Programmable Similarity for Record Matching

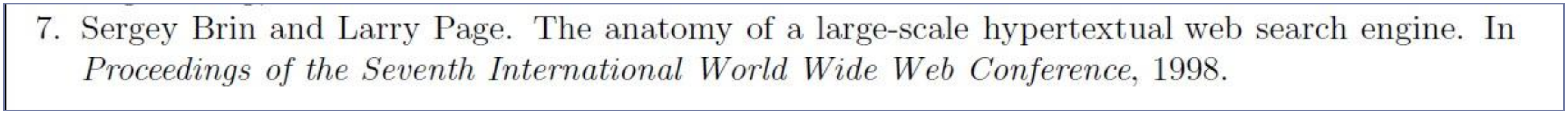
Arvind Arasu
Database Group, Microsoft Research

Joint work with Data Cleaning Research Team @ Microsoft Research

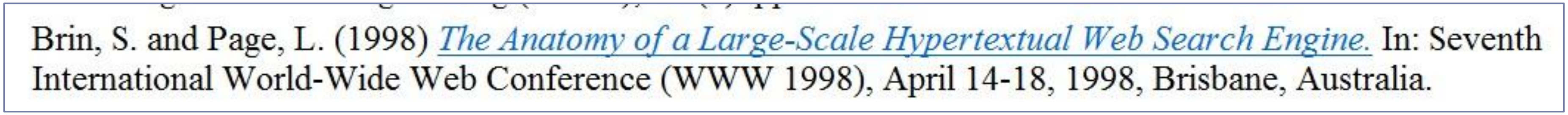
Record Matching

- Automate answering:
  - Do two records (texts) correspond to the same entity?
- Search and analysis applications:
  - Online map-services
    - Address matching
  - Citations: Citeseer, Google Scholar, Bing Academic
    - Citation matching
  - Comparative shopping sites
    - Product matching

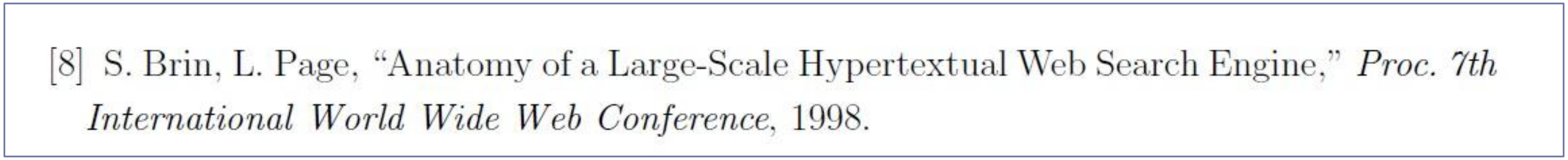
Example: Citations



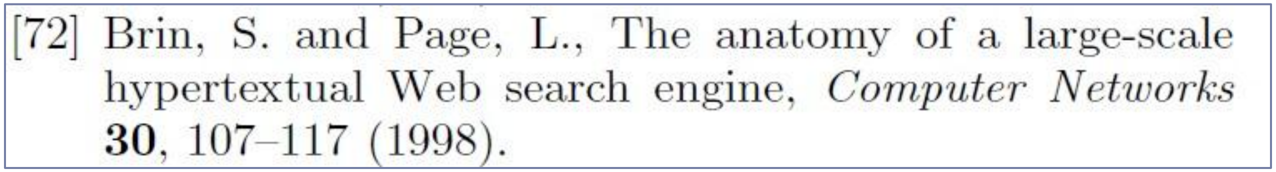
7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.



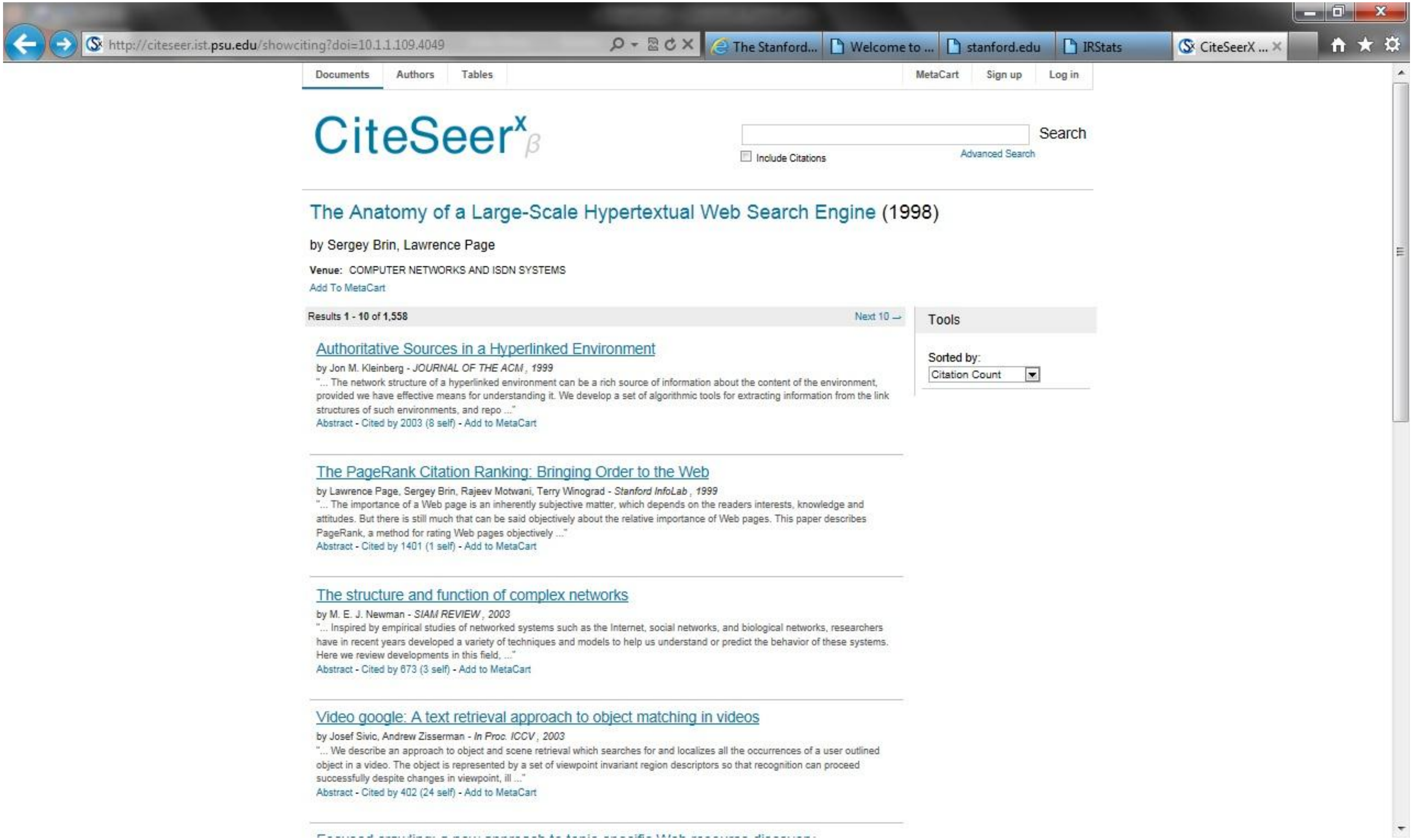
Brin, S. and Page, L. (1998) *The Anatomy of a Large-Scale Hypertextual Web Search Engine.* In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.



[8] S. Brin, L. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine," *Proc. 7th International World Wide Web Conference*, 1998.



[72] Brin, S. and Page, L., The anatomy of a large-scale hypertextual Web search engine, *Computer Networks* **30**, 107–117 (1998).



The Anatomy of a Large-Scale Hypertextual Web Search Engine (1998)

by Sergey Brin, Lawrence Page

Venue: COMPUTER NETWORKS AND ISDN SYSTEMS

Add To MetaCart

Results **1 - 10** of **1,558** Next 10 →

Authoritative Sources in a Hyperlinked Environment
by Jon M. Kleinberg - *JOURNAL OF THE ACM , 1999*
"... The network structure of a hyperlinked environment can be a rich source of information about the content of the environment, provided we have effective means for understanding it. We develop a set of algorithmic tools for extracting information from the link structures of such environments, and repo ..."
Abstract - Cited by 2003 (8 self) - Add to MetaCart

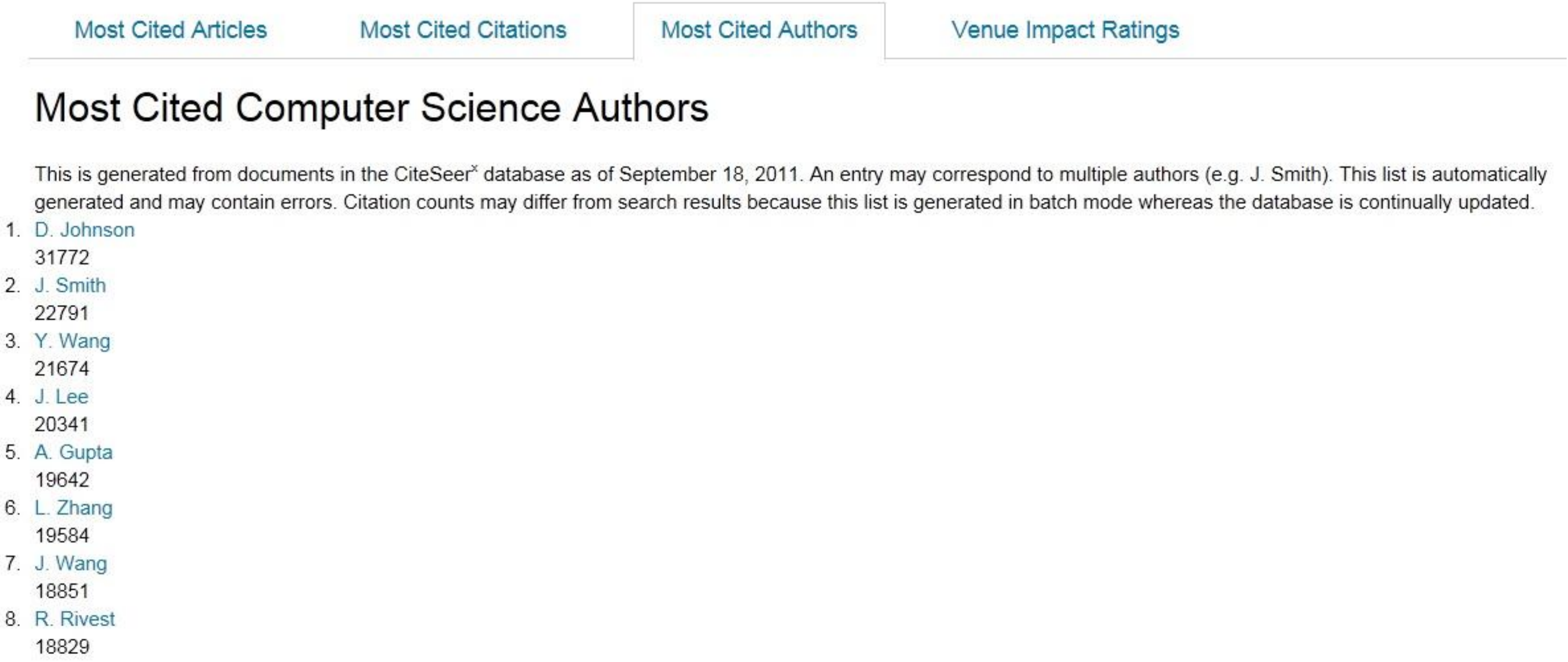
The PageRank Citation Ranking: Bringing Order to the Web
by Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd - *Stanford InfoLab , 1999*
"... The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively ..."
Abstract - Cited by 1401 (1 self) - Add to MetaCart

The structure and function of complex networks
by M. E. J. Newman - *SIAM REVIEW , 2003*
"... Inspired by empirical studies of networked systems such as the Internet, social networks, and biological networks, researchers have in recent years developed a variety of techniques and models to help us understand or predict the behavior of these systems. Here we review developments in this field, ..."
Abstract - Cited by 673 (3 self) - Add to MetaCart

Video google: A text retrieval approach to object matching in videos
by Josef Sivic, Andrew Zisserman - *In Proc. ICCV , 2003*
"... We describe an approach to object and scene retrieval which searches for and localizes all the occurrences of a user outlined object in a video. The object is represented by a set of viewpoint invariant region descriptors so that recognition can proceed successfully despite changes in viewpoint, ill ..."
Abstract - Cited by 402 (24 self) - Add to MetaCart

Focused crawling: a new approach to topic-specific Web resource discovery

Most cited computer science authors?

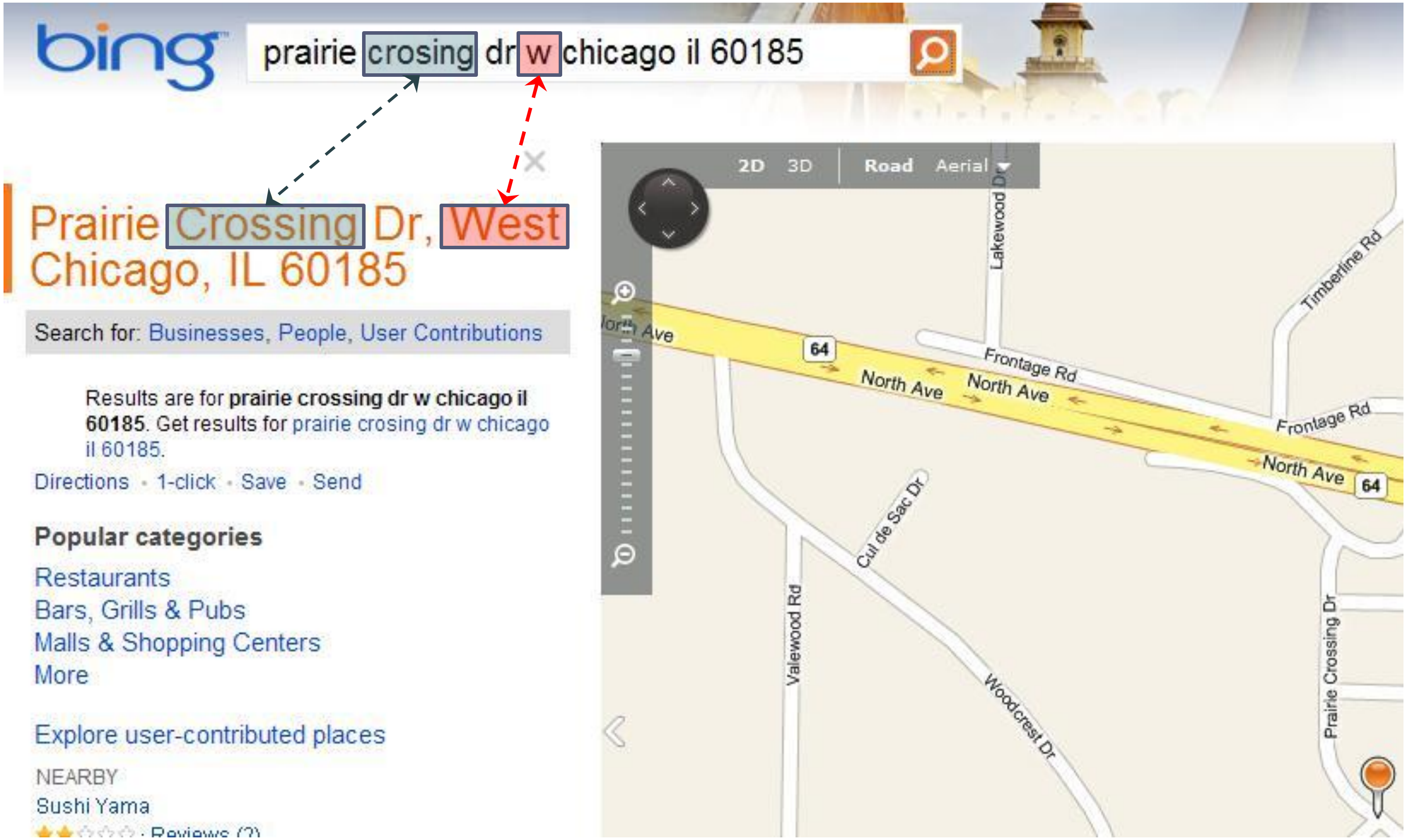


Most Cited Computer Science Authors

This is generated from documents in the CiteSeer^x database as of September 18, 2011. An entry may correspond to multiple authors (e.g. J. Smith). This list is automatically generated and may contain errors. Citation counts may differ from search results because this list is generated in batch mode whereas the database is continually updated.

1.	D. Johnson	31772
2.	J. Smith	22791
3.	Y. Wang	21674
4.	J. Lee	20341
5.	A. Gupta	19642
6.	L. Zhang	19584
7.	J. Wang	18851
8.	R. Rivest	18829

Address Matching



prairie crosing dr w chicago il 60185

Prairie Crossing Dr, West
Chicago, IL 60185

Search for: Businesses, People, User Contributions

Results are for **prairie crossing dr w chicago il 60185**. Get results for prairie crosing dr w chicago il 60185.

Directions · 1-click · Save · Send

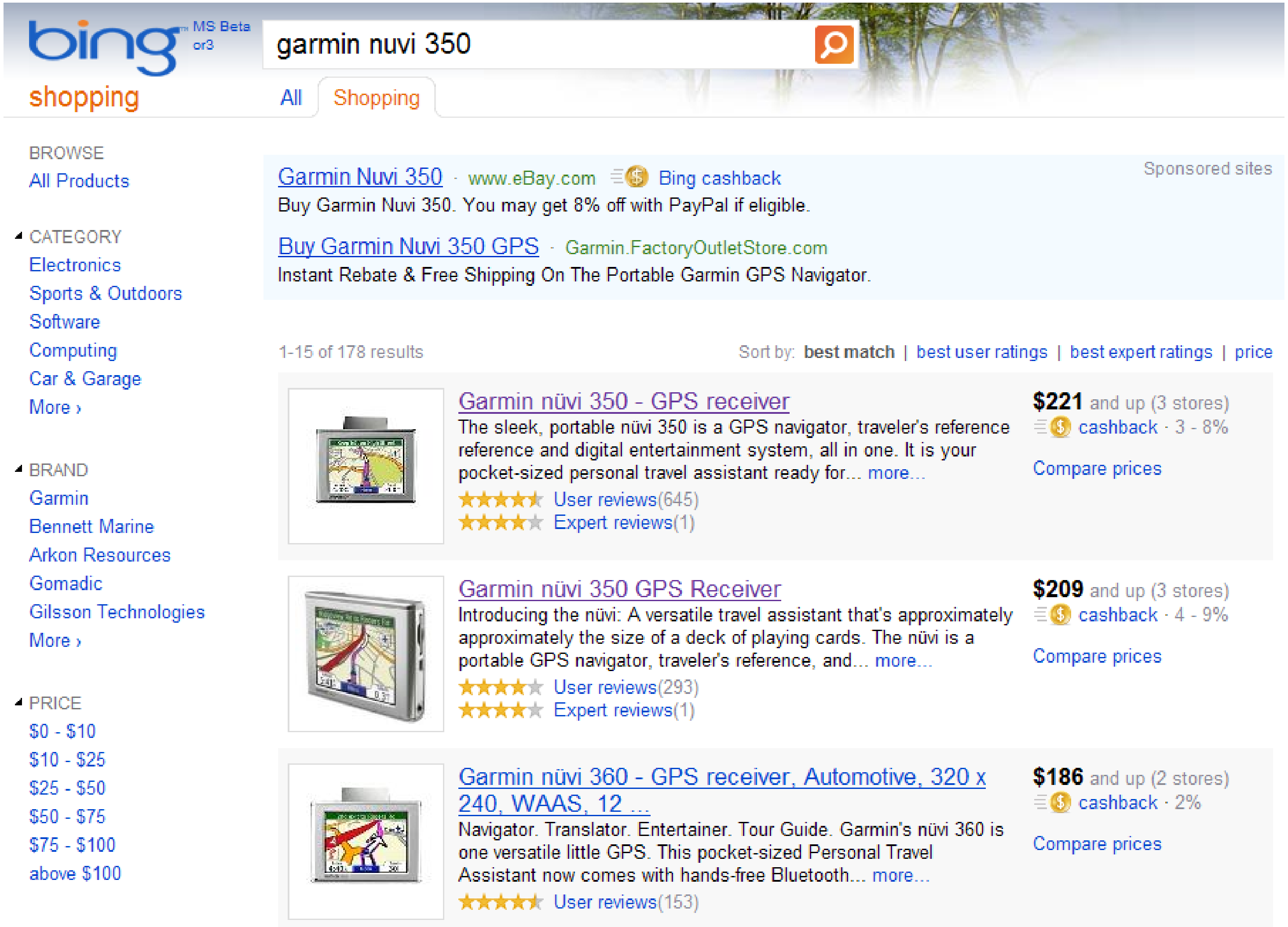
Popular categories

Restaurants
Bars, Grills & Pubs
Malls & Shopping Centers
More

Explore user-contributed places

NEARBY
Sushi Yama
· Reviews (2)

Product Matching



BROWSE
All Products

CATEGORY
Electronics
Sports & Outdoors
Software
Computing
Car & Garage
More ›

BRAND
Garmin
Bennett Marine
Arkon Resources
Gomadic
Gilsson Technologies
More ›

PRICE
$0 - $10
$10 - $25
$25 - $50
$50 - $75
$75 - $100
above $100

Sponsored sites

Garmin Nuvi 350 · www.eBay.com Bing cashback
Buy Garmin Nuvi 350. You may get 8% off with PayPal if eligible.

Buy Garmin Nuvi 350 GPS · Garmin.FactoryOutletStore.com
Instant Rebate & Free Shipping On The Portable Garmin GPS Navigator.

1-15 of 178 results

Sort by: **best match** | best user ratings | best expert ratings | price

Garmin nüvi 350 - GPS receiver
The sleek, portable nüvi 350 is a GPS navigator, traveler's reference reference and digital entertainment system, all in one. It is your pocket-sized personal travel assistant ready for... more...
★★★★★ User reviews(645)
★★★★☆ Expert reviews(1)
$221 and up (3 stores)
cashback · 3 - 8%
Compare prices

Garmin nüvi 350 GPS Receiver
Introducing the nüvi: A versatile travel assistant that's approximately approximately the size of a deck of playing cards. The nüvi is a portable GPS navigator, traveler's reference, and... more...
★★★★☆ User reviews(293)
★★★★☆ Expert reviews(1)
$209 and up (3 stores)
cashback · 4 - 9%
Compare prices

Garmin nüvi 360 - GPS receiver, Automotive, 320 x 240, WAAS, 12 ...
Navigator. Translator. Entertainer. Tour Guide. Garmin's nüvi 360 is one versatile little GPS. This pocket-sized Personal Travel Assistant now comes with hands-free Bluetooth... more...
★★★★★ User reviews(153)
$186 and up (2 stores)
cashback · 2%
Compare prices

Record Matching: State-of-the-art

- Problem Characteristics:
  - AI-complete?
  - Classification problem
- Standard approach:
  - Textual similarity as a signal

Textual Similarity for Matching

Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World Wide Web Conference, 1998.

Brin, S. and Page, L. (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Textual Similarity for Matching

Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World Wide Web Conference, 1998.

Jon M. Kleinberg: Authoritative Sources in a Hyperlinked Environment. J. ACM 46(5): 604-632 (1999)

Record Matching: State-of-the-art

- Problem Characteristics:
  - AI-complete?
  - Classification problem
- Standard approach:
  - Textual similarity as a signal
  - Details:
    - Combining similarities from different columns (signals)
    - Learning approaches
    - Performance optimizations
  - Focus of this talk: ***How to measure textual (string) similarity?***

Overview

- Introduction
- Textual Similarity
  - Limitations of current similarity functions
- Programmable Similarity
  - Semantics
  - Usability
  - Performance
- Conclusion

Textual Similarity

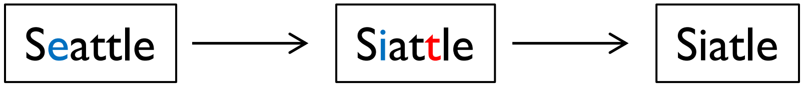
- String Similarity Function:
  - $Sim(string, string) \rightarrow numeric\ value$
- A "good" similarity function:
  - Strings representing the same concept $\Longrightarrow$ high similarity
  - Strings representing different concepts $\Longrightarrow$ low similarity

Edit Distance

- EditDistance (s1, s2): Minimum number of edits to transform s1 to s2
- Edit:
  - Insert a character
  - Delete a character
  - Substitute a character
- Note: EditDistance(s1, s2) = EditDistance (s2, s1)
- “distance” opposite of “similarity”

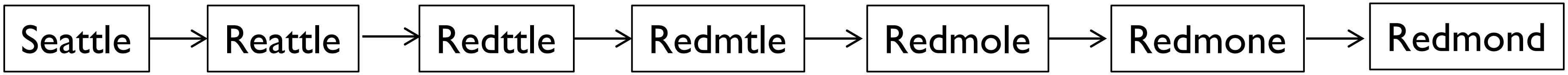
Edit Distance

EditDistance (“Seattle”, “Siatle”) = 2



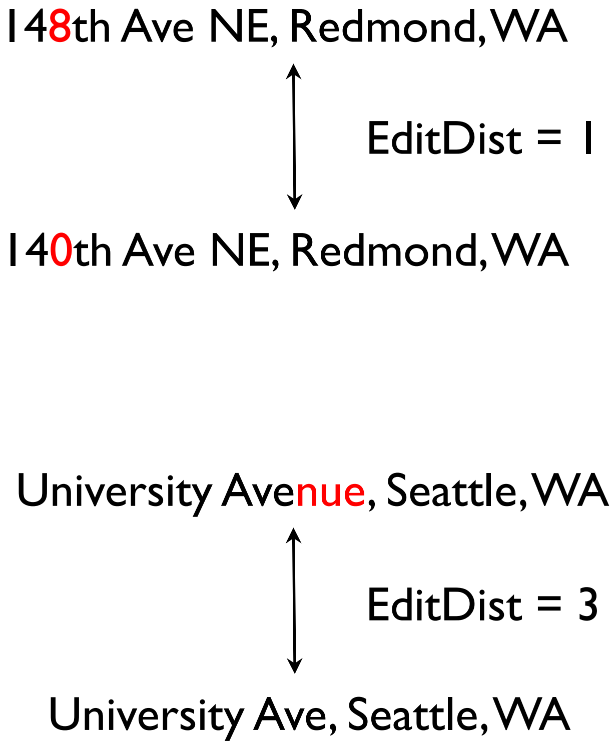
Seattle → Siattle → Siatle

EditDistance (“Seattle”, “Redmond”) = 6



Seattle → Reattle → Redttle → Redmtle → Redmole → Redmone → Redmond

Edit Distance Limitations



148th Ave NE, Redmond, WA

EditDist = 1

140th Ave NE, Redmond, WA

University Avenue, Seattle, WA

EditDist = 3

University Ave, Seattle, WA

Jaccard Similarity

- Statistical measure
- Originally defined over sets
- String = set of words

$$Jaccard(s1, s2) = \frac{|s1 \ \cap s2|}{|s1 \ \cup s2|}$$

- Range of values: [0,1]

Jaccard Similarity

148th Ave NE, Redmond, WA

↕

140th Ave NE, Redmond, WA

$$Jaccard = \frac{4}{4+2} \approx 0.66$$

Weighted Jaccard Similarity

$$Weight\ Function = \ wt\colon Elements \to \ \mathbb{R}^{+}$$

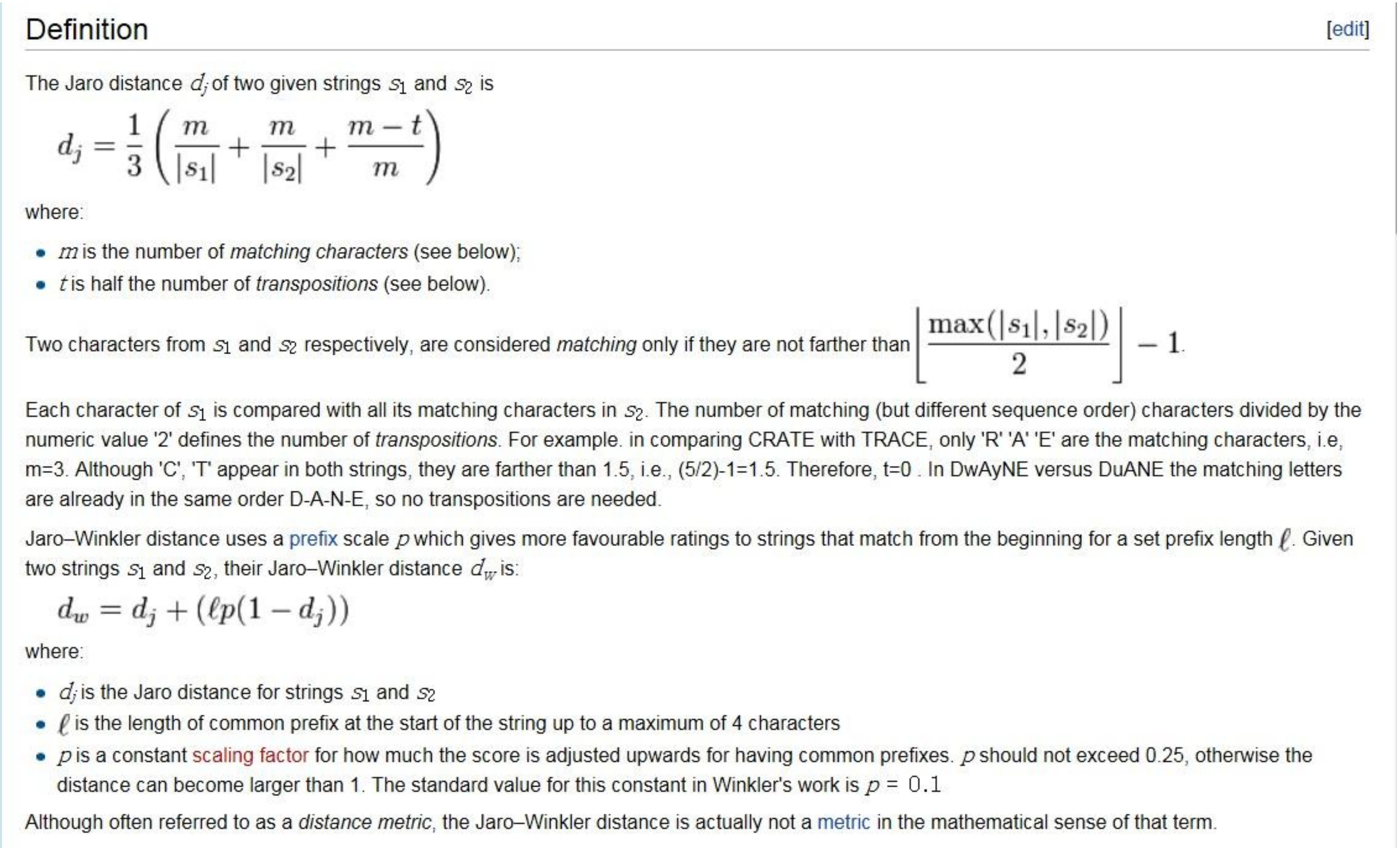
$$\boxed{WtJaccard(s1, s2) = \frac{wt(s1\ \cap s2)}{wt(s1\ \cup s2)}}$$

$$wt(s) = \sum_{e\ \in s} wt(e)$$

List of other Similarity Functions

- Affine edit distance
- Cosine similarity
- Hamming distance
- Generalized edit distance
- Jaro distance
- Monge-Elkan distance
- Q-gram
- Smith-Waterman distance
- Soundex
- TF/IDF
- … many more

Jaro-Winkler distance



Definition

[edit]

The Jaro distance d_j of two given strings s_1 and s_2 is

$$d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

where:

- m is the number of *matching characters* (see below);
- t is half the number of *transpositions* (see below).

Two characters from s_1 and s_2 respectively, are considered *matching* only if they are not farther than $\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1$.

Each character of s_1 is compared with all its matching characters in s_2. The number of matching (but different sequence order) characters divided by the numeric value '2' defines the number of *transpositions*. For example. in comparing CRATE with TRACE, only 'R' 'A' 'E' are the matching characters, i.e, m=3. Although 'C', 'T' appear in both strings, they are farther than 1.5, i.e., (5/2)-1=1.5. Therefore, t=0 . In DwAyNE versus DuANE the matching letters are already in the same order D-A-N-E, so no transpositions are needed.

Jaro–Winkler distance uses a prefix scale p which gives more favourable ratings to strings that match from the beginning for a set prefix length ℓ. Given two strings s_1 and s_2, their Jaro–Winkler distance d_w is:

$$d_w = d_j + (\ell p(1 - d_j))$$

where:

- d_j is the Jaro distance for strings s_1 and s_2
- ℓ is the length of common prefix at the start of the string up to a maximum of 4 characters
- p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. p should not exceed 0.25, otherwise the distance can become larger than 1. The standard value for this constant in Winkler's work is $p = 0.1$

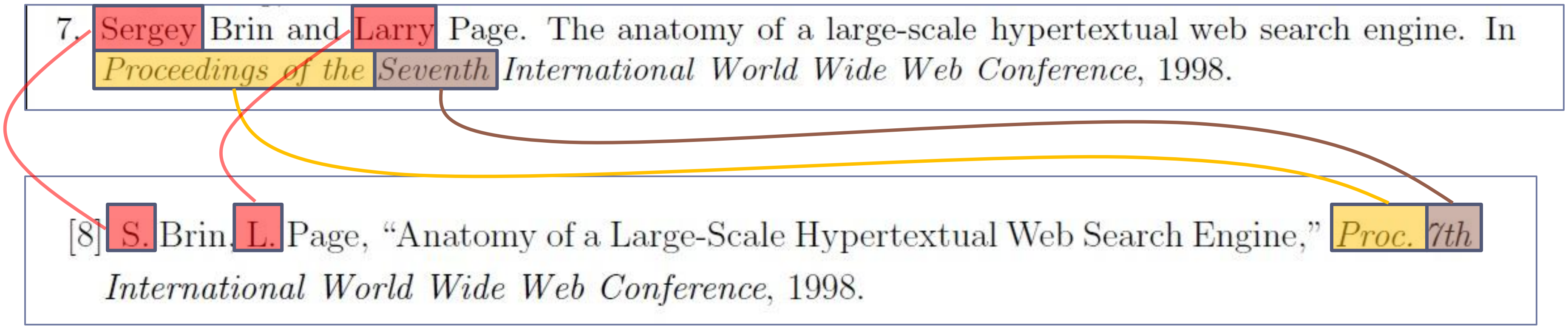
Although often referred to as a *distance metric*, the Jaro–Winkler distance is actually not a metric in the mathematical sense of that term.

List of other Similarity Functions

- Affine edit distance
- Cosine similarity
- Hamming distance
- Generalized edit distance
- Jaro distance
- Monge-Elkan distance
- Q-gram
- Smith-Waterman distance
- Soundex
- TF/IDF
- … many more

- Limitation: “variations” syntactic & predefined

Complex Variations



7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

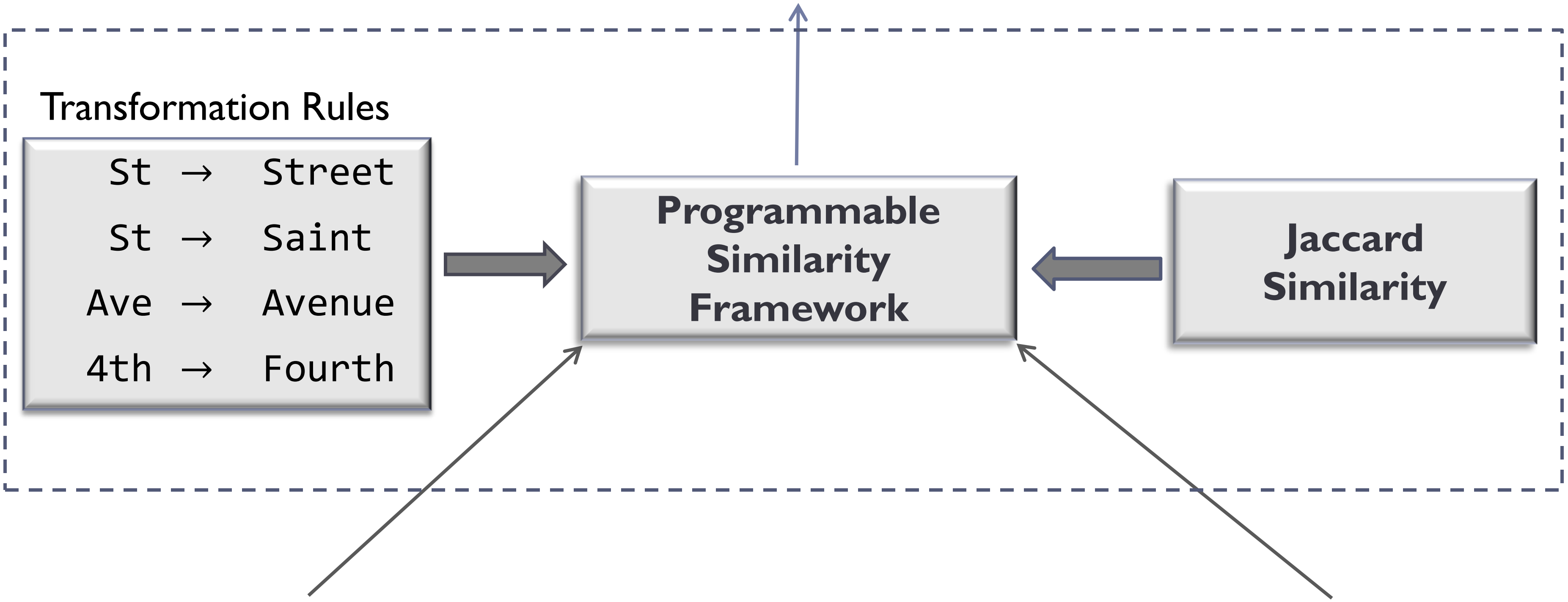
[8] S. Brin, L. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine," *Proc. 7th International World Wide Web Conference*, 1998.

- Synonyms
- Abbreviations
- Missing/additional information

Our approach

- Programmable Similarity:
  - Simple similarity function (Jaccard)
  - Variations as explicit input (“program”)

Programmable Similarity



Transformation Rules

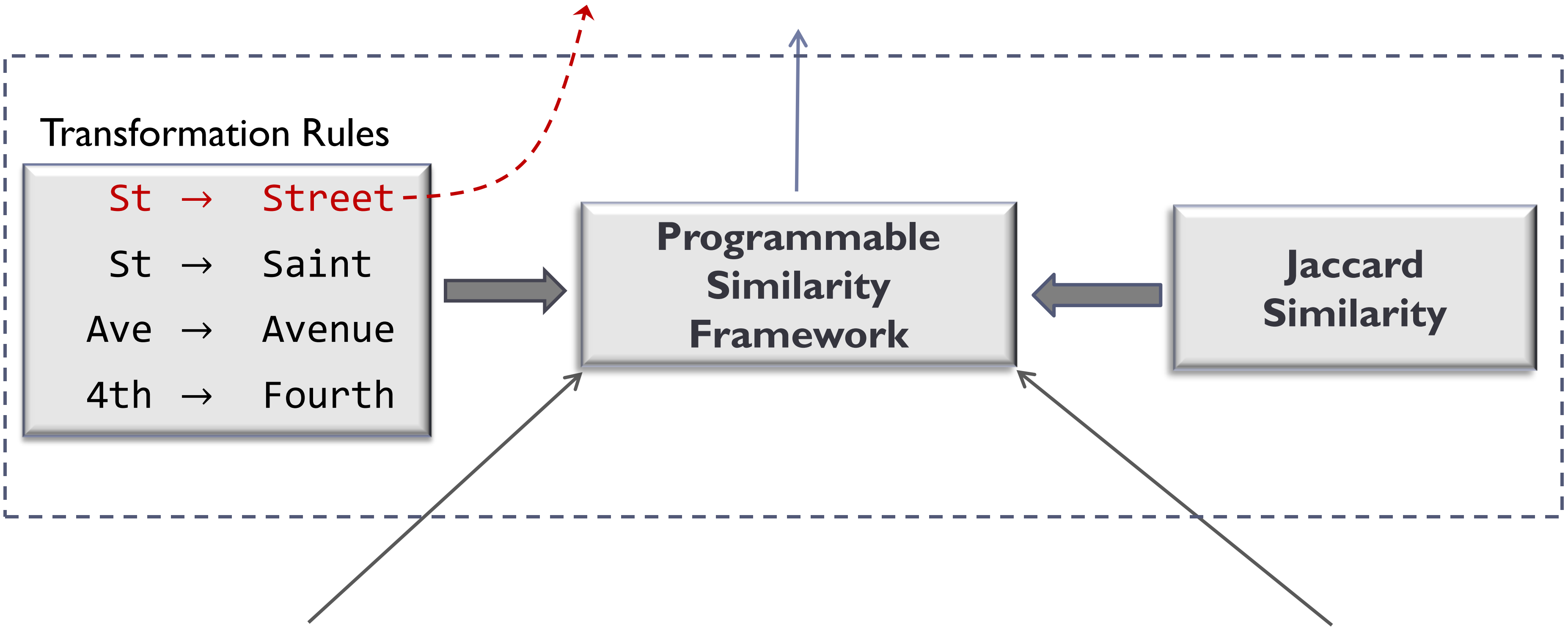
St	→	Street
St	→	Saint
Ave	→	Avenue
4th	→	Fourth

Programmable Similarity Framework

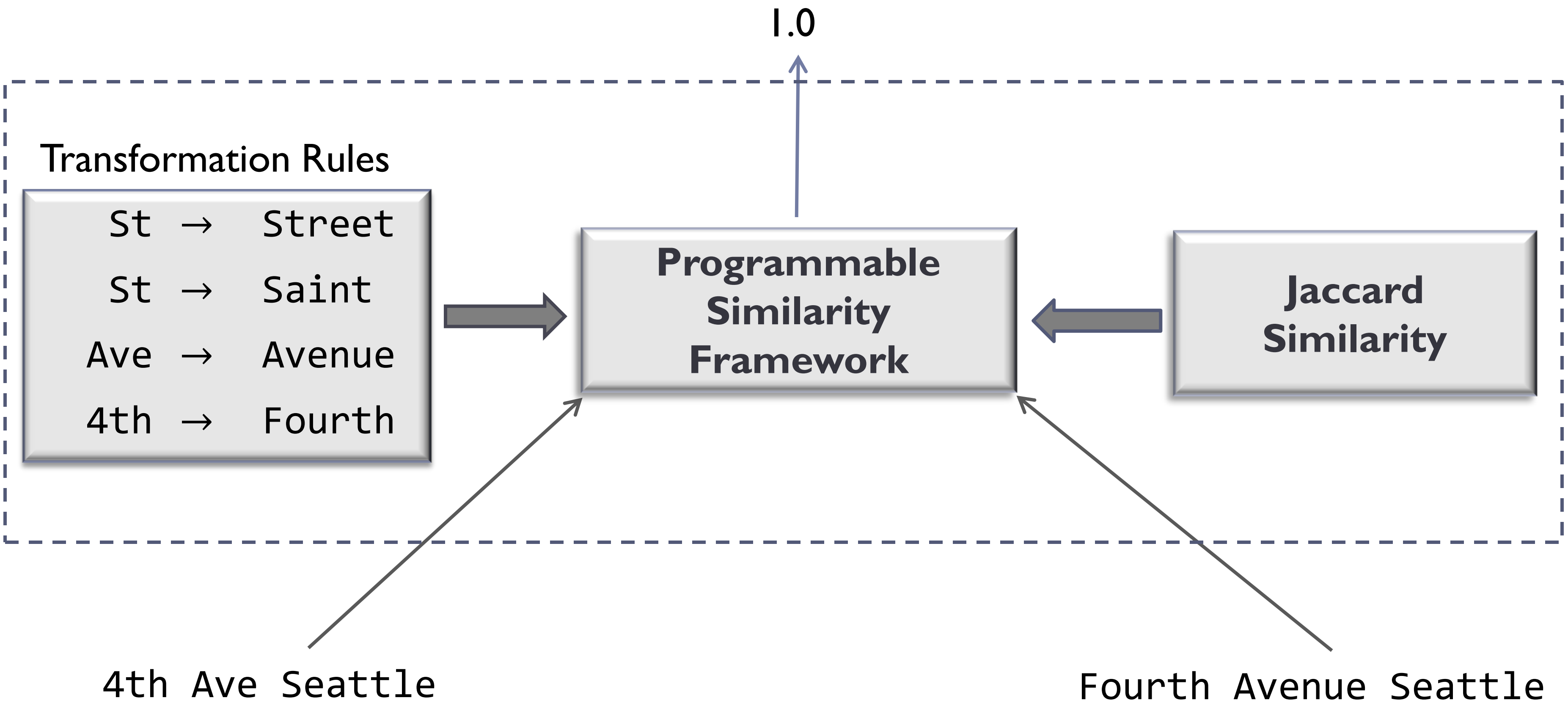
Jaccard Similarity

Programmable Similarity

St is an alternate representation of Street



Programmable Similarity



1.0

Transformation Rules

St	→	Street
St	→	Saint
Ave	→	Avenue
4th	→	Fourth

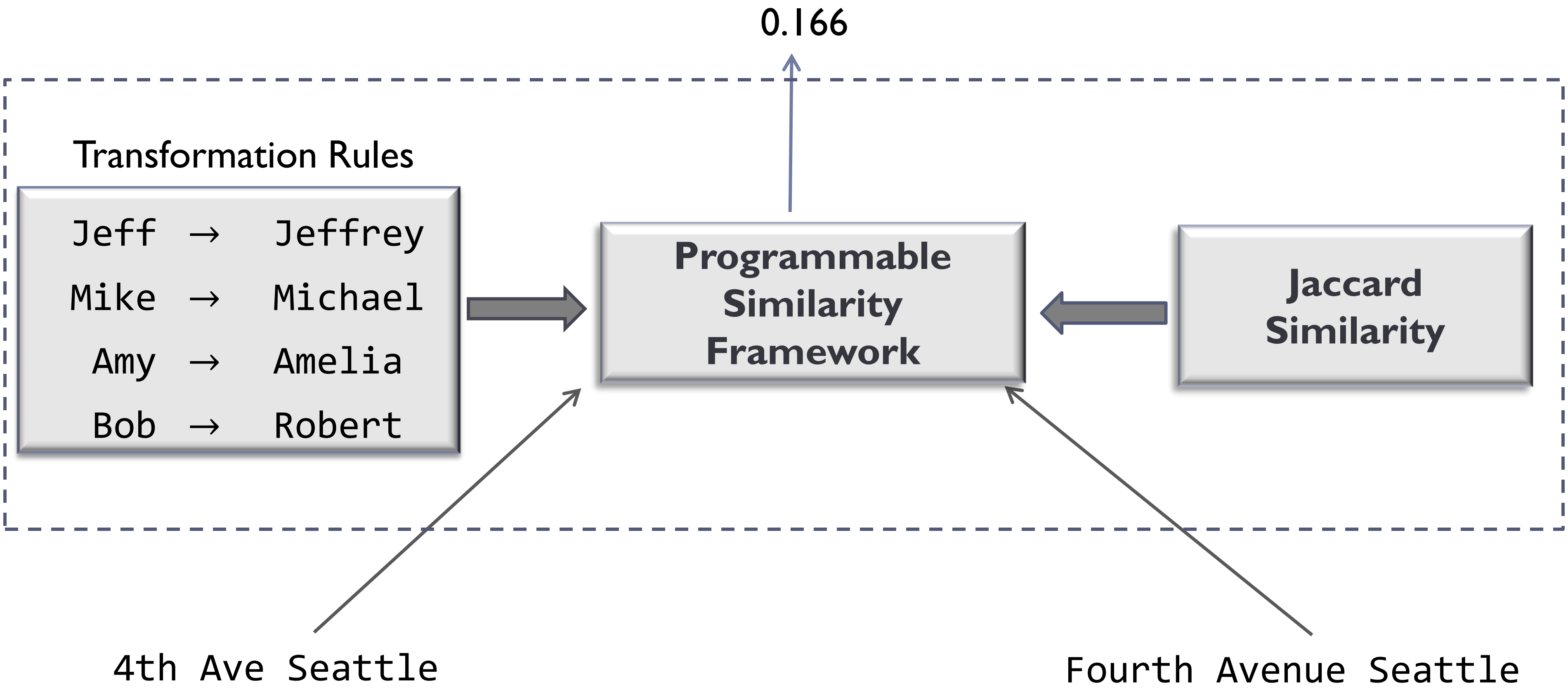
Programmable Similarity Framework

Jaccard Similarity

`4th Ave Seattle`

`Fourth Avenue Seattle`

Programmable Similarity



0.166

Transformation Rules

Jeff	→	Jeffrey
Mike	→	Michael
Amy	→	Amelia
Bob	→	Robert

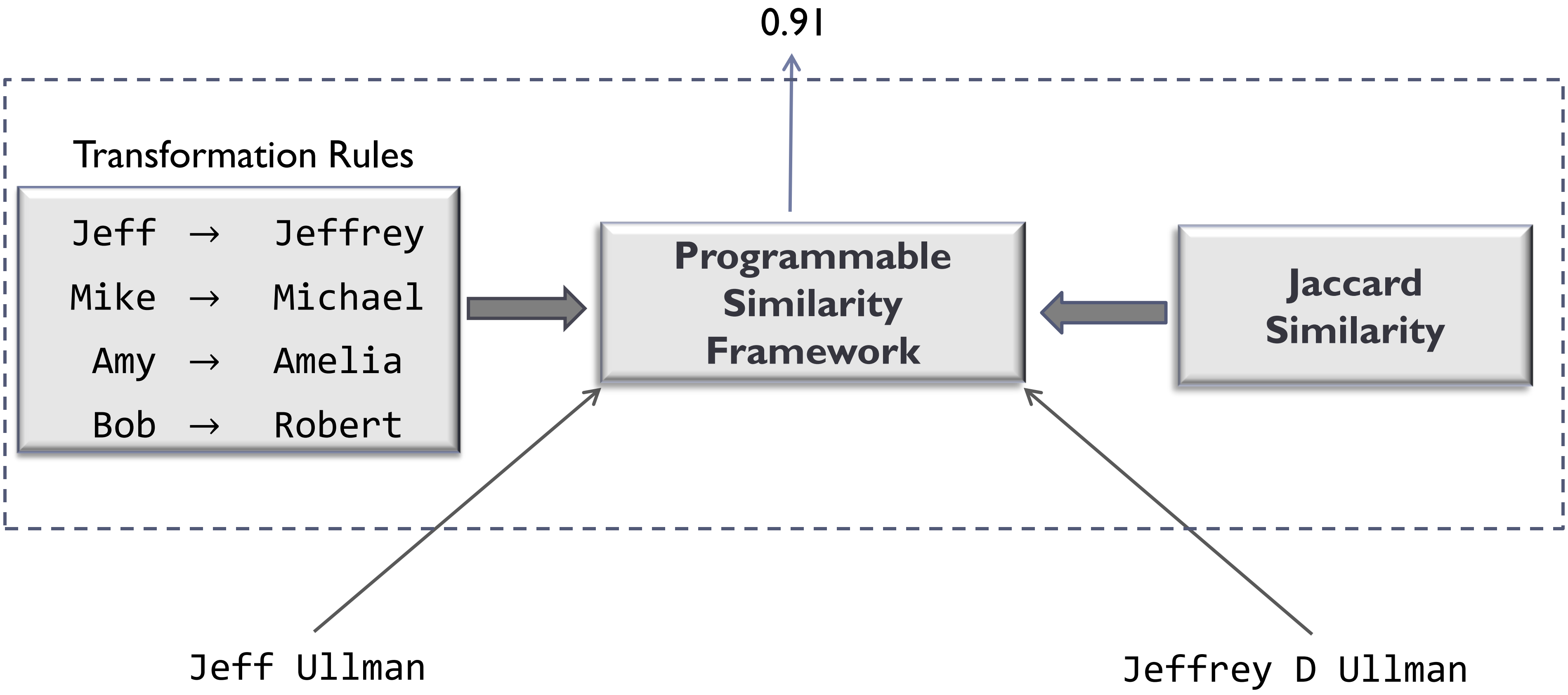
Programmable Similarity Framework

Jaccard Similarity

4th Ave Seattle

Fourth Avenue Seattle

Programmable Similarity

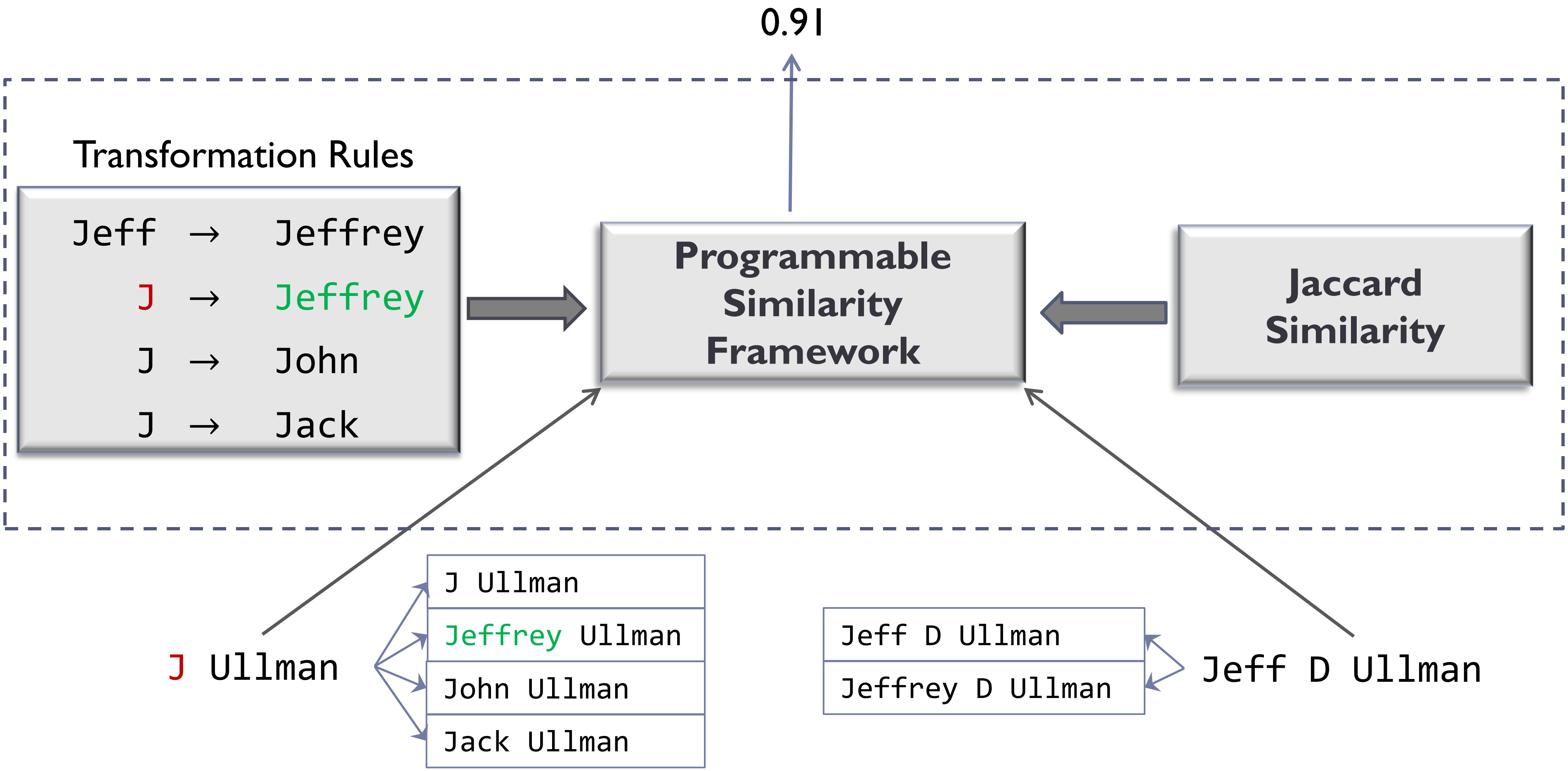


0.91

Transformation Rules

Jeff	→	Jeffrey
Mike	→	Michael
Amy	→	Amelia
Bob	→	Robert

Programmable Similarity: Semantics



0.91

Transformation Rules

Jeff	→	Jeffrey
J	→	Jeffrey
J	→	John
J	→	Jack

Programmable Similarity Framework

Jaccard Similarity

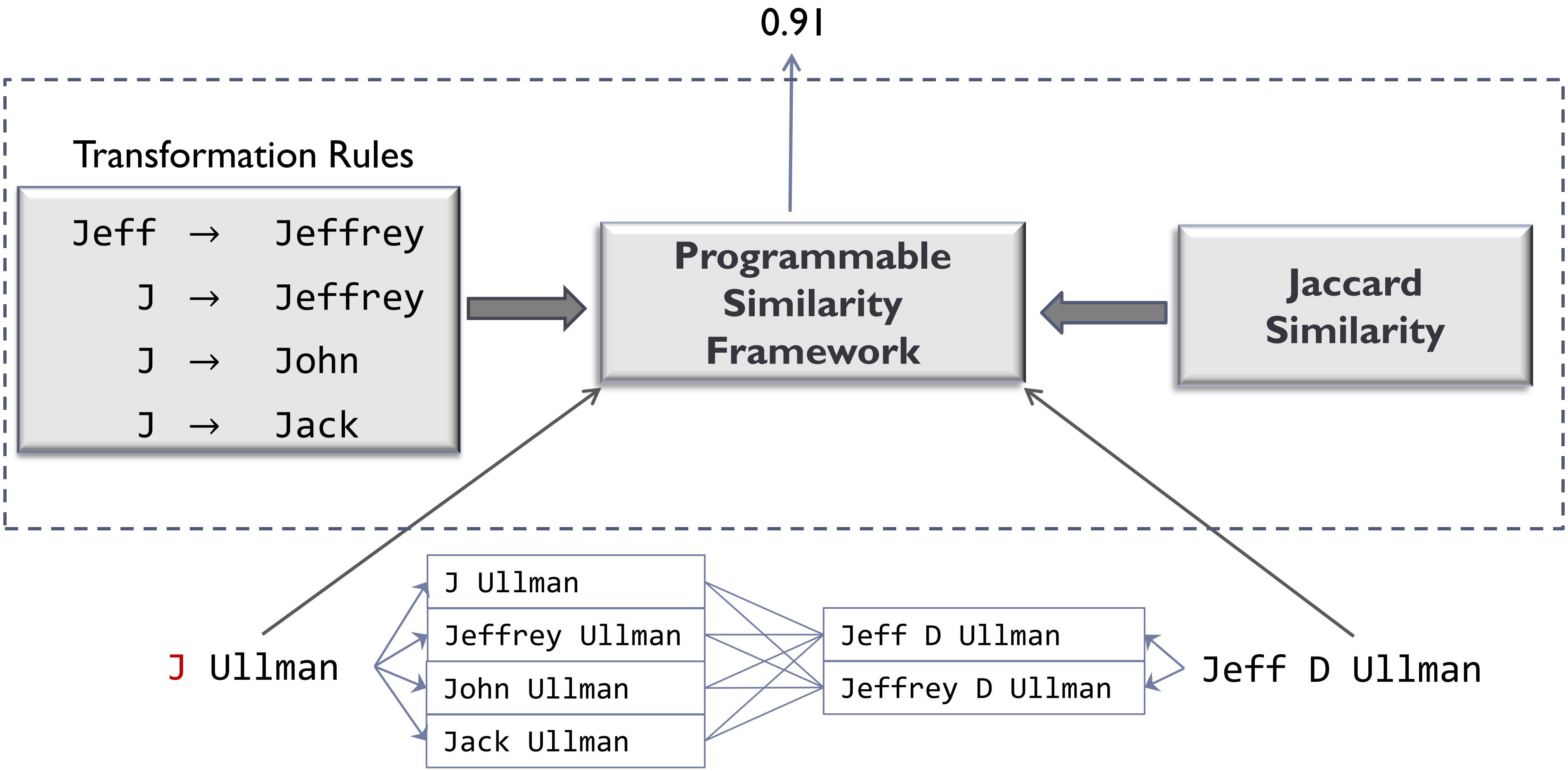
J Ullman

- J Ullman
- Jeffrey Ullman
- John Ullman
- Jack Ullman

Jeff D Ullman

- Jeff D Ullman
- Jeffrey D Ullman

Programmable Similarity: Semantics



0.91

Transformation Rules

Jeff	→	Jeffrey
J	→	Jeffrey
J	→	John
J	→	Jack

Programmable Similarity Framework

Jaccard Similarity

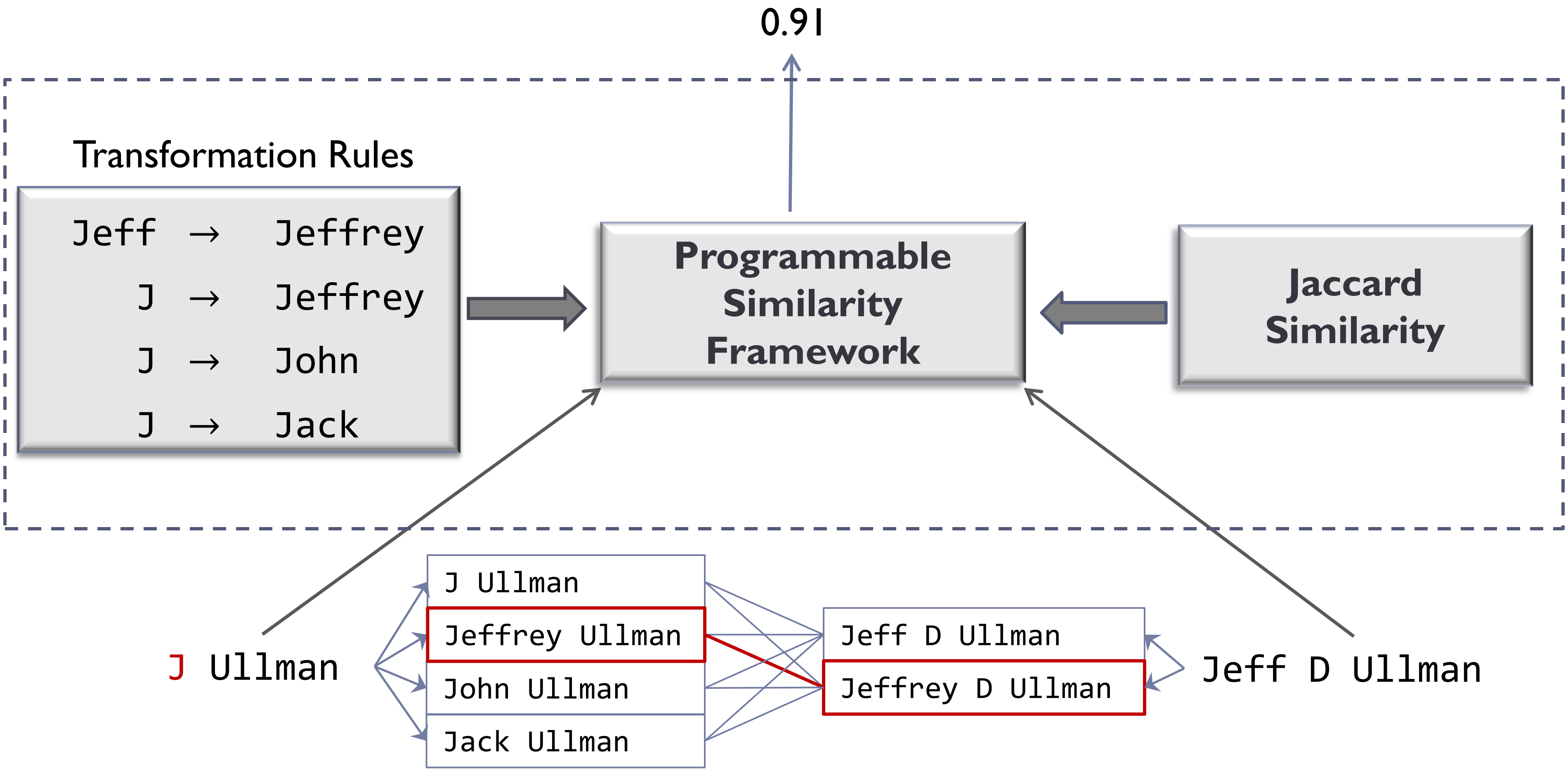
J Ullman

J Ullman
Jeffrey Ullman
John Ullman
Jack Ullman

Jeff D Ullman
Jeffrey D Ullman

Jeff D Ullman

Programmable Similarity: Semantics



0.91

Transformation Rules

Jeff	→	Jeffrey
J	→	Jeffrey
J	→	John
J	→	Jack

Programmable Similarity Framework

Jaccard Similarity

J Ullman

J Ullman

Jeffrey Ullman

John Ullman

Jack Ullman

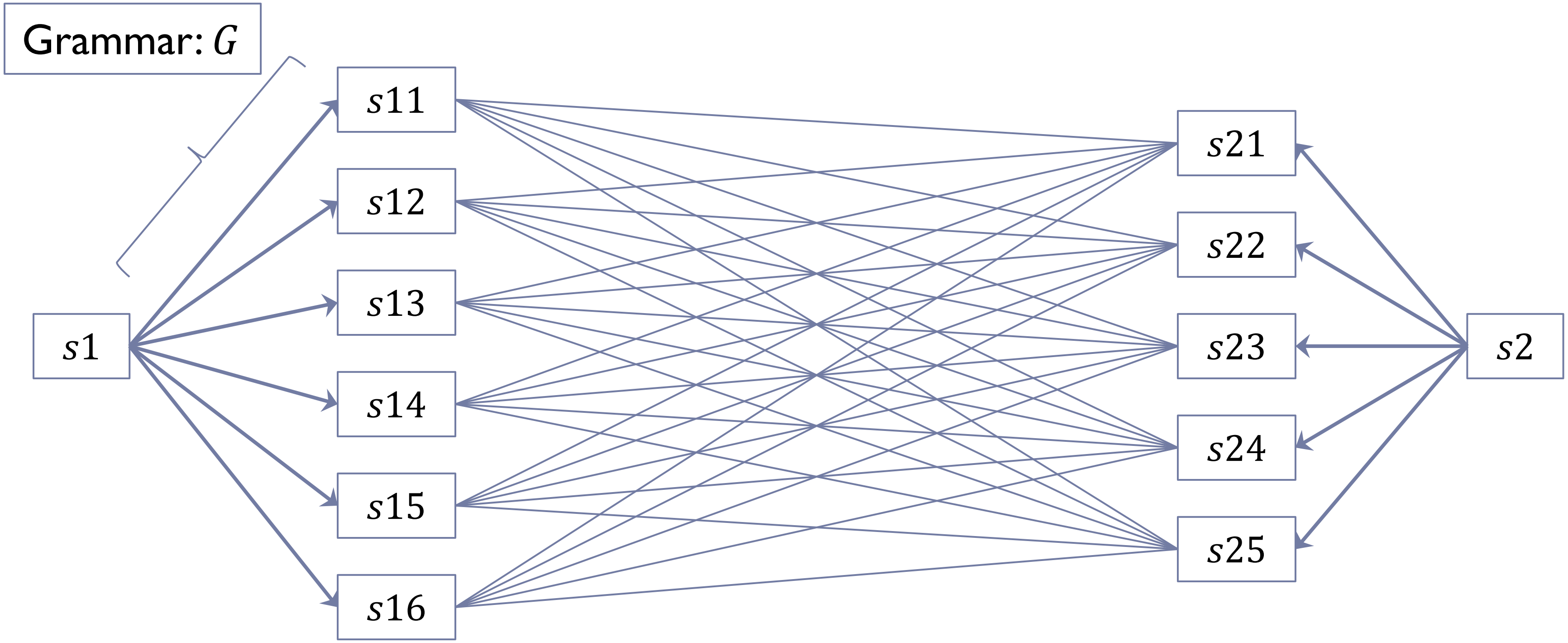
Jeff D Ullman

Jeffrey D Ullman

Jeff D Ullman

Programmable Similarity: Semantics

$ProgSim(s1, s2)$: $\text{Max}_{j,k}\ Jaccard(s1j, s2k)$



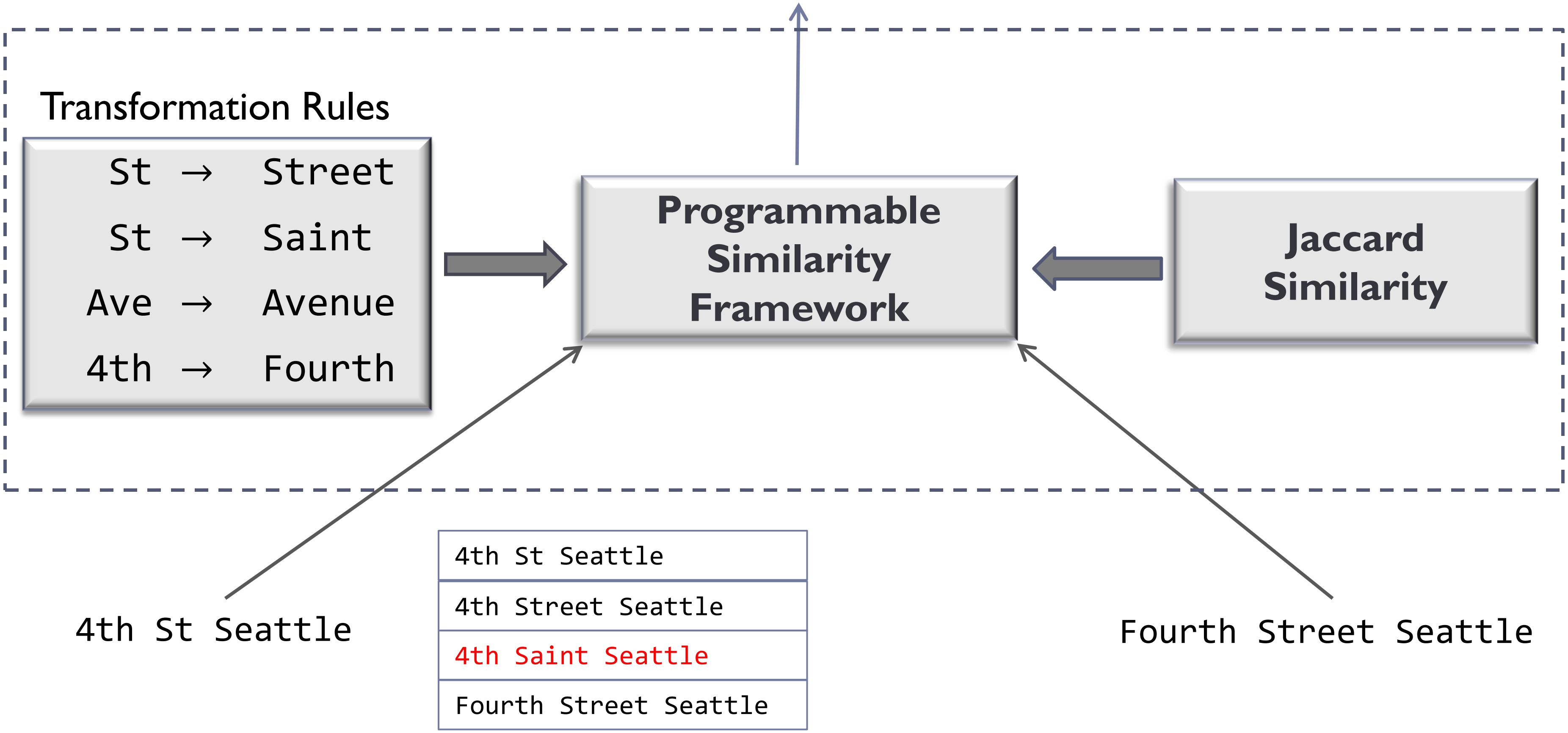
Variants of $s1$

Variants of $s2$

Overview

- Introduction
- Textual Similarity
  - Limitations of current similarity functions
- Programmable Similarity
  - Semantics
  - Usability
  - Performance
- Conclusion

Nonsensical Variations?

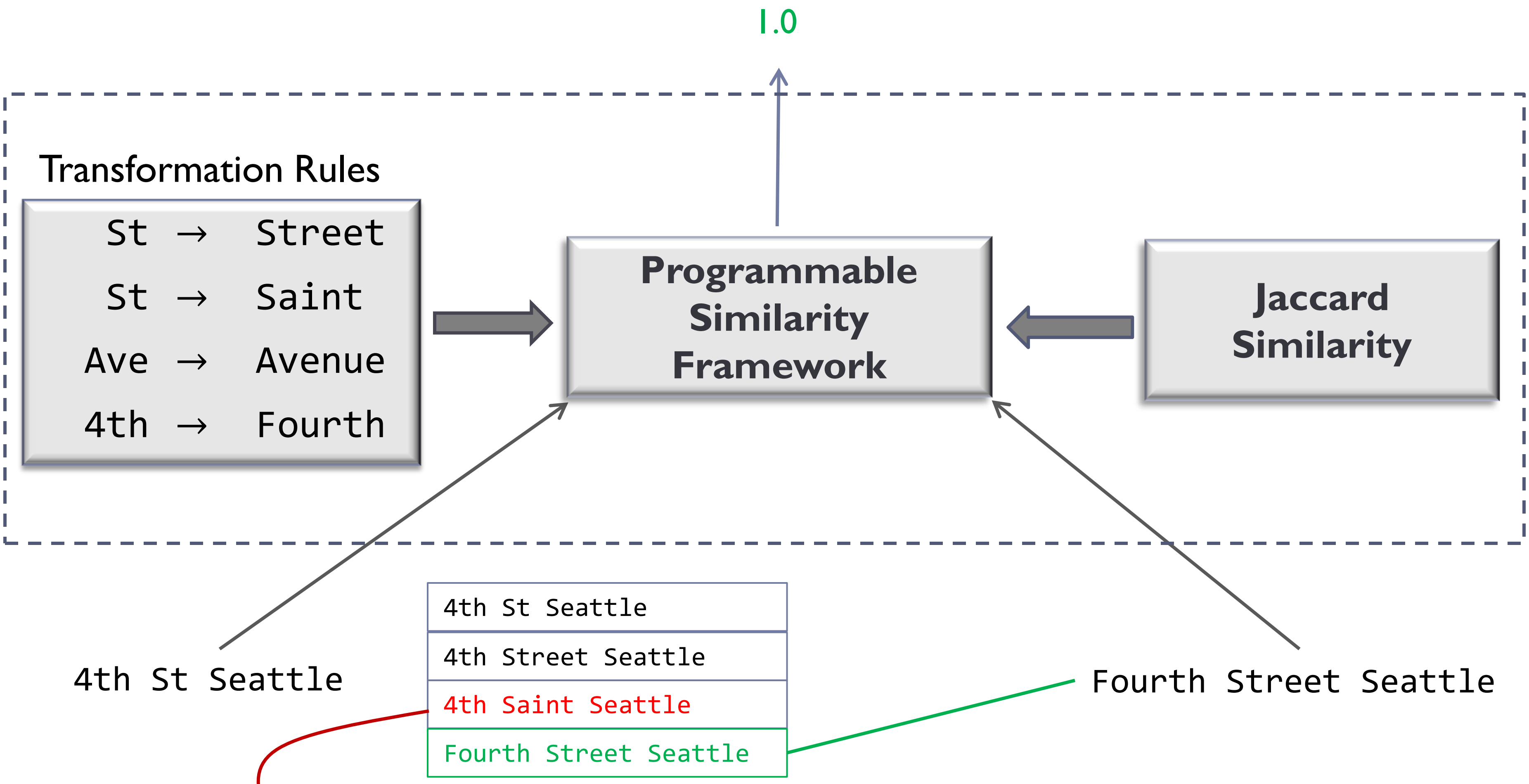


Transformation Rules

St	→	Street
St	→	Saint
Ave	→	Avenue
4th	→	Fourth

4th St Seattle
4th Street Seattle
4th Saint Seattle
Fourth Street Seattle

Nonsensical Variations?



1.0

Transformation Rules

St	→	Street
St	→	Saint
Ave	→	Avenue
4th	→	Fourth

Programmable Similarity Framework

Jaccard Similarity

4th St Seattle

4th St Seattle
4th Street Seattle
4th Saint Seattle
Fourth Street Seattle

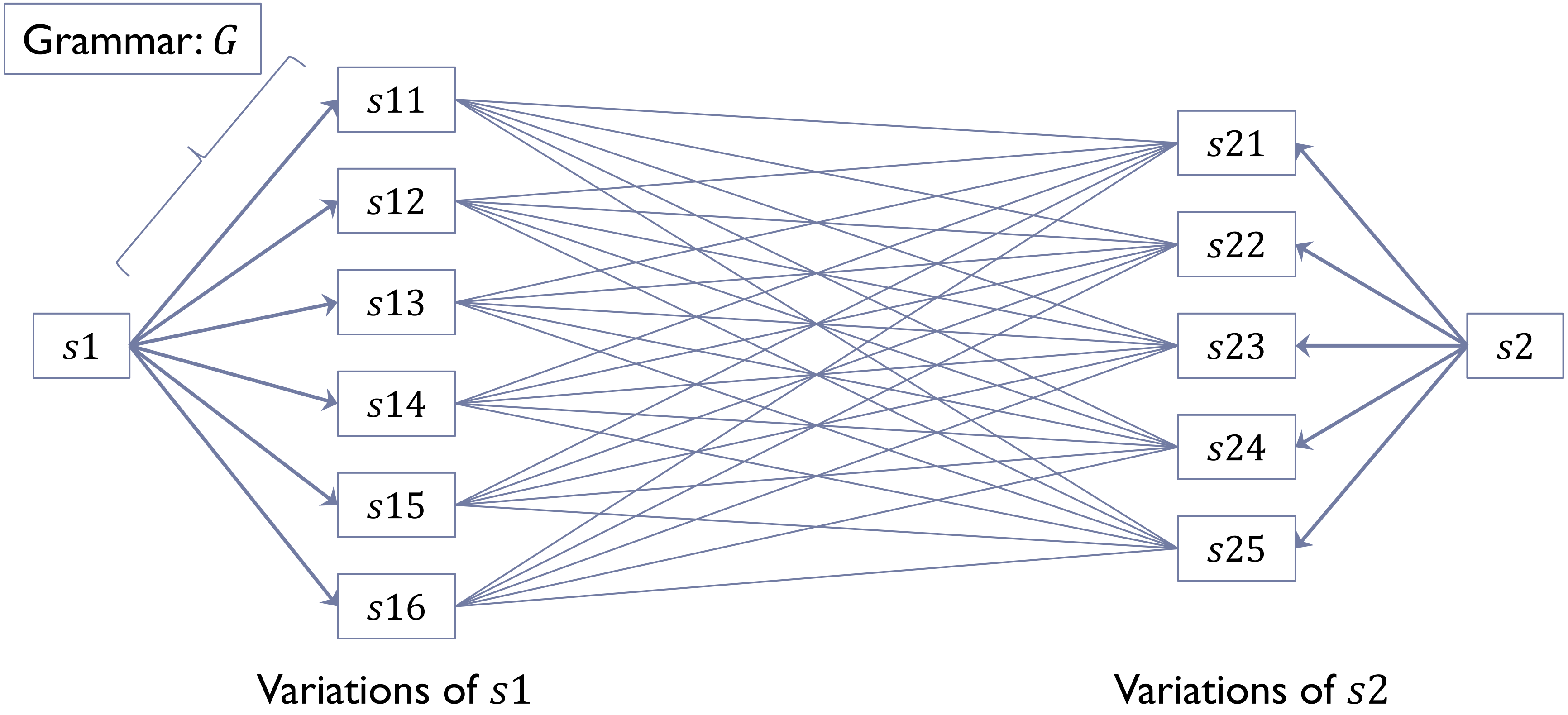
Fourth Street Seattle

Overall similarity not affected

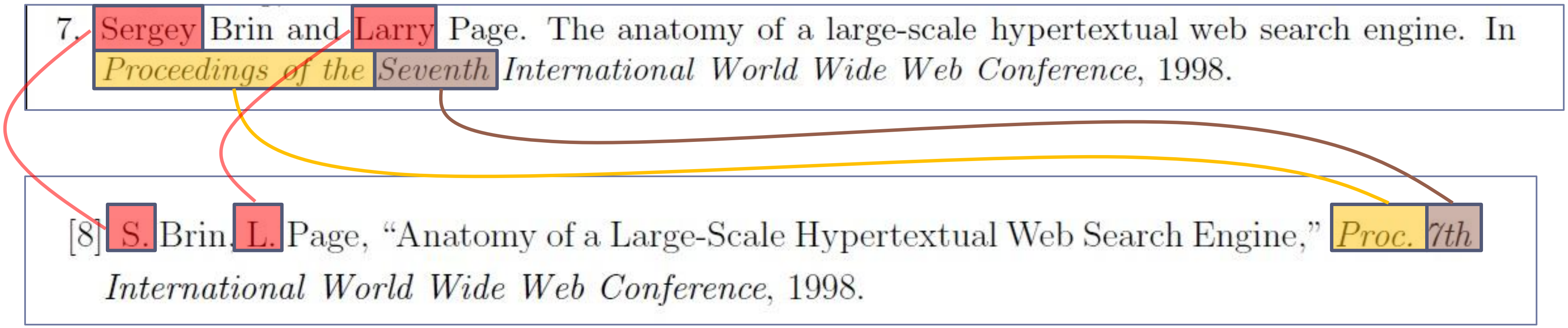
Programmable Similarity

Avg?

$ProgSim(s1, s2)$: ~~Max~~$_{j,k}$ $Jaccard(s1j, s2k)$



Similarity for Citations

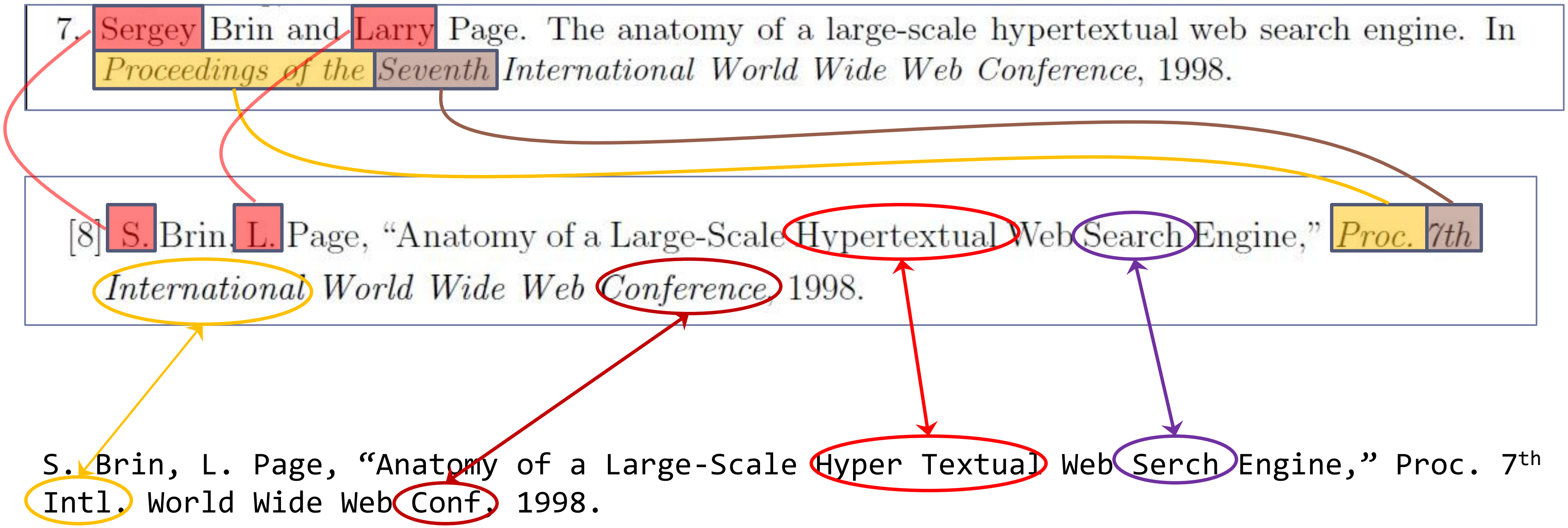


7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[8] S. Brin, L. Page, “Anatomy of a Large-Scale Hypertextual Web Search Engine,” *Proc. 7th International World Wide Web Conference*, 1998.

S	→	Sergey
L	→	Larry
7th	→	Seventh
Proc	→	Proceedings of the

Similarity for Citations



7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[8] S. Brin, L. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine," *Proc. 7th International World Wide Web Conference*, 1998.

S. Brin, L. Page, "Anatomy of a Large-Scale Hyper Textual Web Serch Engine," Proc. 7th Intl. World Wide Web Conf. 1998.

How to anticipate and enumerate all these variations?

Example Rule Class: Edits

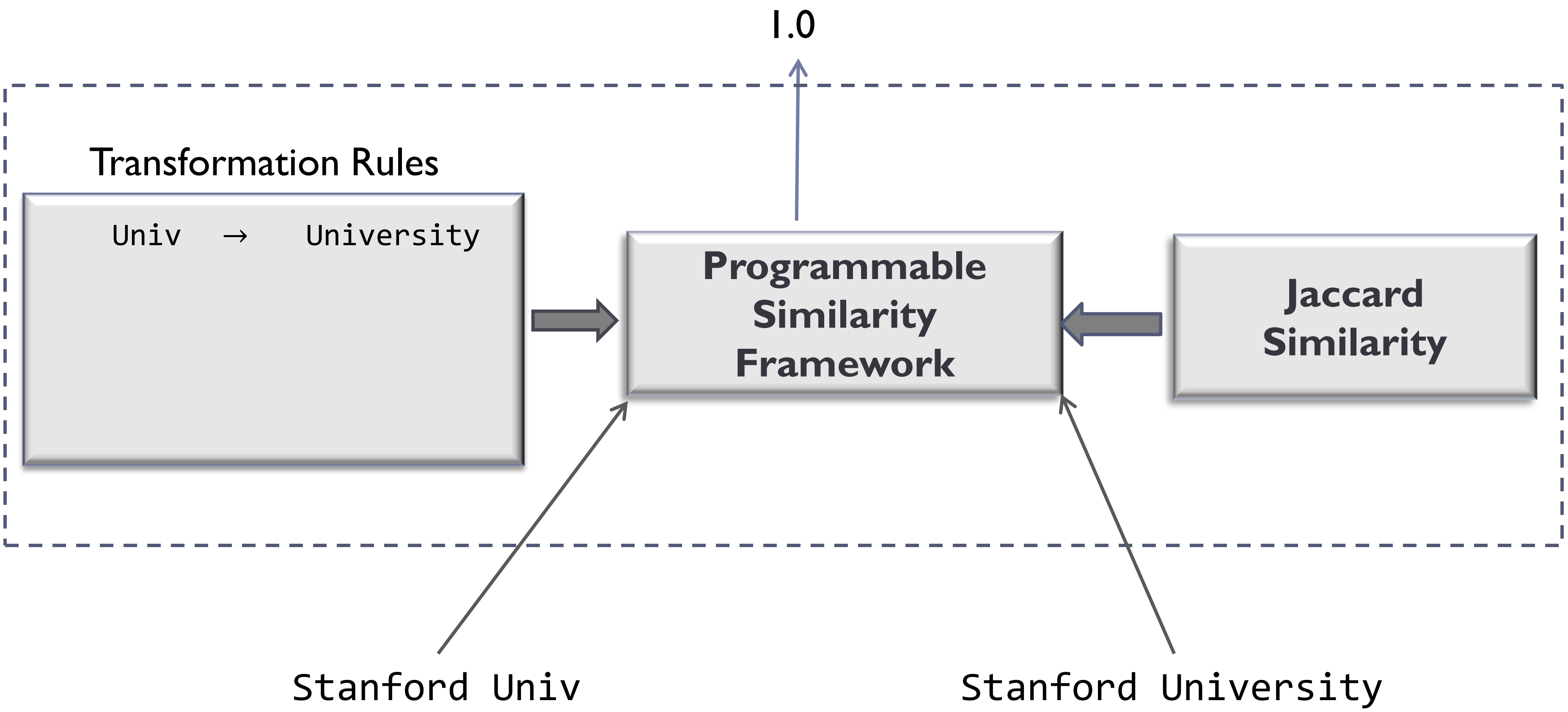
$$\{\, w' \rightarrow w \mid w \epsilon\ Dictionary\ of\ Words\ \ \wedge\ EditDistance(w', w) \leq k \}$$

Stanf0rd	→	Stanford
Stnford	→	Stanford
Stanfrd	→	Stanford
Berkley	→	Berkeley
…	…	…

Too Large?

Dictionary of Cities
Stanford
Berkeley
…

Dynamic Edit Rules


1.0

Transformation Rules

Univ → University

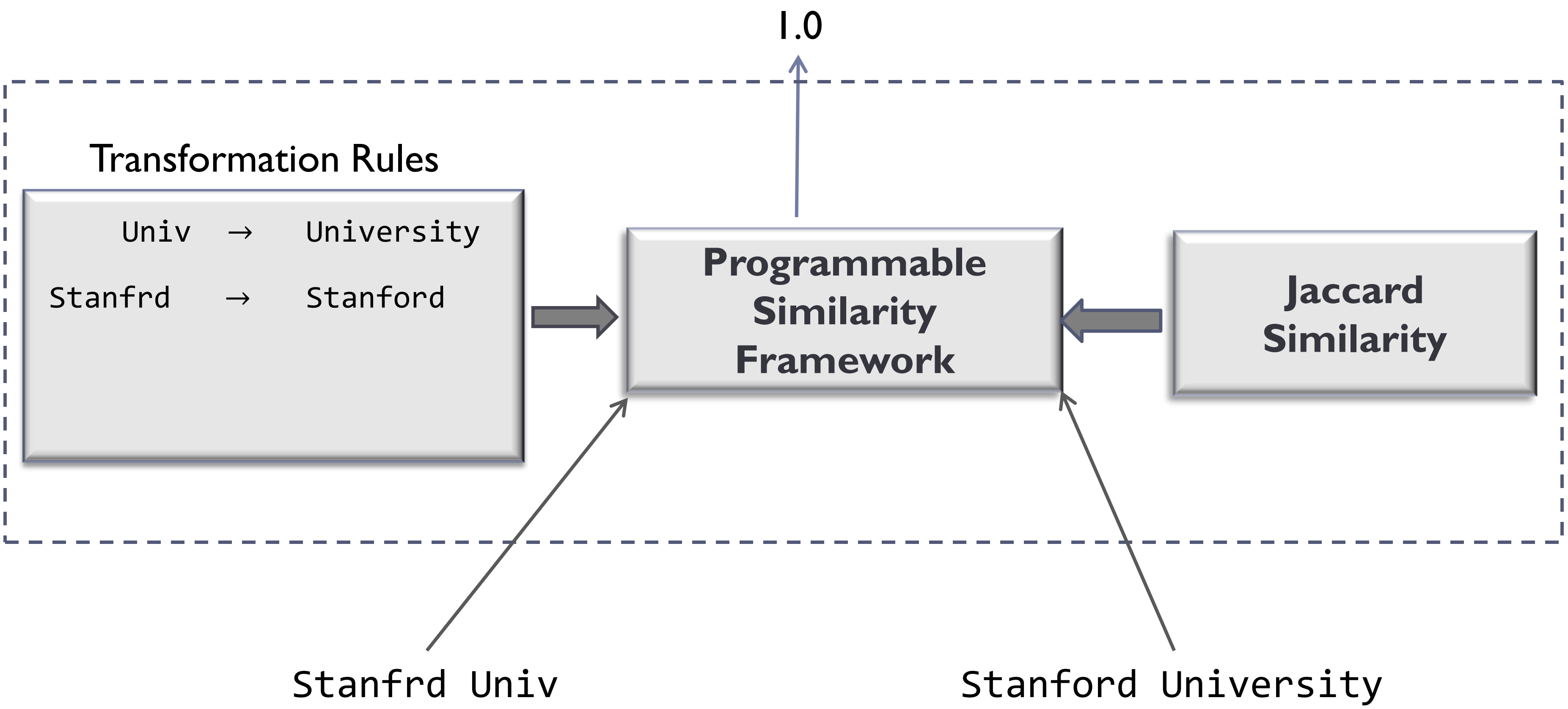
Programmable Similarity Framework

Jaccard Similarity

Stanford Univ

Stanford University

Dynamic Edit Rules



Example Rule Class: First Name Initials

$$\{\, l \rightarrow w \mid w \,\epsilon\, Dictionary\ of\ FirstNames \ \ \wedge\ w = lu\ \ \wedge\ l\ \in letters\}$$

H	→	Hector
J	→	Jeffrey
J	→	Jennifer
…	→	…
…	…	…

Dictionary of First Names
Hector
Jeffrey
Jennifer
…

Example Rule Class: Number related

1 → One

2 → Two

3 → Three

… … …

1st → First

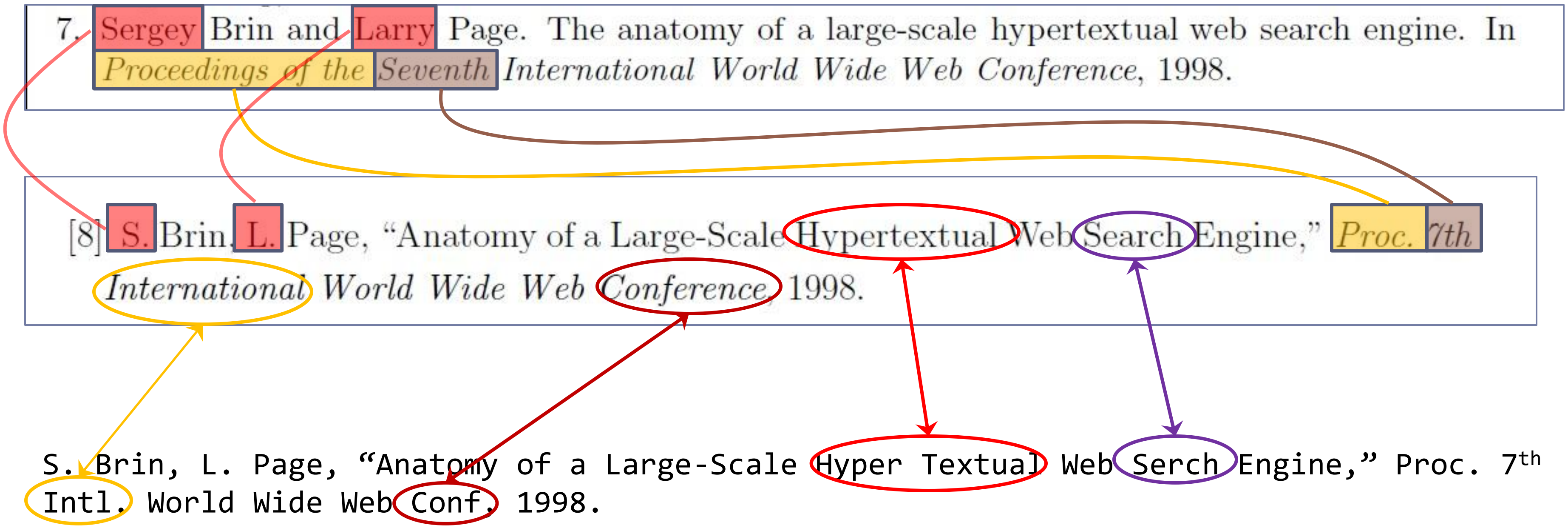
2nd → Second

3rd → Third

… … …

Easily generated programmatically

Similarity for Citations



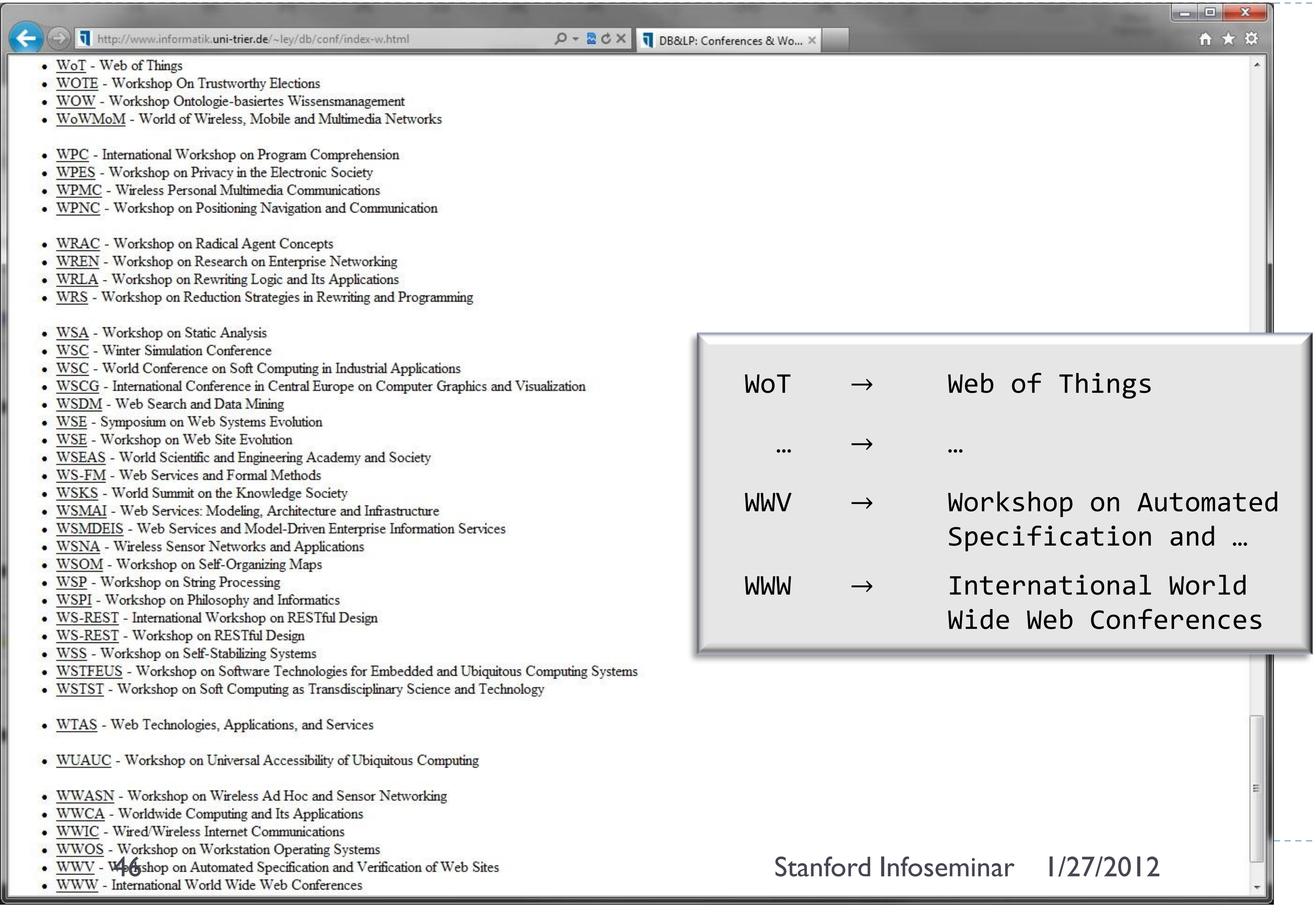
7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[8] S. Brin, L. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine," *Proc. 7th International World Wide Web Conference*, 1998.

S. Brin, L. Page, "Anatomy of a Large-Scale Hyper Textual Web Serch Engine," Proc. 7th Intl. World Wide Web Conf. 1998.

How to anticipate and enumerate all these variations?

Web as a source of rules


http://www.informatik.uni-trier.de/~ley/db/conf/index-w.html
DB&LP: Conferences & Wo...

- WoT - Web of Things
- WOTE - Workshop On Trustworthy Elections
- WOW - Workshop Ontologie-basiertes Wissensmanagement
- WoWMoM - World of Wireless, Mobile and Multimedia Networks

- WPC - International Workshop on Program Comprehension
- WPES - Workshop on Privacy in the Electronic Society
- WPMC - Wireless Personal Multimedia Communications
- WPNC - Workshop on Positioning Navigation and Communication

- WRAC - Workshop on Radical Agent Concepts
- WREN - Workshop on Research on Enterprise Networking
- WRLA - Workshop on Rewriting Logic and Its Applications
- WRS - Workshop on Reduction Strategies in Rewriting and Programming

- WSA - Workshop on Static Analysis
- WSC - Winter Simulation Conference
- WSC - World Conference on Soft Computing in Industrial Applications
- WSCG - International Conference in Central Europe on Computer Graphics and Visualization
- WSDM - Web Search and Data Mining
- WSE - Symposium on Web Systems Evolution
- WSE - Workshop on Web Site Evolution
- WSEAS - World Scientific and Engineering Academy and Society
- WS-FM - Web Services and Formal Methods
- WSKS - World Summit on the Knowledge Society
- WSMAI - Web Services: Modeling, Architecture and Infrastructure
- WSMDEIS - Web Services and Model-Driven Enterprise Information Services
- WSNA - Wireless Sensor Networks and Applications
- WSOM - Workshop on Self-Organizing Maps
- WSP - Workshop on String Processing
- WSPI - Workshop on Philosophy and Informatics
- WS-REST - International Workshop on RESTful Design
- WS-REST - Workshop on RESTful Design
- WSS - Workshop on Self-Stabilizing Systems
- WSTFEUS - Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems
- WSTST - Workshop on Soft Computing as Transdisciplinary Science and Technology

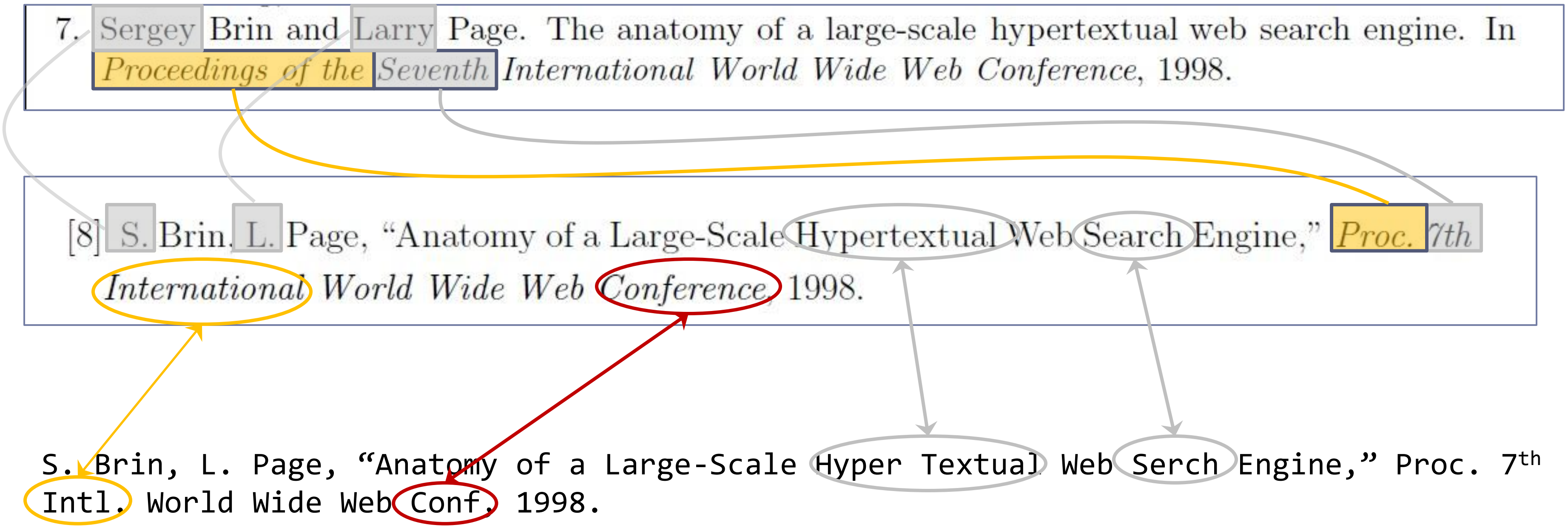
- WTAS - Web Technologies, Applications, and Services

- WUAUC - Workshop on Universal Accessibility of Ubiquitous Computing

- WWASN - Workshop on Wireless Ad Hoc and Sensor Networking
- WWCA - Worldwide Computing and Its Applications
- WWIC - Wired/Wireless Internet Communications
- WWOS - Workshop on Workstation Operating Systems
- WWV - Workshop on Automated Specification and Verification of Web Sites
- WWW - International World Wide Web Conferences

WoT	→	Web of Things
…	→	…
WWV	→	Workshop on Automated Specification and …
WWW	→	International World Wide Web Conferences

Similarity for Citations



7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[8] S. Brin, L. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine," *Proc. 7th International World Wide Web Conference*, 1998.

S. Brin, L. Page, "Anatomy of a Large-Scale Hyper Textual Web Serch Engine," Proc. 7th Intl. World Wide Web Conf. 1998.

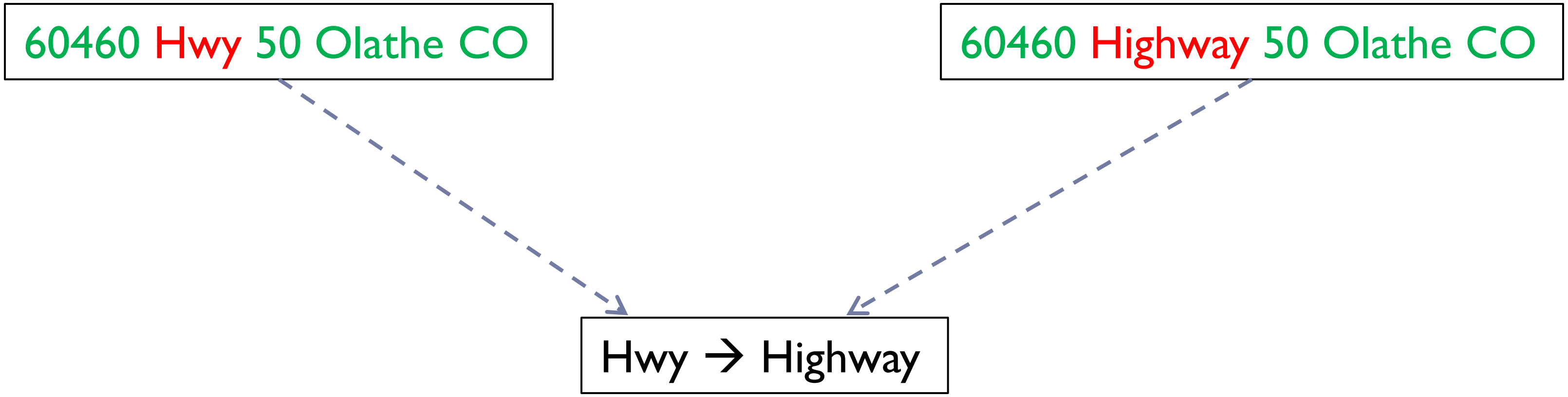
How to anticipate and enumerate all these variations?

Learning from Examples

60460 Hwy 50 Olathe CO

60460 Highway 50 Olathe CO

Learning from Examples



60460 Hwy 50 Olathe CO

60460 Highway 50 Olathe CO

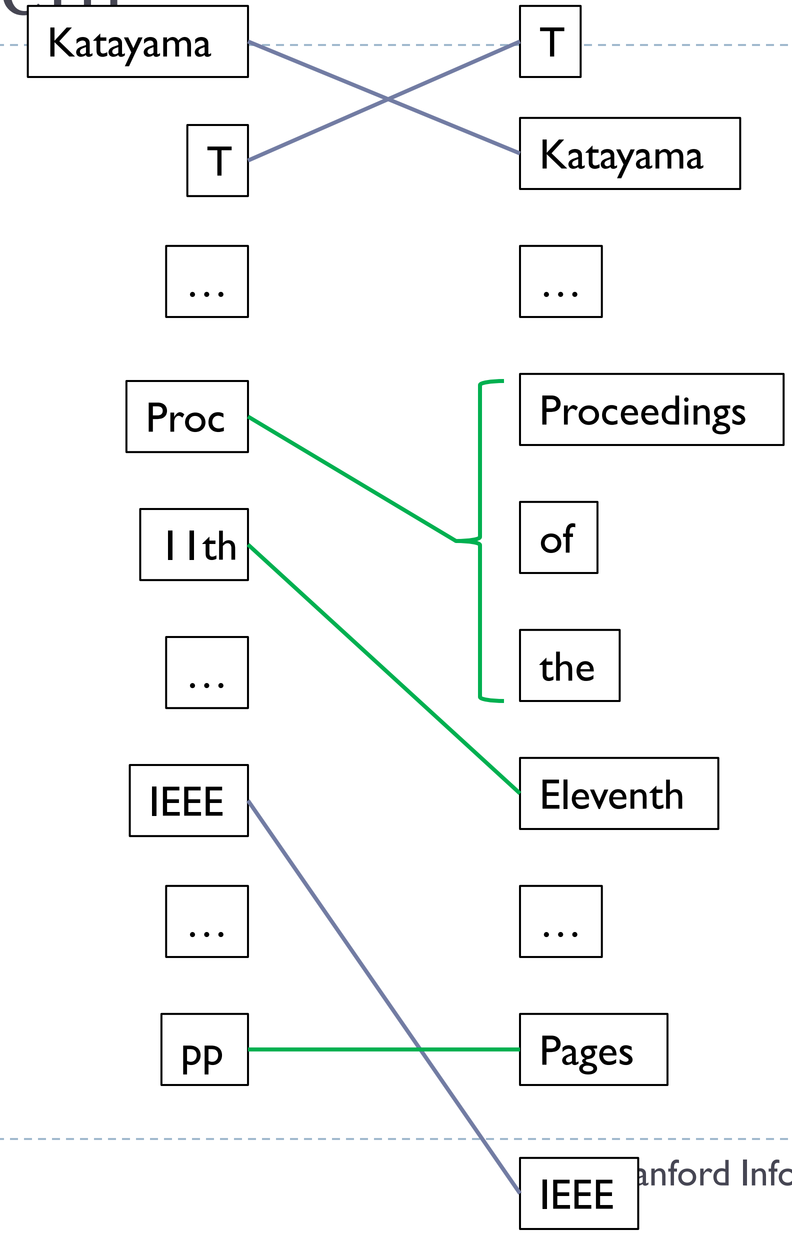
Hwy → Highway

Alignment

Katayama,T., “A hierarchical and functional software process description and its enaction”, Proc. 11th ICSE, IEEE, 1989, pp.343-352

T. Katayama, “A hierarchical and functional software process description and its enaction,” In: Proceedings of the Eleventh Int. Conf. On Soft. Eng. Pages: 343-352, IEEE Computer Society Press, Pittsburgh, PA, Jan 1989.

Alignment



Alignment:
A matching involving
(hyper-)edges

Problem Formulation

Output *k* transformations that maximize alignment of input matching strings.

Comments:

- As we increase *k* correct transformations start appearing before incorrect ones.
- There is a greedy $1/_2\ (1 - 1/_{e^2}) = 0.43$ approximation algorithm
- Connections to Machine Translation
  - Thanks: Dr. Fernando Pereira

Source of Rules (Summary)

- Manual
- Web
- Programmatic
- Learning

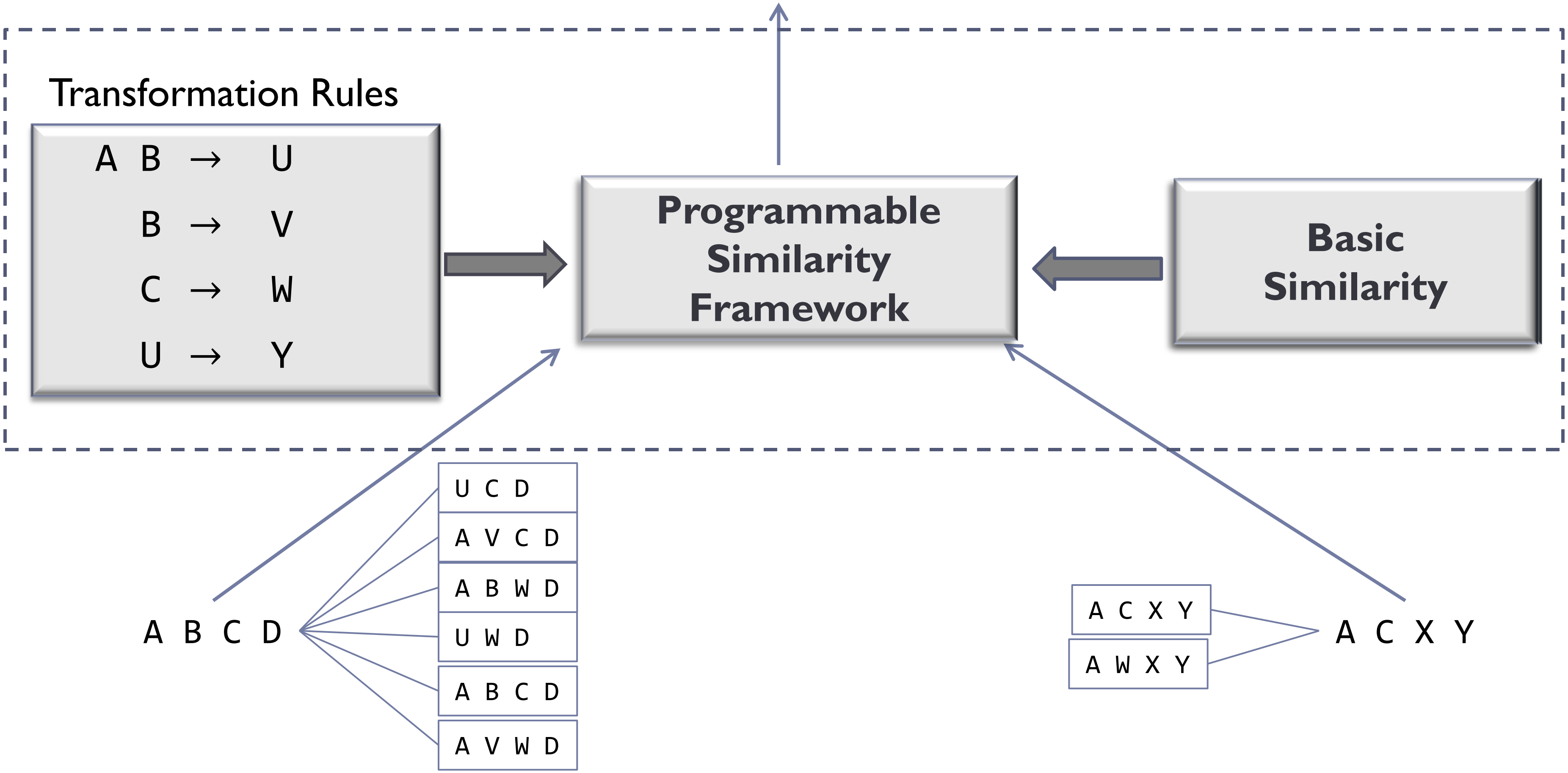
Overview

- Introduction
- Textual Similarity
  - Limitations of current similarity functions
- Programmable Similarity
  - Semantics
  - Usability
  - Performance
- Conclusion

Computing Similarity

- Compute similarity of string $s1$ and $s2$ under transformations R
- Undecidable in general ☹
- Engineering simplification
  - Only one "level" of derivation while applying transformations

Non-recursive derivation



Transformation Rules

A B → U

B → V

C → W

U → Y

Programmable Similarity Framework

Basic Similarity

A B C D

U C D

A V C D

A B W D

U W D

A B C D

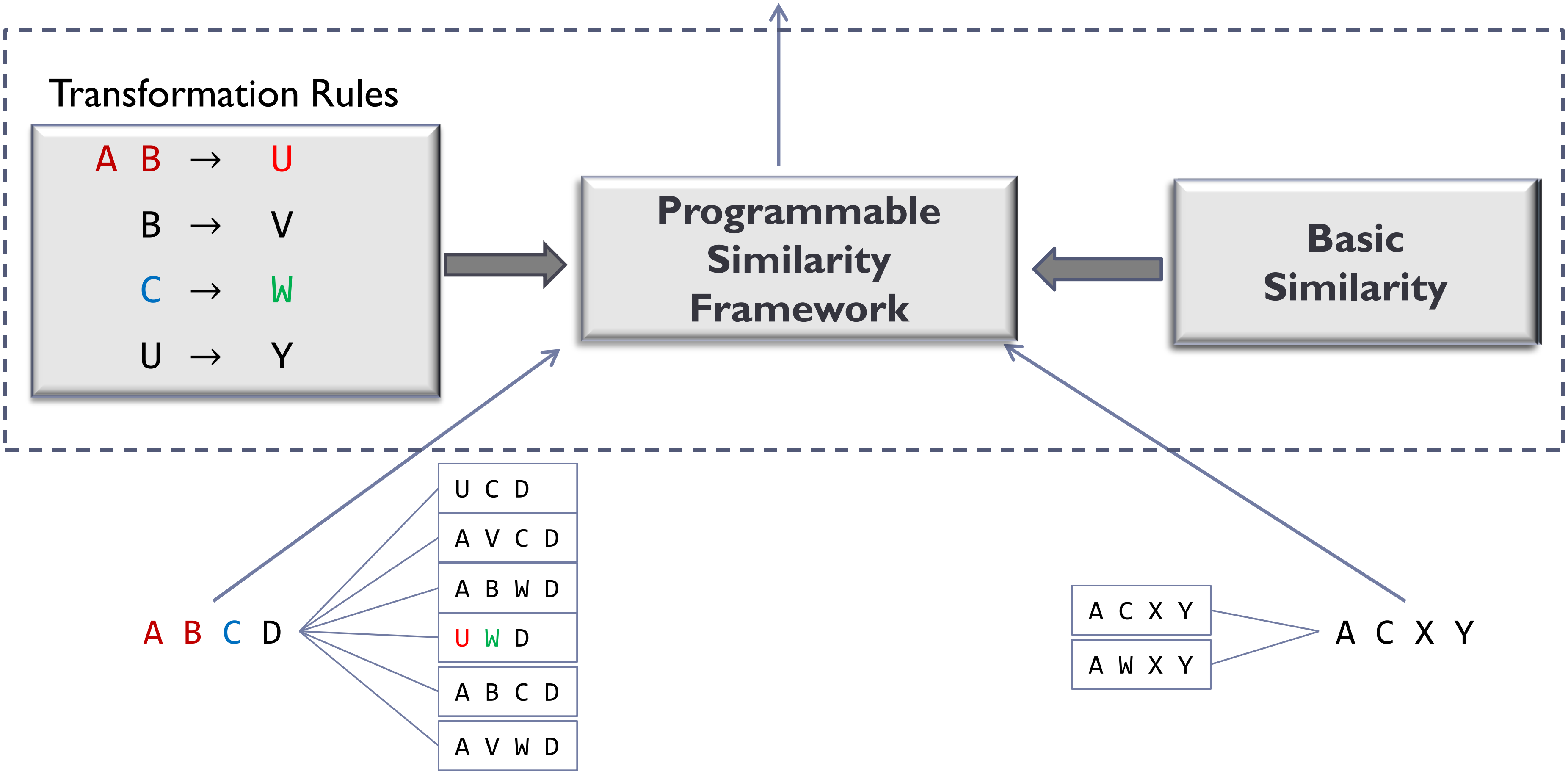
A V W D

A C X Y

A C X Y

A W X Y

Non-recursive derivation



Transformation Rules

A B → U

B → V

C → W

U → Y

Programmable Similarity Framework

Basic Similarity

A B C D

U C D

A V C D

A B W D

U W D

A B C D

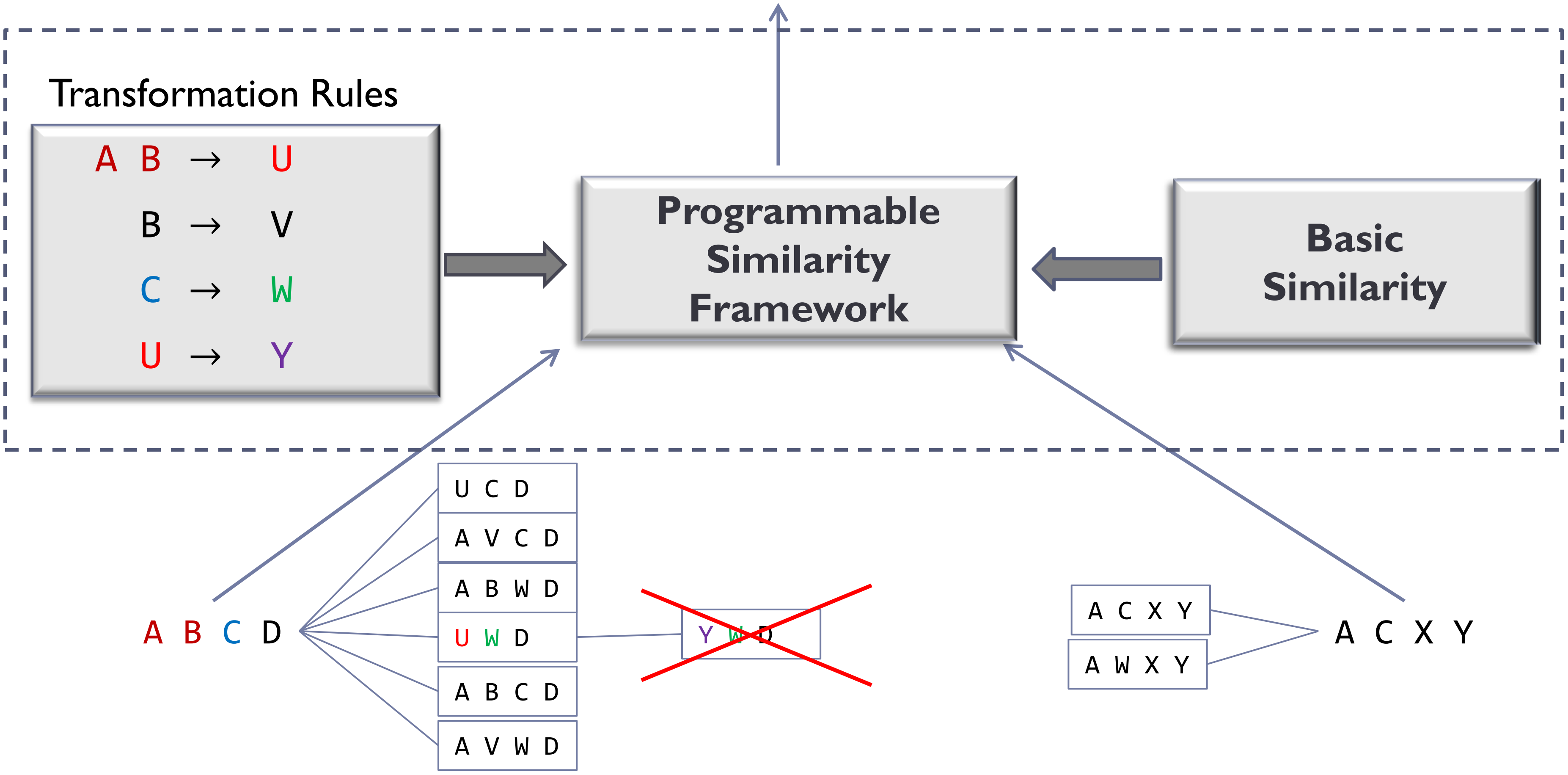
A V W D

A C X Y

A C X Y

A W X Y

Non-recursive derivation



Computing Similarity

- Compute similarity of string $s1$ and $s2$ under transformations R
- Undecidable in general ☹
- Engineering simplification
  - Only one "level" of recursion while applying transformations

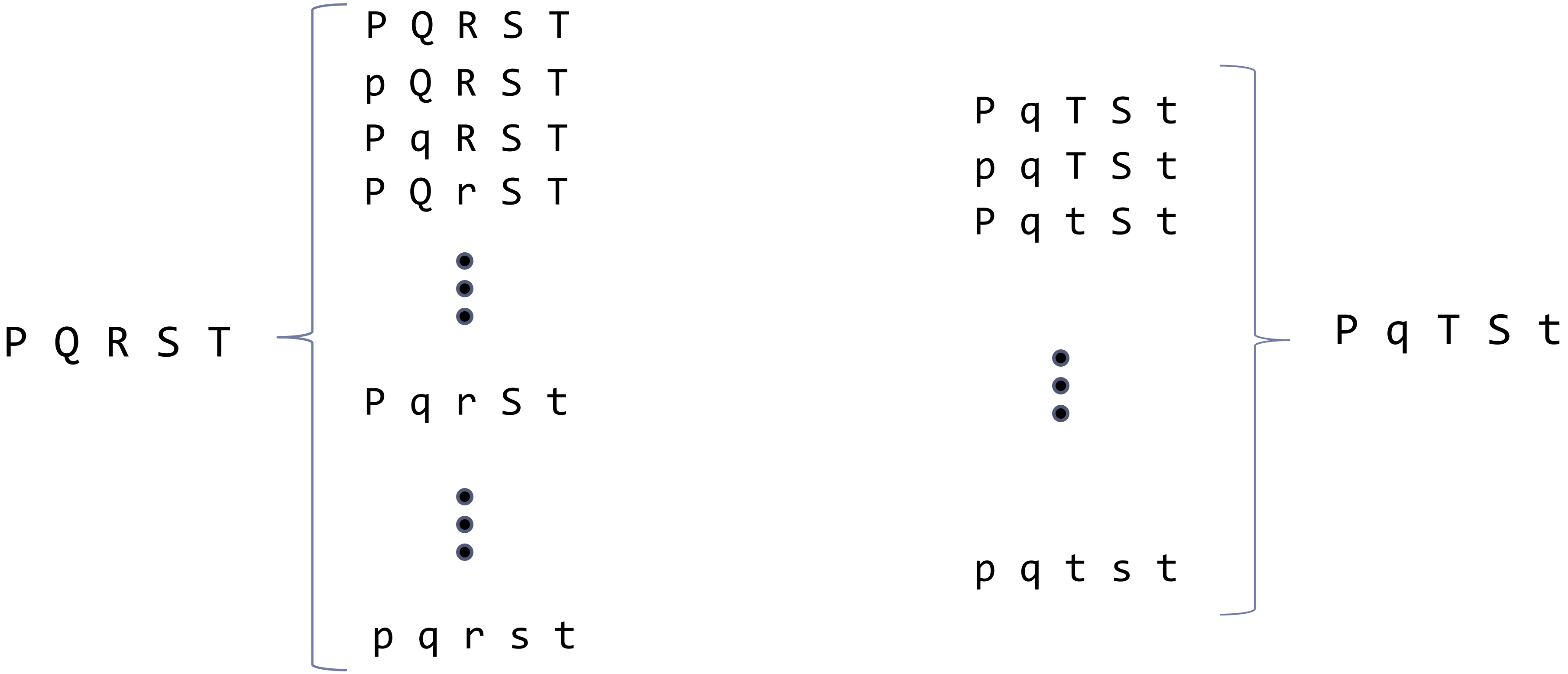
Running Example

Tokens/Words: `A, B, C, …, Z, a, b, …, z`

Transformation Rules:

A	→	a
B	→	b
C	→	c
…	…	…
Z	→	z

Computing Similarity



P Q R S T

P Q R S T

p Q R S T

P q R S T

P Q r S T

⋮

P q r S t

⋮

p q r s t

P q T S t

p q T S t

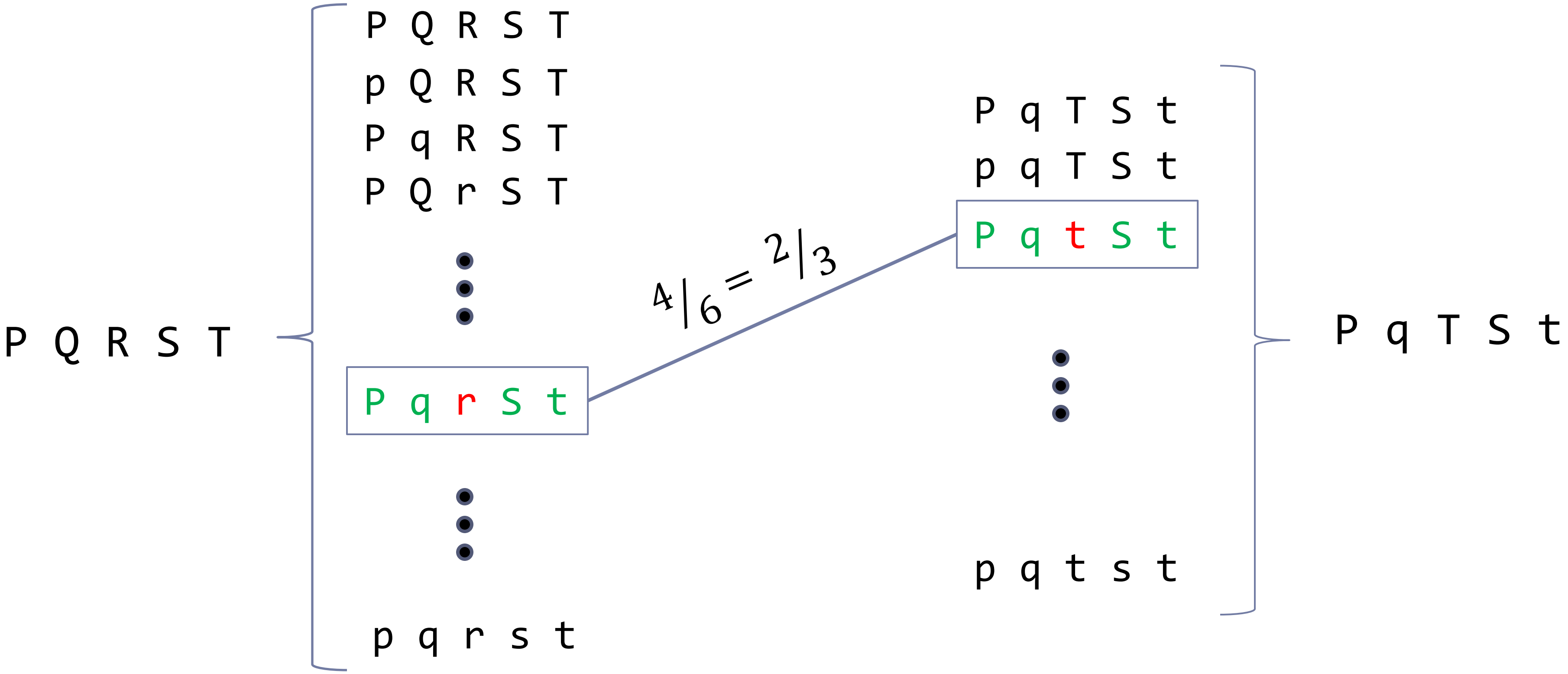
P q t S t

⋮

p q t s t

P q T S t

Computing Similarity



P Q R S T

P Q R S T

p Q R S T

P q R S T

P Q r S T

⋮

P q r S t

⋮

p q r s t

$4/6 = 2/3$

P q T S t

p q T S t

P q t S t

⋮

p q t s t

P q T S t

Computing Similarity

- NP-Hard in general
- Polynomial for *unit rules*
  - Reduce to maximum bipartite matching
  - Note: Works only for Jaccard variants

Unit rule: A → a

Multi rule: A B → a

Multi rule: A → a b

Reduction to Matching

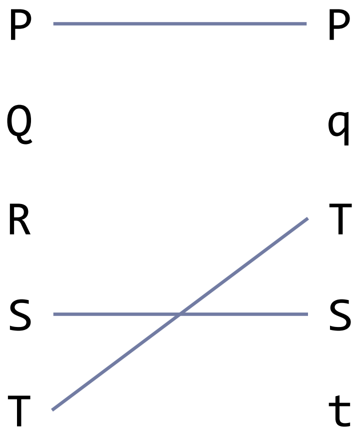
P Q R S T

P	P
Q	q
R	T
S	S
T	t

P q T S t

Reduction to Matching

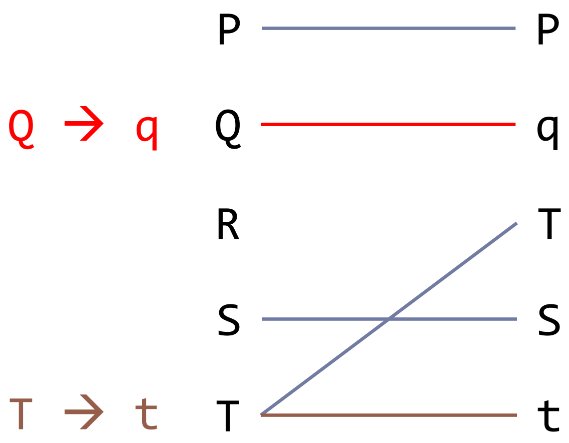
P Q R S T



P q T S t

Reduction to Matching

P Q R S T



Q → q

T → t

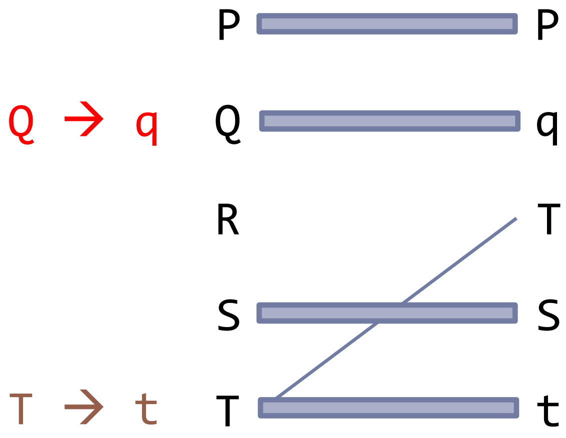
P Q R S T

P q T S t

P q T S t

Reduction to Matching

P Q R S T



P q T S t

Max Intersection = Max Matching = 4

Max Jaccard = Max Intersection / (10 – Max Intersection) = 4/6 = 2/3

Computing Similarity

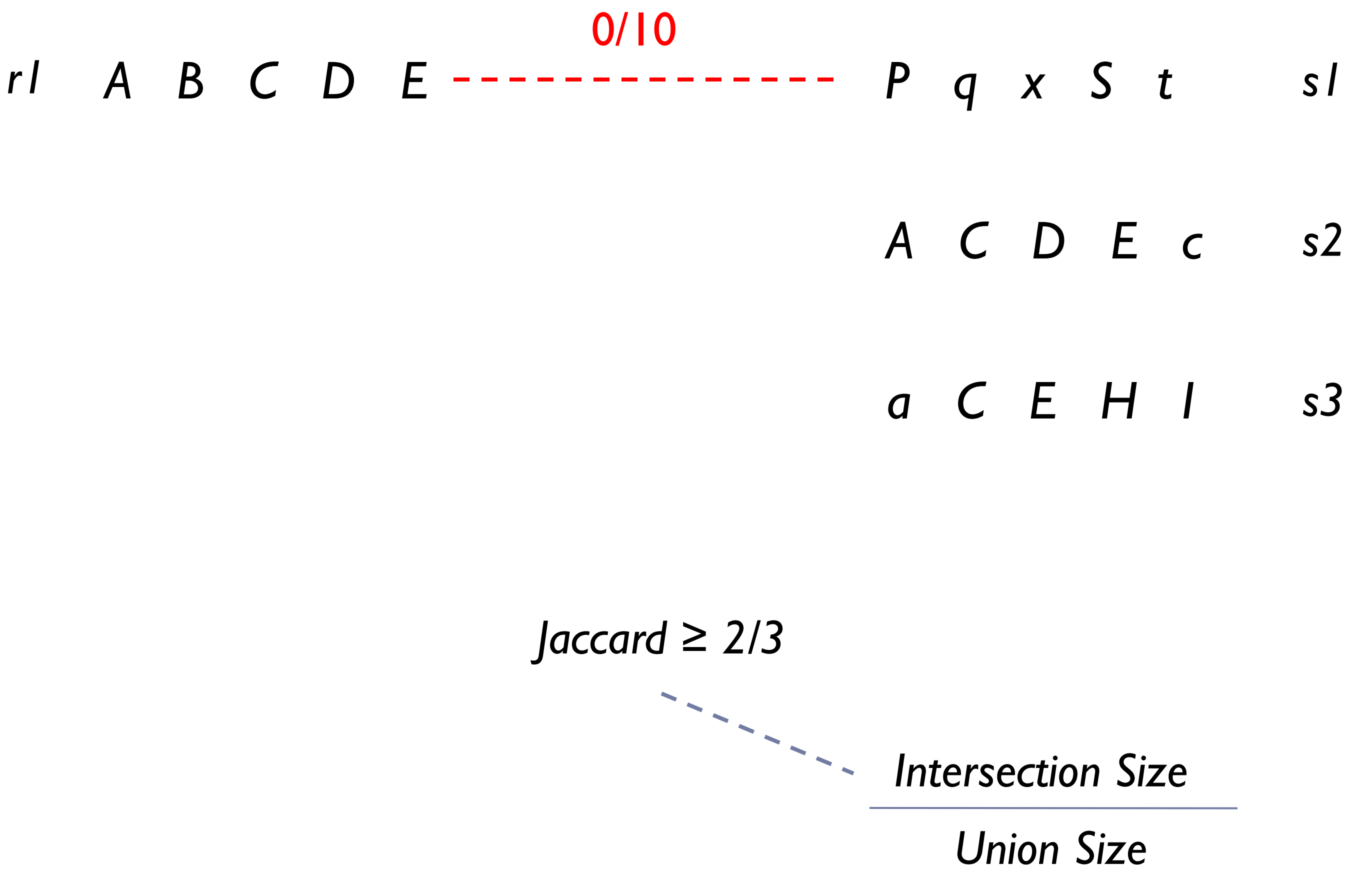
- NP-Hard in general
- Polynomial for *unit rules*
  - Reduce to maximum bipartite matching
- General Heuristic
  - Enumerate all variations due to multi-rules
  - Use polynomial algorithm for each pair of variations
  - Works well in practice
    - Unit rules more common
    - Multi rules produce fewer variations

Record Matching: Practical Considerations

- Index Setting:
  - Input: Relation S (to index) and a single record r
  - Output: All records of S with similarity $\geq \vartheta$ with r
- Join Setting (Similarity Join):
  - Input: Two relations R and S
  - Output: All pairs of records from R and S with similarity $\geq \vartheta$

Similarity in presence of transformations

Similarity Lookup (No Transformation)



0/10

r1 A B C D E *P q x S t s1*

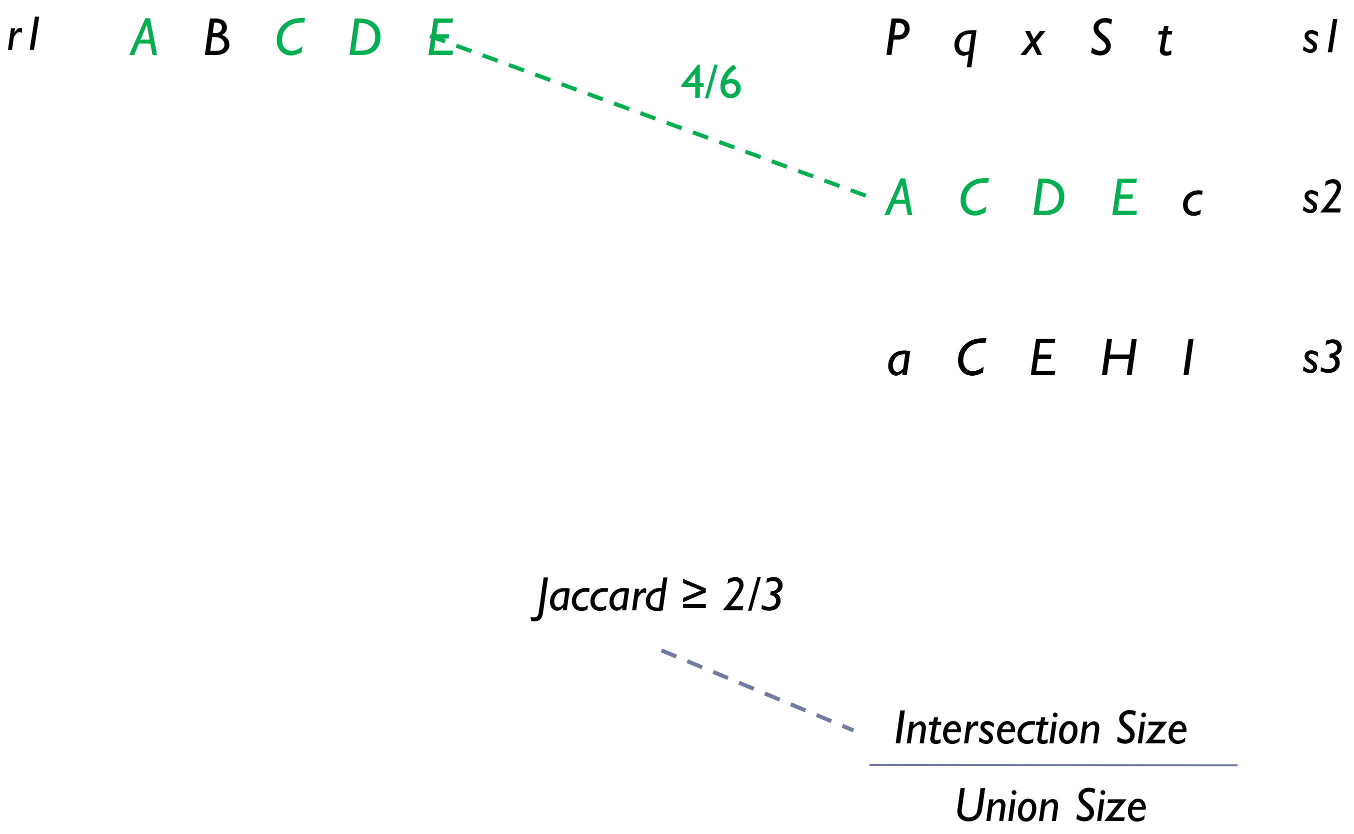
A C D E c s2

a C E H I s3

Jaccard ≥ *2/3*

$$\frac{\textit{Intersection Size}}{\textit{Union Size}}$$

Similarity Lookup (No Transformation)



r1 A B C D E

4/6

P q x S t s1

A C D E c s2

a C E H I s3

Jaccard ≥ 2/3

$$\frac{\text{Intersection Size}}{\text{Union Size}}$$

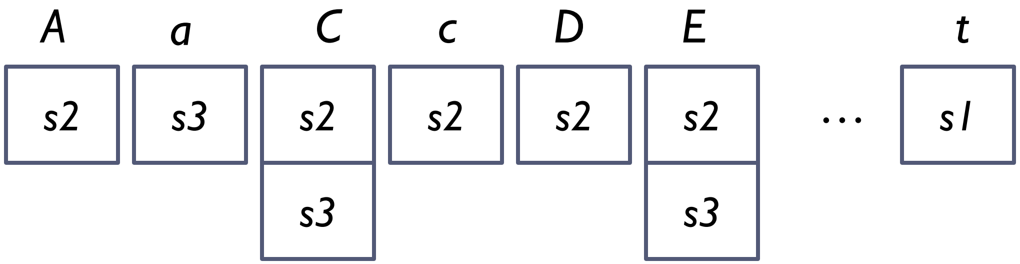
Similarity Lookup (No Transformation)

r1   *A B C D E*

P q x S t   *s1*

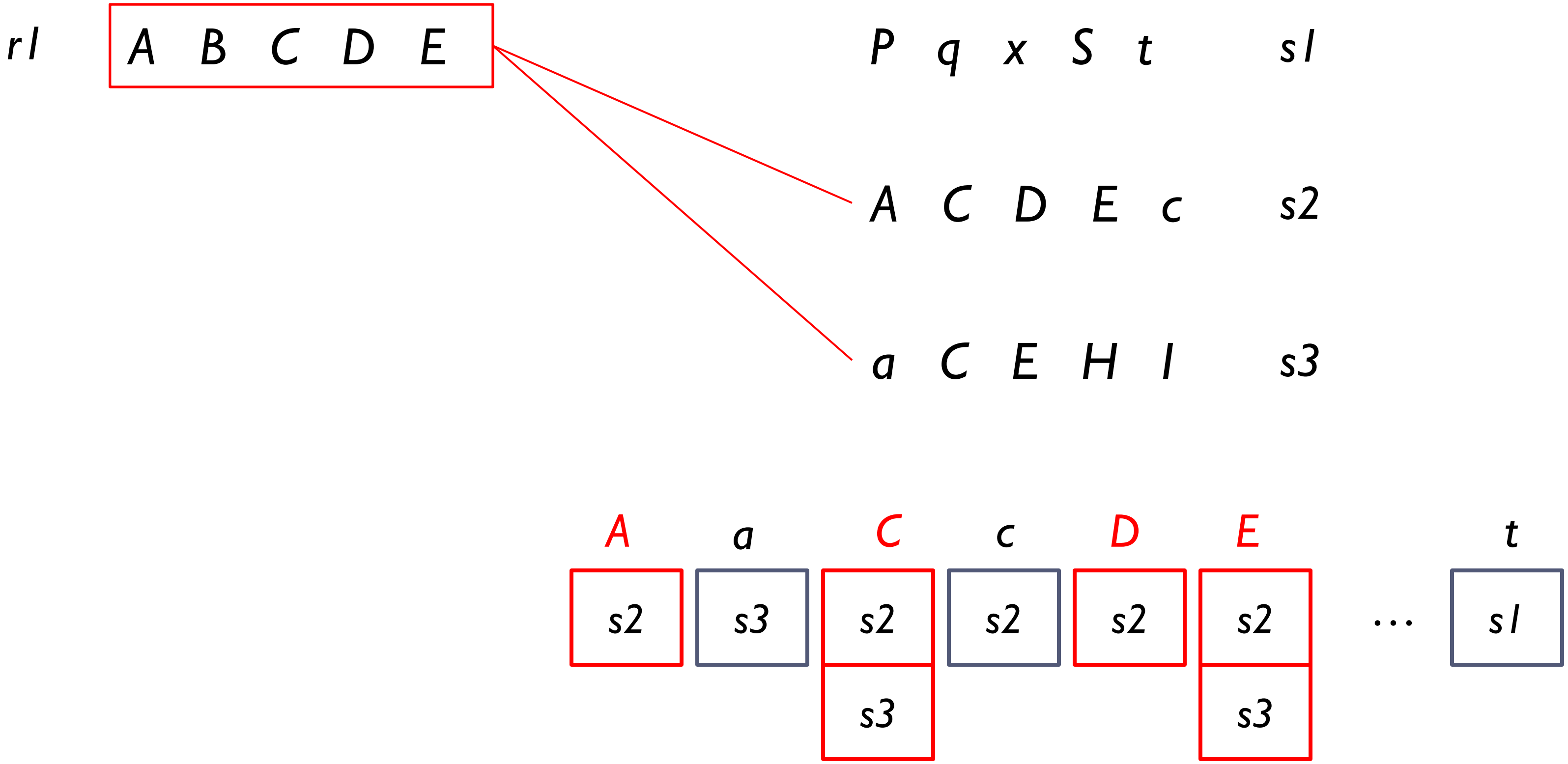
A C D E c   *s2*

a C E H I   *s3*



A	*a*	*C*	*c*	*D*	*E*		*t*
s2	*s3*	*s2*	*s2*	*s2*	*s2*	…	*s1*
		s3			*s3*		

Similarity Lookup (No Transformation)



r1 A B C D E

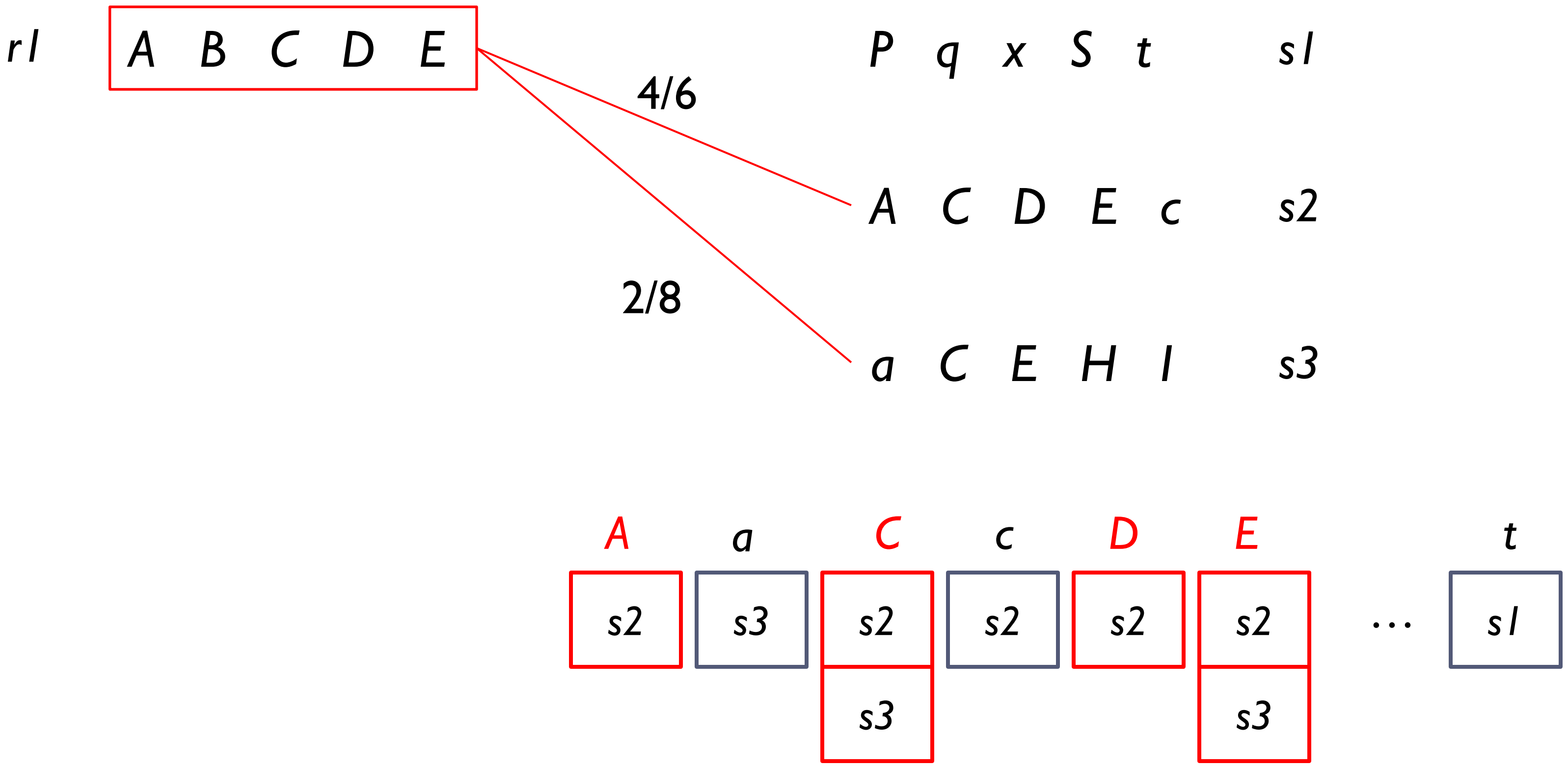
P q x S t s1

A C D E c s2

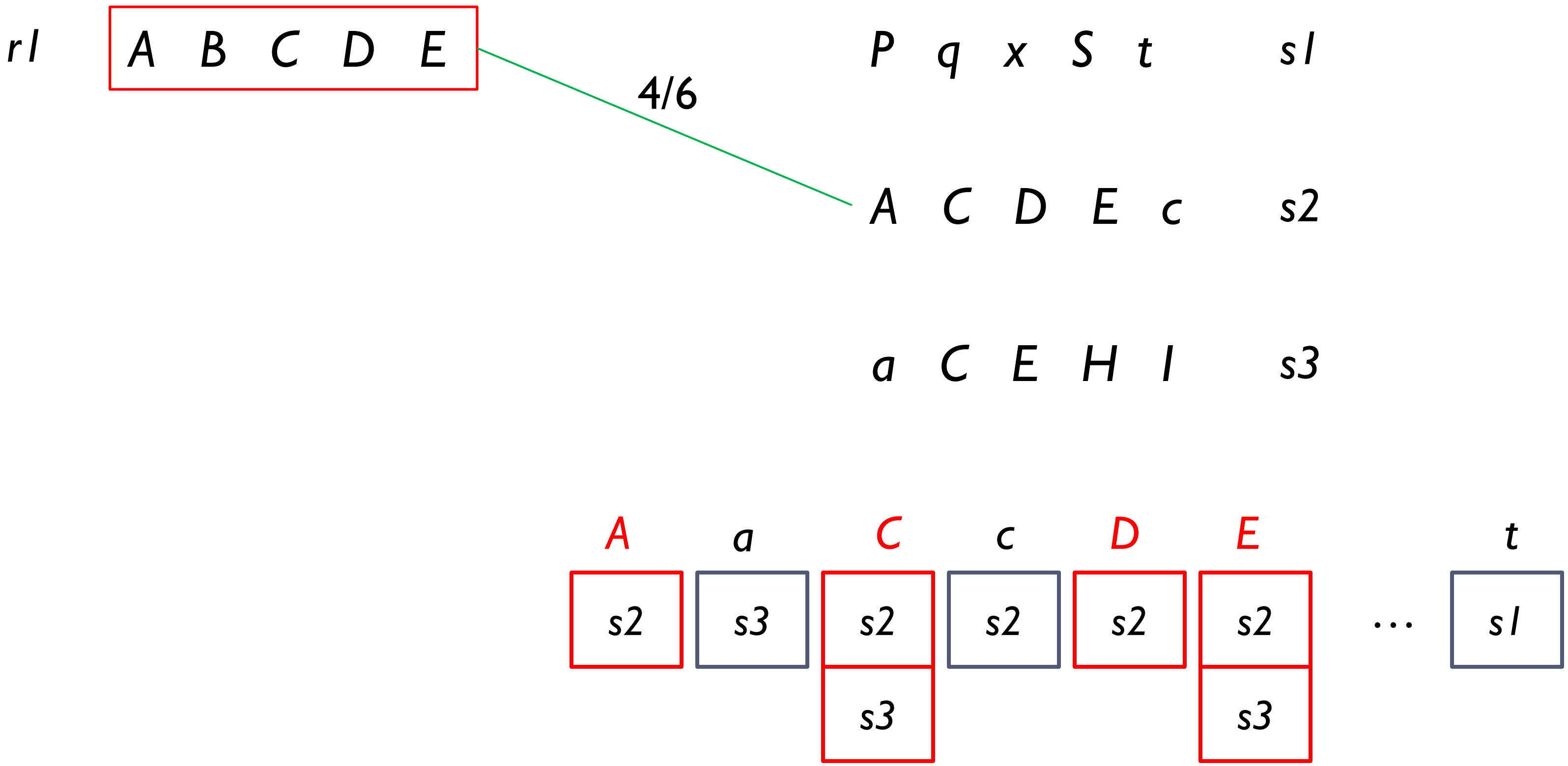
a C E H I s3

A	a	C	c	D	E	…	t
s2	s3	s2 s3	s2	s2	s2 s3		s1

Similarity Lookup (No Transformation)



Similarity Join (No Transformation)



r1 *A B C D E*

4/6

P q x S t *s1*

A C D E c *s2*

a C E H I *s3*

A	*a*	*C*	*c*	*D*	*E*		*t*
s2	*s3*	*s2*	*s2*	*s2*	*s2*	…	*s1*
		s3			*s3*		

Running Example

Tokens/Words: A, B, C, …, Z, a, b, …, z

Transformation Rules:

A	→	a
B	→	b
C	→	c
…	…	…
Z	→	z

Similarity Lookup

32

r11	A	B	C	D	E
r12	a	B	C	D	E
r13	A	b	C	D	E

…

…

P	q	x	S	t	s11
p	q	x	S	t	s12
P	q	x	s	t	s13

…

A	C	D	E	c	s21
a	C	D	E	c	s22
A	c	D	E	c	s23

…

a	C	E	H	I	s31
a	c	E	H	I	s32
a	C	e	H	I	s33

…

Inverted Index (Naïve)

Cost of matching *r1*: 32 x 5 = 160 index lookups

A	*a*	*C*	*c*	*D*	*E*	e
s21	*s22*	*s21*	*s21*	*s21*	*s21*	*s25*
s23	*s26*	*s22*	*s22*	*s22*	*s22*	*s29*
s24	…	*s23*	*s23*	*s23*	*s23*	…
…	*s31*	…	…	…	…	*s33*
	s32	*s31*	*s32*		*s31*	…
	s33	*s33*	*s36*		*s32*	
	…	…	…		…	

Inverted Index (Compressed)

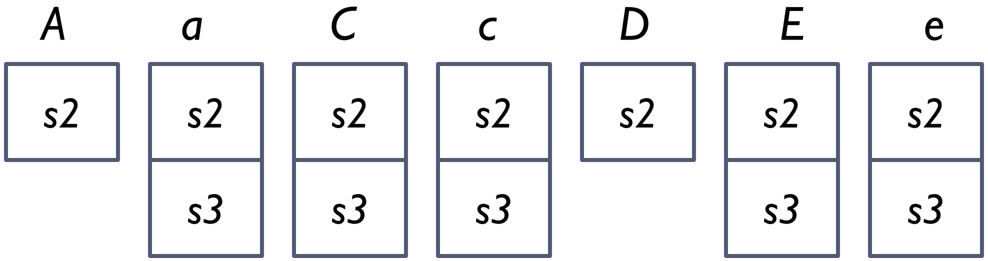
A	a	C	c	D	E	e
s21	s22	s21	s21	s21	s21	s25
s23	s26	s22	s22	s22	s22	s29
s24	…	s23	s23	s23	s23	…
…	s31	…	…	…	…	s33
	s32	s31	s32		s31	…
	s33	s33	s36		s32	
	…	…	…		…	

Inverted Index (Compressed)

A	*a*	*C*	*c*	*D*	*E*	*e*
s2	*s22*	*s21*	*s21*	*s21*	*s21*	*s25*
	s26	*s22*	*s22*	*s22*	*s22*	*s29*
	…	*s23*	*s23*	*s23*	*s23*	…
	s31	…	…	…	…	*s33*
	s32	*s31*	*s32*		*s31*	…
	s33	*s33*	*s36*		*s32*	
	…	…	…		…	

Inverted Index (Compressed)

r1 A B C D E



A: s2

a: s2, s3

C: s2, s3

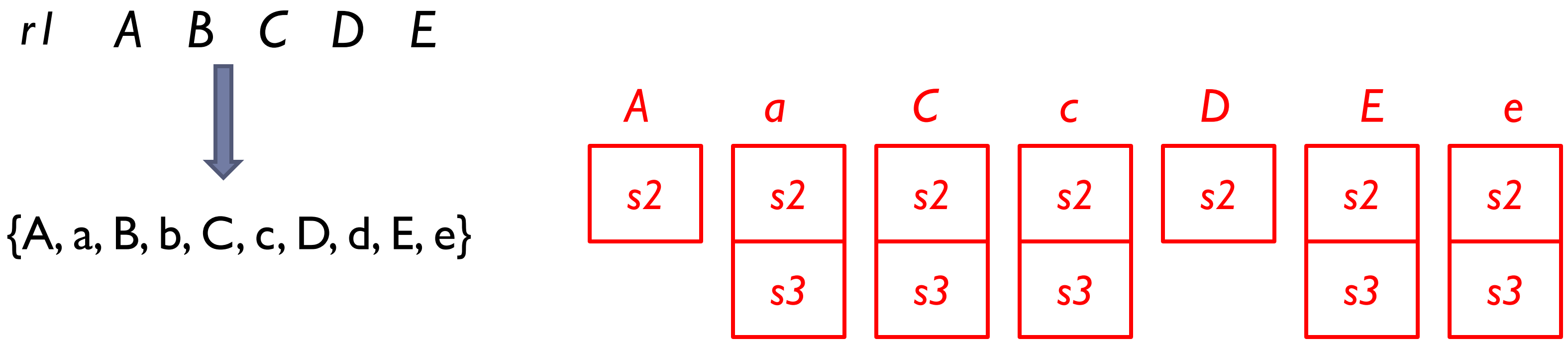
c: s2, s3

D: s2

E: s2, s3

e: s2, s3

Inverted Index (Compressed)



r1 *A* *B* *C* *D* *E*

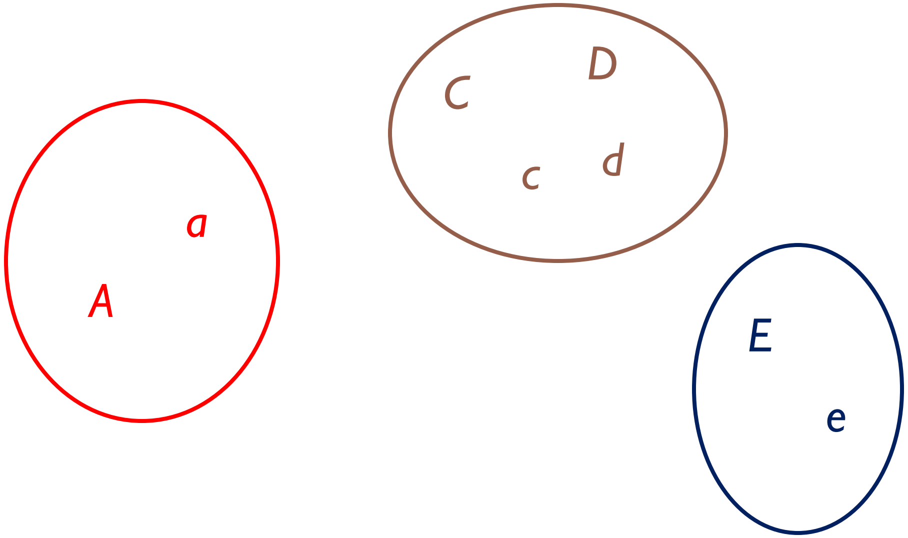
{A, a, B, b, C, c, D, d, E, e}

A	*a*	*C*	*c*	*D*	*E*	*e*
s2	s2	s2	s2	s2	s2	s2
	s3	s3	s3		s3	s3

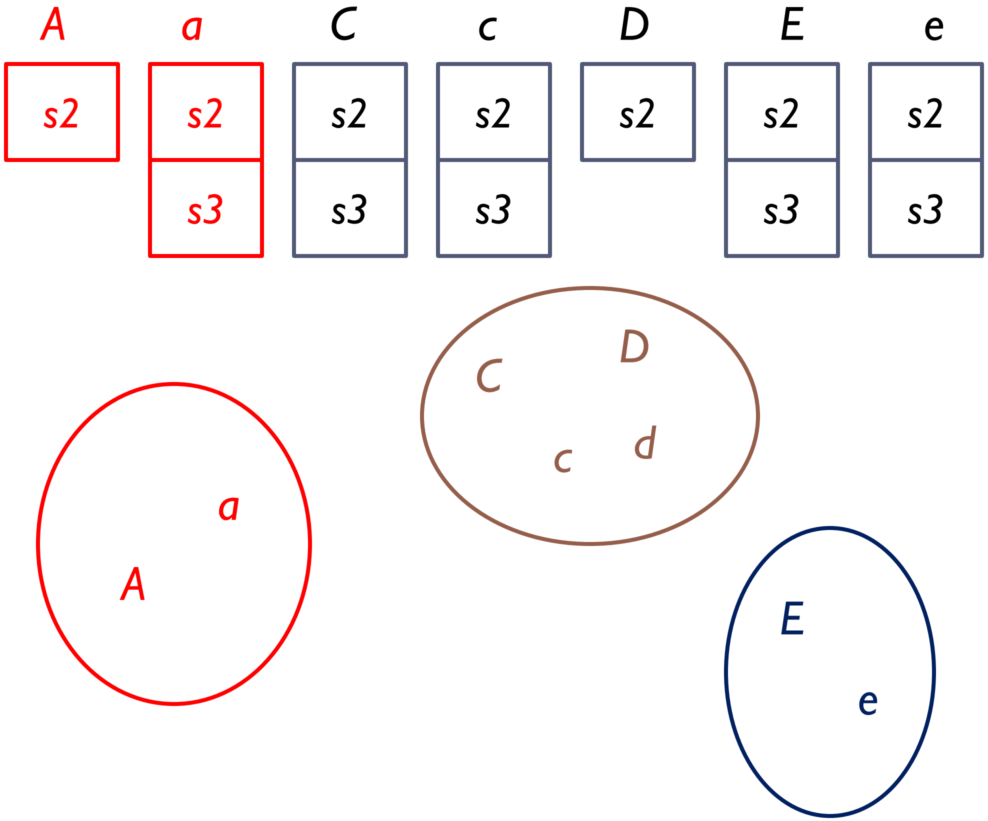
Cost of matching *r1*: 10 index lookups + 2 similarity computations

Token Clustering

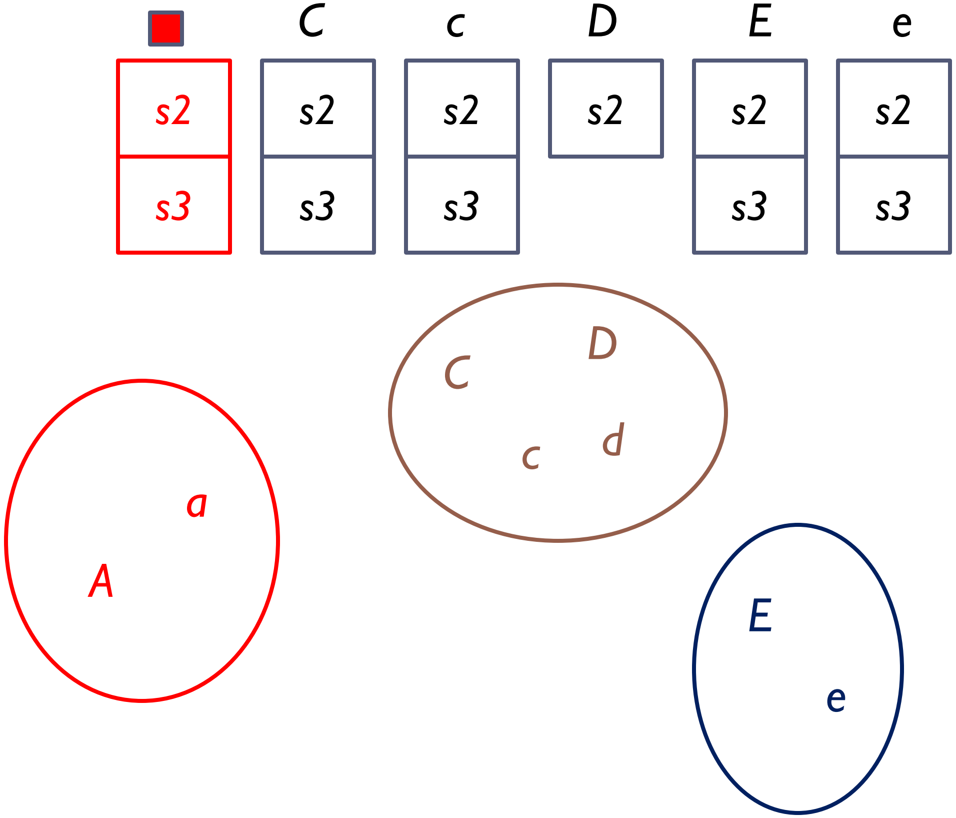
A	a	C	c	D	E	e
s2	s2	s2	s2	s2	s2	s2
	s3	s3	s3		s3	s3



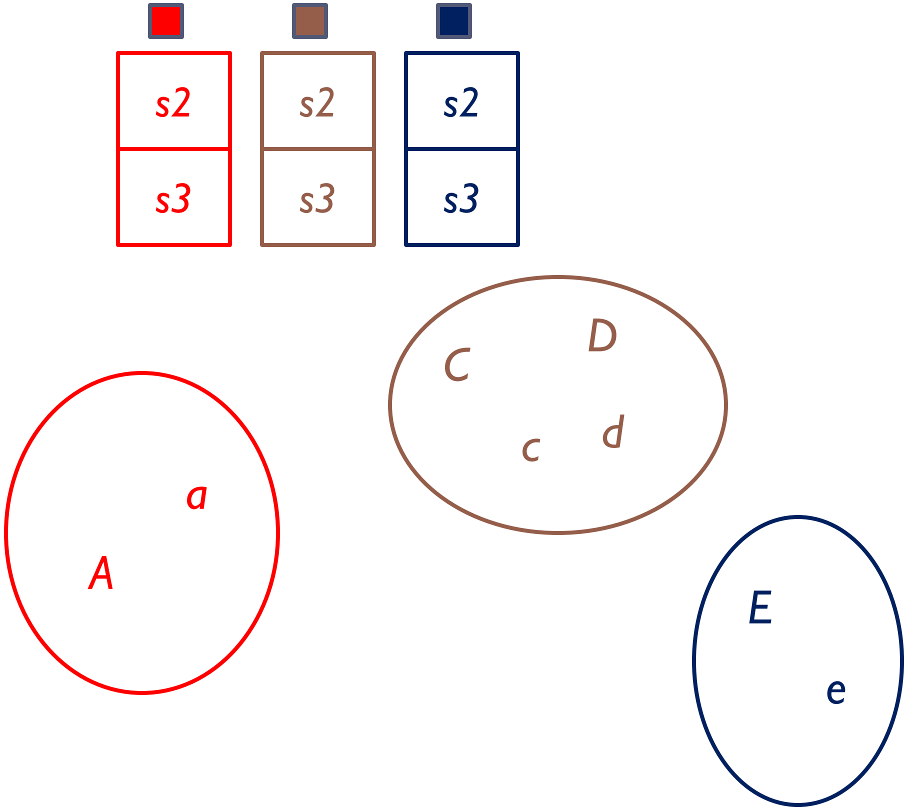
Token Clustering



Token Clustering

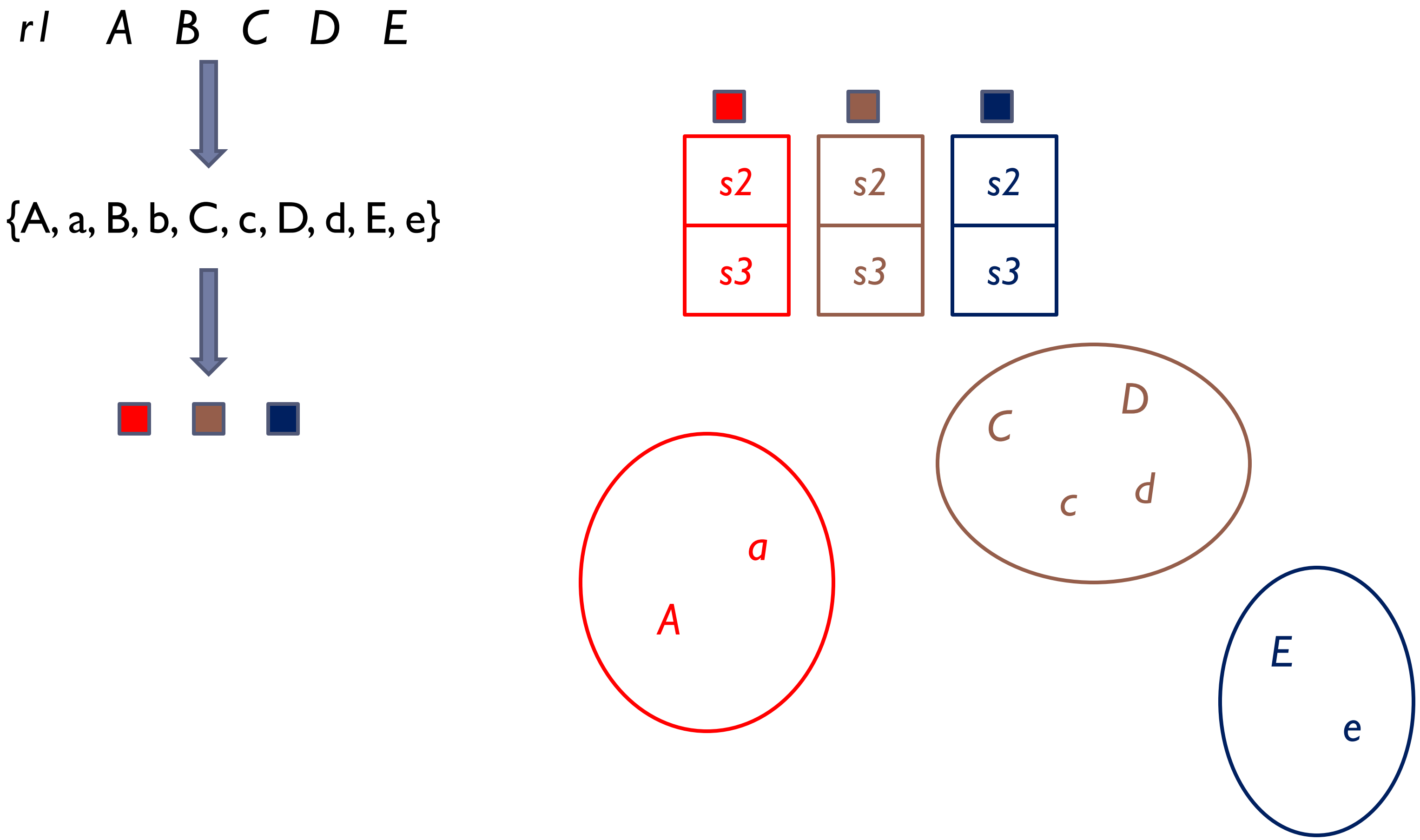


Token Clustering



Token Clustering

Cost of matching *r1*: 3 index lookups + 2 similarity computations



Representative Performance

- Bing Maps data:
  - 10M addresses
  - > 24M transformations (mostly programmatic – edit, abbreviations)
  - Average lookup time ~3ms

Conclusion

- Programmable similarity for record matching
- Advantages:
  - Customizability
  - Single similarity function
    - Software engineering advantages
  - Efficient Indexing

References & Acknowledgments

- Arvind Arasu, Venkatesh Ganti, Raghav Kaushik: Efficient Exact Set-Similarity Joins. VLDB 2006: 918-929
- Arvind Arasu, Surajit Chaudhuri, Raghav Kaushik: Transformation-based Framework for Record Matching. ICDE 2008: 40-49
- Arvind Arasu, Surajit Chaudhuri, Raghav Kaushik: Learning String Transformations From Examples. PVLDB 2(1): 514-525 (2009)

Acknowledgments:

Surajit Chaudhuri, Zhimin Chen, Kris Ganjam, Venky Ganti, Raghav Kaushik, Vivek Narasayya, Dong Xin