred for each string in the relation
  - The probability of generating a string according to another string's model is considered the similarity score of these strings

Hidden Markov Models

- Based on a very simple Markov model with two states



$$sim_{HMM}(r_1, r_2) = \prod_{t \in \mathbf{r}_1} (a_0 P(t|GE) + a_1 P(t|r_2))$$

$$P(t|r_2) = \frac{\text{number of times } t \text{ appears in } r_2}{|\mathbf{r}_2|} \qquad P(t|GE) = \frac{\sum_{r \in R} \text{number of times } t \text{ appears in } r}{\sum_{r \in R} |\mathbf{r}|}$$

Hybrid

- GES
  - Edit similarity of word tokens
  - Edit operations: token insertion, token deletion and token replacement
  - Cost of each operation depends on the weight of the token
  - Cost of replacing token t_1 with token t_2 is

$$(1 - sim_{edit}(t_1, t_2))\cdot w(t_1)$$

Hybrid

- SoftTFIDF
  - Cosine w/tf-idf formula: Summing multiplication of normalized tf-idf weights of common tokens
  - SoftTFIDF: Summing multiplication of normalized tf-idf weights of "*close*" tokens
  - *Close*ness based on another similarity function suitable for comparing shorter strings
    - Jaro-Winkler measure for word tokens

Evaluation

- Accuracy measures from IR
  - Precision (Pr)
    - The percentage of similar records among the records that have a similarity score above the threshold θ
  - Recall (Re)
    - the ratio of the number of similar records that have similarity score above the threshold θ to the total number of similar records
  - F1-measure (F_1)
    - harmonic mean of precision and recall, i.e.: $F_1 = \frac{2 \times Pr \times Re}{Pr + Re}$

UofT:DB Group

Datasets

- Enhanced UIS Data Generator [MergePurge-DMKD'98]
  - Gets a clean dataset as input
  - Creates clusters of erroneous records from each clean record by injecting edit errors (character insertion, deletion, replacement or swap), token swap or abbreviation errors
- Clean data sources
  - DBLP titles
  - Company Names
  - People names/Addresses

Data Generator

- Provides the following parameters:
  - The *size* of the dataset to be generated
  - The *fraction of clean tuples* to be utilized to generate erroneous duplicates
  - The *distribution of duplicates*: uniform, Zipfian or Poisson distribution.
  - The *percentage of erroneous duplicates*
  - The *extent of error in each erroneous tuple*
  - *token swap error*
  - *The extent and type of abbreviation errors (if any)*

Datasets

Classification of the datasets used in the experiments

Group	Name	Percentage of: Erroneous Duplicates in the Dataset	Error in each Duplicate Record	Token Swap	Abbr. Error
Dirty	D1	90	30	20	50
	D2	50	30	20	50
Medium Error	M1	30	30	20	50
	M2	10	30	20	50
	M3	90	10	20	50
	M4	50	10	20	50
Low Error	L1	30	10	20	50
	L2	10	10	20	50
Single Error	Abbr.	50	0	0	50
	TokenSwap	50	0	20	0
	LowEdit	50	10	0	0
	MediumEdit	50	20	0	0
	HighEdit	50	30	0	0

Samples From Datasets

A record from the clean company names source: "Morgan Stanley Group Inc."

90% Erroneous duplicates 30% Errors in duplicates
20% Token swap 50% Abbreviation Error

Stsalney Morgan cncorporsated Group
jMorgank Stanlwey Grouio Inc.
Morgan Stanley Group Inc.
Sanlne Morganj Inocrorpated Group
Sgalet Morgan Icnorporated Group

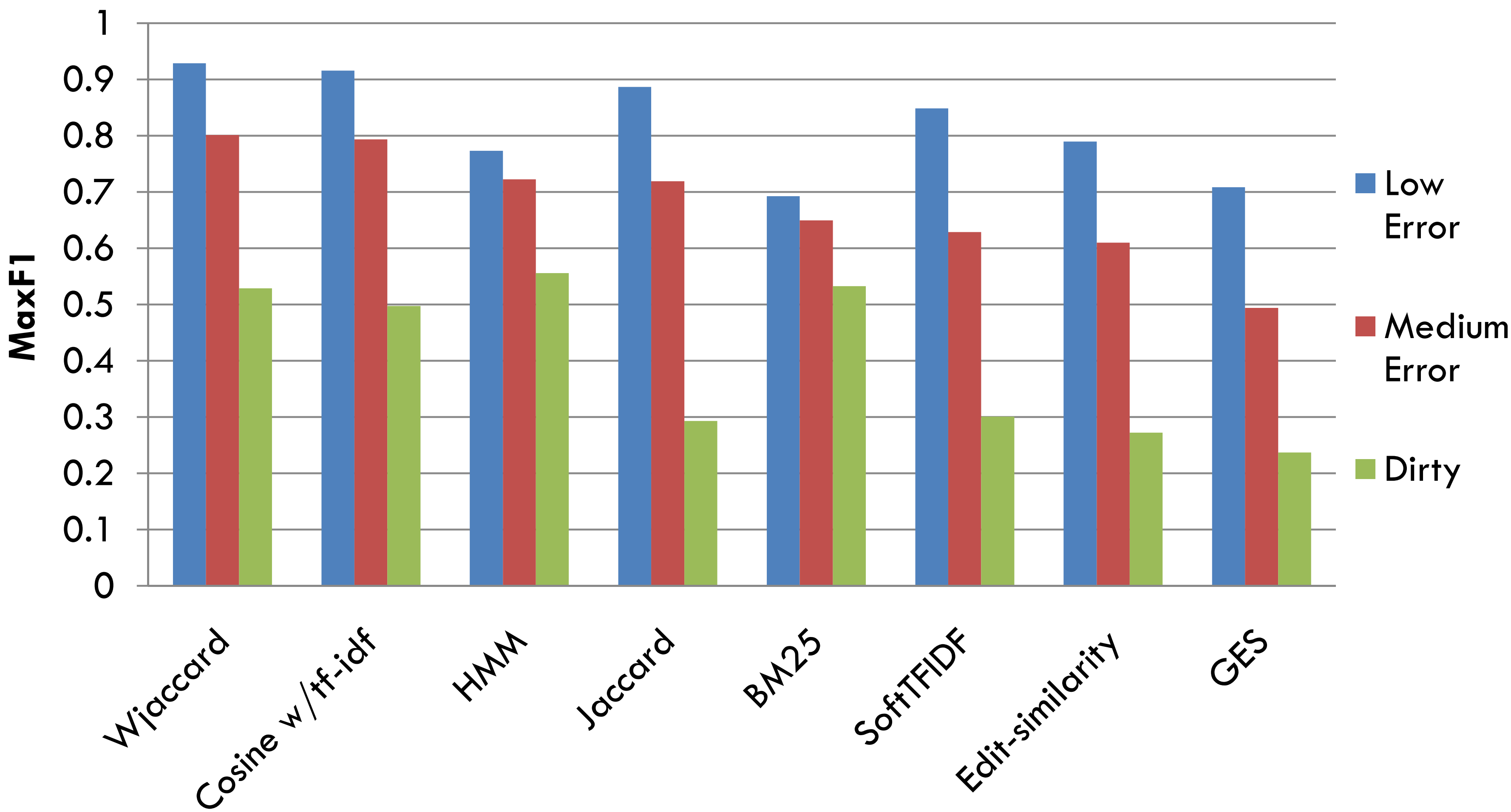
90% Erroneous duplicates 10% Errors in duplicates
20% Token swap 50% Abbreviation Error

Morgan Stanle Grop Incorporated
Stalney Morgan Group Inc.
Morgan Stanley Group In.
Stanley Moragn Grou Inc.
Morgan Stanley Group Inc.

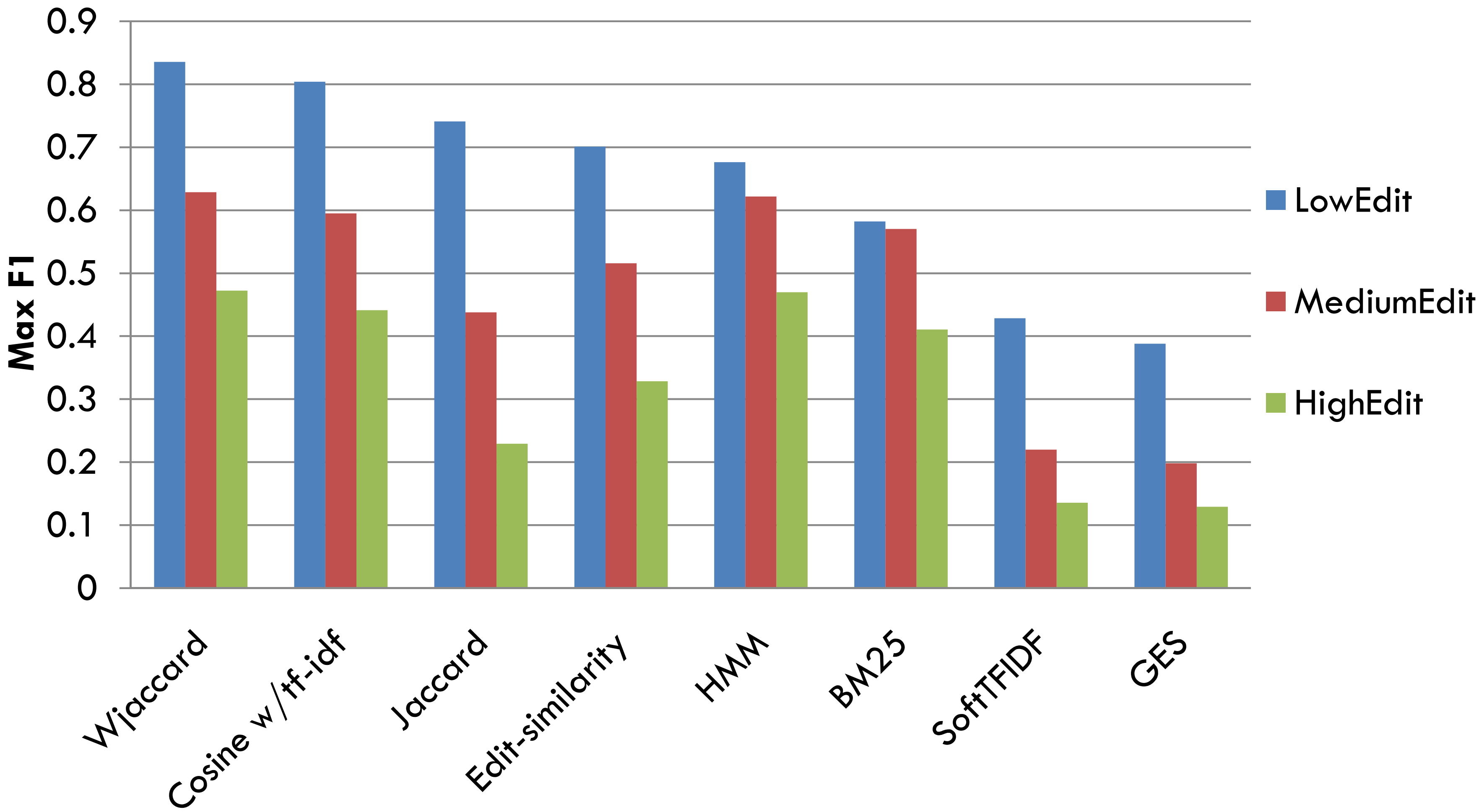
Results

- Effect of amount of errors on accuracy
- Effect of type of errors on accuracy
- Effect of threshold
  - maximum accuracy for different thresholds
- Comparison of thresholds that achieve
  - maximum accuracy vs. best performance

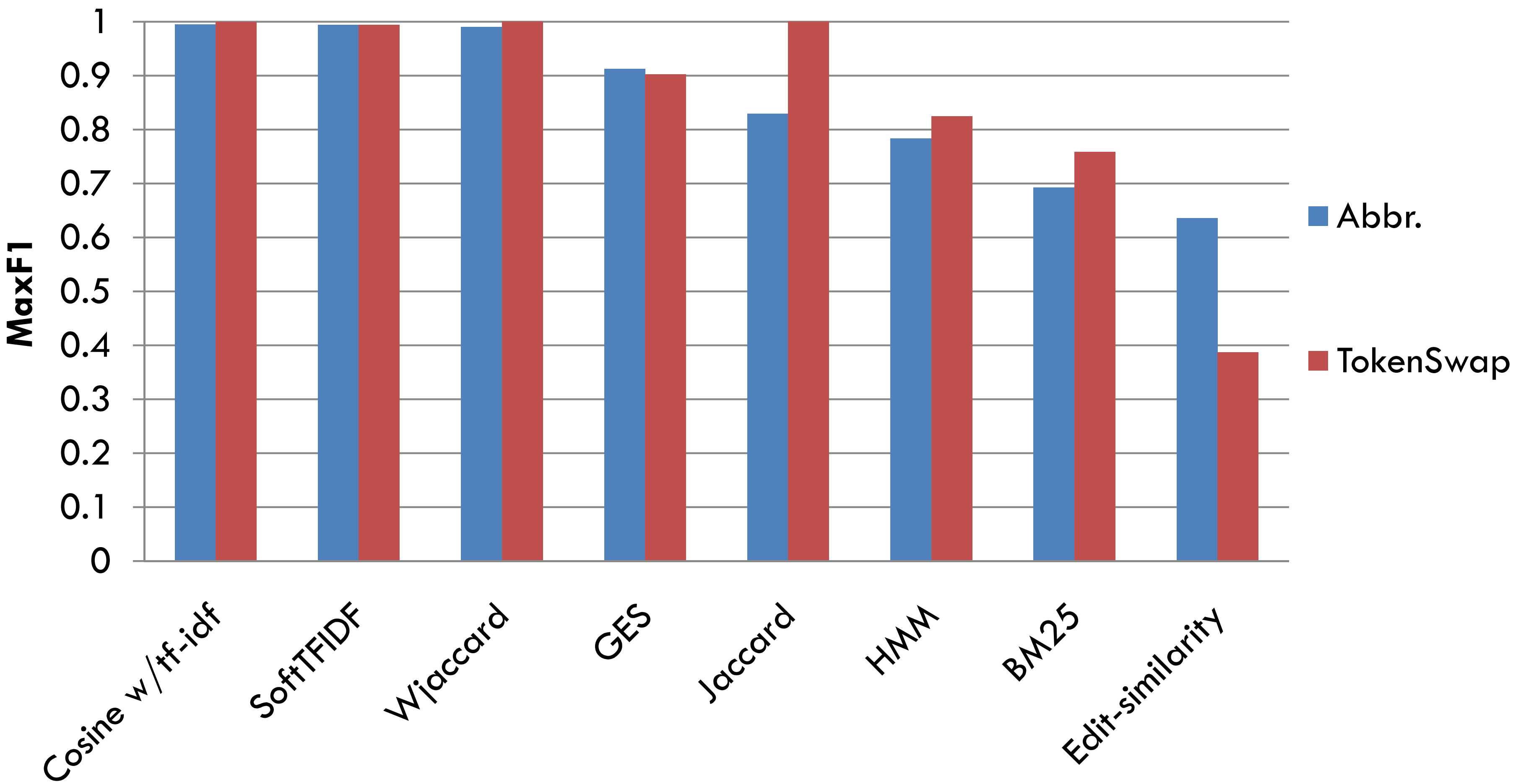
Effect of Amount of Errors



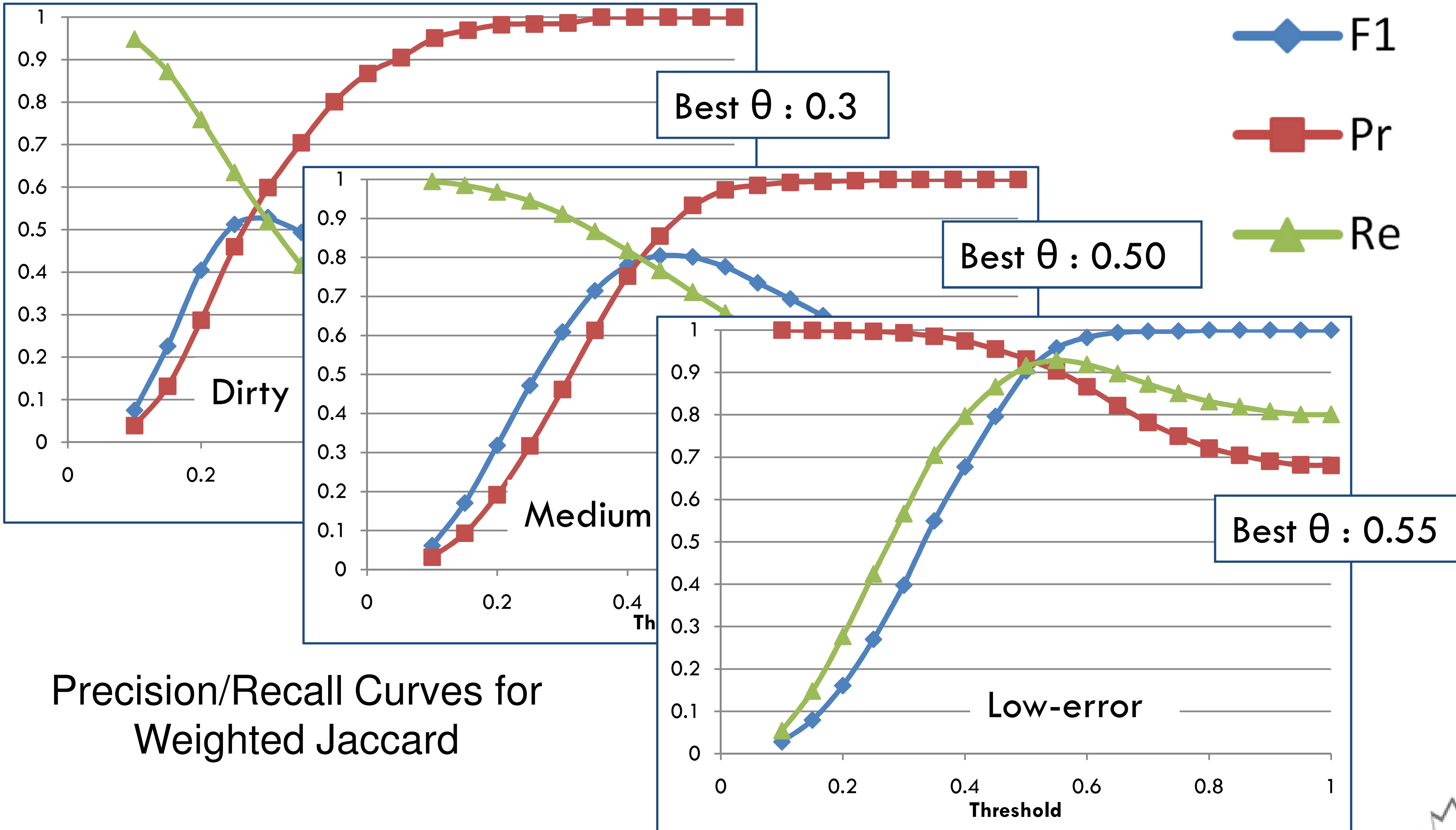
Effect of Edit Errors



Effect of Abbr. & Token-swap Errors



Effect of Threshold



Precision/Recall Curves for Weighted Jaccard

Accuracy in Efficient Techniques

- Performance of some recent techniques depends on the value of the threshold
  - PartEnum and WtEnum outperform LSH when threshold > 0.85

Jaccard Join		Weighted Jaccard Join	
Threshold	F1	Threshold	F1
0.65 (Best Accuracy)	0.719	0.50 (Best Accuracy)	0.801
0.80	0.611	0.80	0.581
0.85	0.571	0.85	0.581
0.90 (Best Performance by PartEnum)	0.548	0.90 (Best Performance by WtEnum)	0.560

On Medium-Error Datasets

Conclusion

- Simple overlap measures (weighted Jaccard) as accurate as complex hybrid and IR measures
  - **Future work:** Seeking more accurate similarity measures for string matching
- The value of the threshold that results in the most accurate join depends on the type and amount of errors in the data
  - **Future work:** Determining the value of the threshold for the most accurate measures
- There is a gap in recent work on efficient similarity join: improved performance may result in low accuracy
  - **Future work:** Finding algorithms that are both efficient and accurate, and evaluation of the accuracy of previously proposed techniques

The End

Questions ?

References

- [Tutorial-VLDB05] Approximate Joins: Concepts and Techniques.

  N. Koudas and D. Srivastava. VLDB'05 Tutorial
- [Tutorial-SIGMOD'06] Record linkage: similarity measures and algorithms.

  N. Koudas, S. Sarawagi and D. Srivastava. SIGMOD'06 Tutorial
- [DDSurvey-TKDE'07] Duplicate Record Detection: A Survey

  Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis and Vassilios S. Verykios. In IEEE Transactions on Knowledge and Data Engineering
- [NameMatching-IJCAI'03] A Comparison of String Distance Metrics for Name-Matching Tasks

  William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg. In IJCAI-03 Workshop on Information Integration on the Web
- [D.App.σ-SIGMOD'07] Benchmarking declarative approximate selection predicates.
  A. Chandel, O. Hassanzadeh, N.Koudas, M. Sadoghi, and D. Srivastava. In SIGMOD'07.

References

- [LSH, STOC'97] Locality-preserving hashing in multidimensional spaces.
  Indyk, Motwani, Raghavan, and Vempala. In STOC'97.
- [MP-LSH-VLDB'07] Multi-Probe LSH: Efficient Indexing for HighDimensional Similarity Search

  Qin Lv, William Josephson, Zhe Wang, Moses Charikar and Kai Li In VLDB'07
- [ExactSSJoin-VLDB'06] Efficient Exact Set Similarity Joins
  A. Arasu, V. Ganti, R. Kausshik, In VLDB'06
- [FMS-SIGMOD'03] Robust and efficient fuzzy match for online data cleaning

  Surajit Chaudhuri, Kris Ganjam, Venkatesh Ganti and Rajeev Motwani. In SIGMOD'03

References

- [VGRAM-VLDB'07] VGRAM: Improving Performance of Approximate Queries on String Collections Using Variable-Length Grams

  Chen Li, Bin Wang and Xiaochun Yang. In VLDB'07
- [UnderstandingIDF-Jdoc'04] Understanding Inverse Document Frequency: On theoretical arguments for IDF

  Stephen Robertson. In Journal of Documentation 2004
- [LM-SIGIR'98] A Language Modeling Approach to Information Retrieval

  Jay M. Ponte and W. Bruce Croft In SIGIR'98
- [HMM-SIGIR'99] A hidden Markov model information retrieval system

  R. H. Miller, Tim Leek and Richard M. Schwartz in SIGIR'99
- [MergePurge-DMKD'98] M. A. Hern´andez and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery, 2(1):9–37,1998.

UofT:DB Group