

Match Graph Generation for Symbolic Indirect Correlation

Daniel Lopresti¹, George Nagy², and Ashutosh Joshi²

¹ Computer Science and Engineering,
Lehigh University, Bethlehem, PA 18015

² Electrical, Computer, and Systems
Engineering, Rensselaer Polytechnic
Institute, Troy, NY 12180

Symbolic Indirect Correlation (SIC)

SIC is a new pattern recognition paradigm.

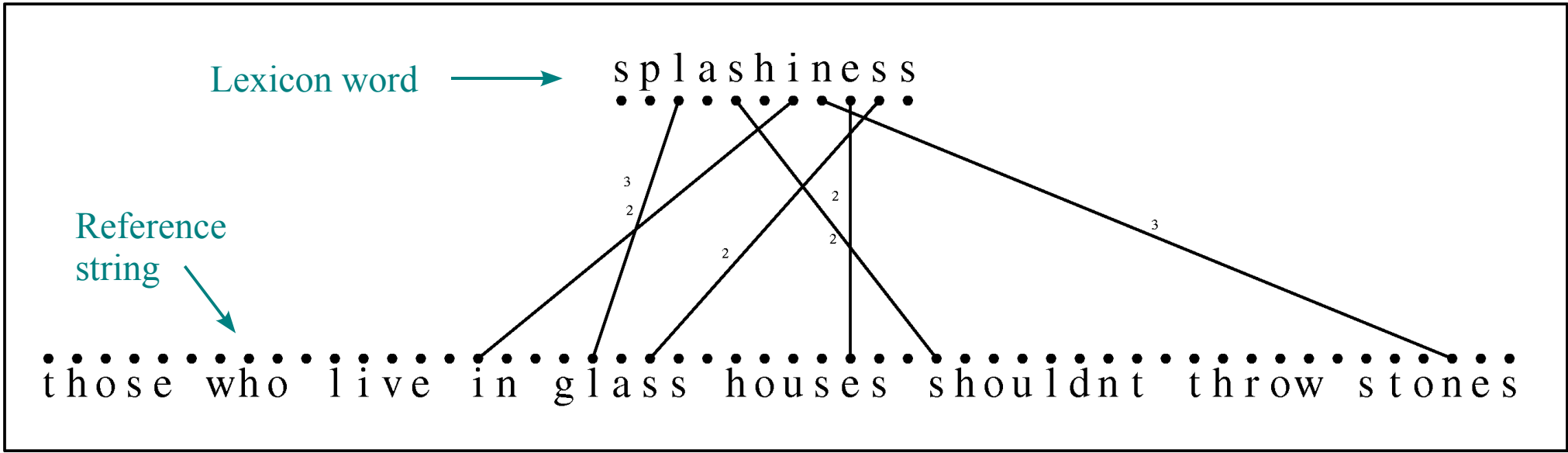
<i>Symbolic</i>	because it exploits the ordering of matches in lexical (symbolic) strings.
<i>Indirect</i>	because it is based on two levels of comparisons.
<i>Correlation</i>	because it can be viewed as making use of sliding windows.

SIC is still a relatively new idea and largely untested.

Outline of SIC Approach

1. *Lexical Matching.* Match polygrams in every lexicon word against the transcription of the reference signal (offline preprocessing).
2. *Feature Matching.* Match feature strings derived from the query and reference signals.
3. *Graph Matching.* Match the feature graph (Step 2) against the lexical graphs (Step 1) for each word in the lexicon.
4. *Result.* Output the best matching lexicon word from Step 3 as the result.

Lexical Match Graph*



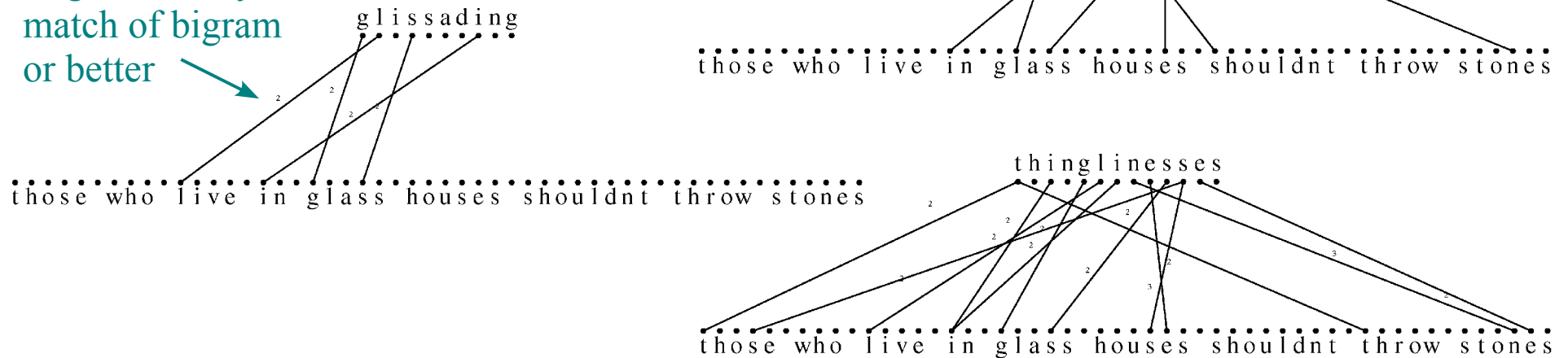
Note there is edge for every match of bigram or better.

* All match graphs shown in this presentation and the paper were generated automatically by running the algorithms in question; none were drawn by hand.

SIC Example

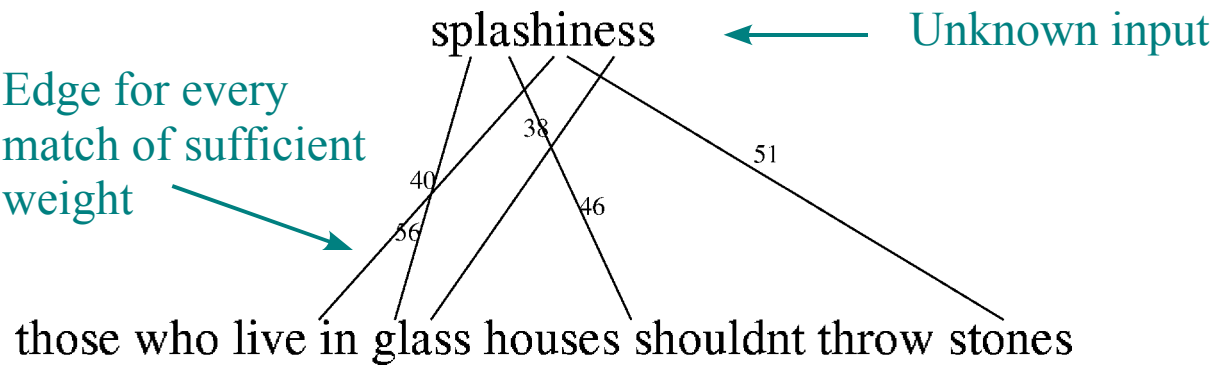
Lexical domain

Edge for every match of bigram or better



Signal domain

Edge for every match of sufficient weight

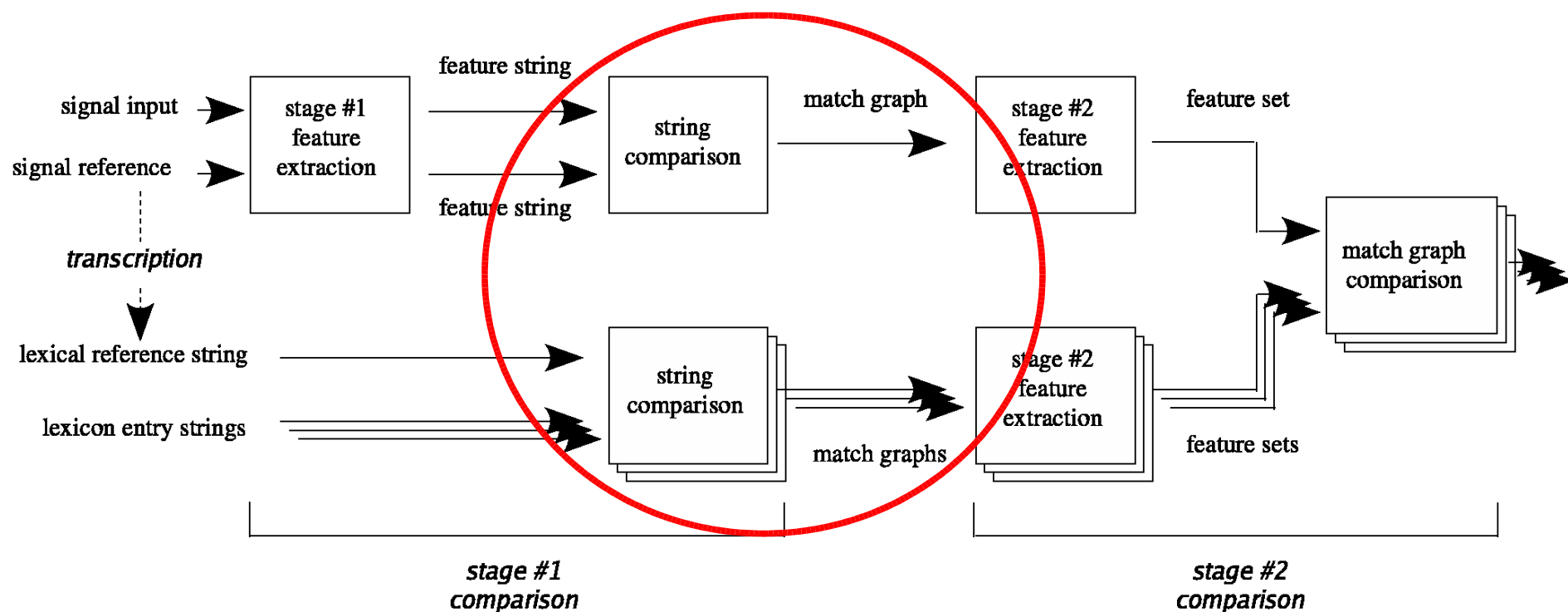


SIC Advantages

- Matches based on signal subsequences of any length, although typically longer than single characters or phonemes.
- Common distortions in handwriting and camera- and tablet-based OCR (stretching, contraction) and speech (time-warping) can be accommodated.
- Independent of medium, feature set, and vocabulary.
- No training – only a reference set as in Nearest Neighbor – thus allowing unsupervised adaptation.
- Extensible to phrase recognition.

Present Study

SIC performance is impacted by errors in any stage:



For this study, we bypass final stages of SIC and compare results of match graph generation directly.

Approximate String Matching

SIC uses Smith-Waterman string matching algorithm*:

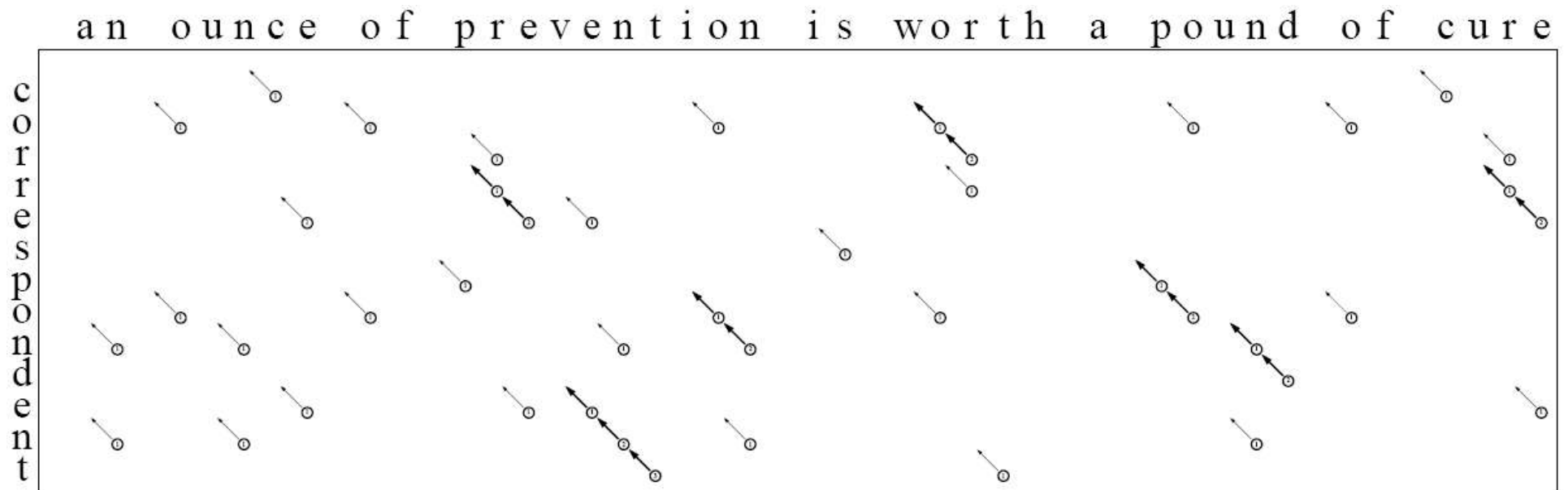
$$\begin{array}{l} dist_{0,0} = 0 \\ dist_{i,0} = 0 \\ dist_{0,j} = 0 \end{array} \quad dist_{i,j} = \min \begin{cases} 0 \\ dist_{i-1,j} + c_{del}(s_i) \\ dist_{i,j-1} + c_{ins}(t_j) \\ dist_{i-1,j-1} + c_{sub}(s_i, t_j) \end{cases}$$
$$1 \leq i \leq m, 1 \leq j \leq n$$

Note this differs from more widely-known Wagner-Fischer (Needleman-Wunsch) version as it allows for multiple matches that can start and end anywhere.

* “Identification of common molecular sequences,” T. F. Smith and M. S. Waterman, *Journal of Molecular Biology*, vol. 147, pp. 195-197, 1981.

Lexical Distance Matrix Example

We have developed a series of visualizations for reviewing results of intermediate steps in computation:



* Again note that this graph was generated automatically by running the algorithm in question.

Signal Features

To evaluate match graph generation, we performed a pilot study using synthesized images of text strings.

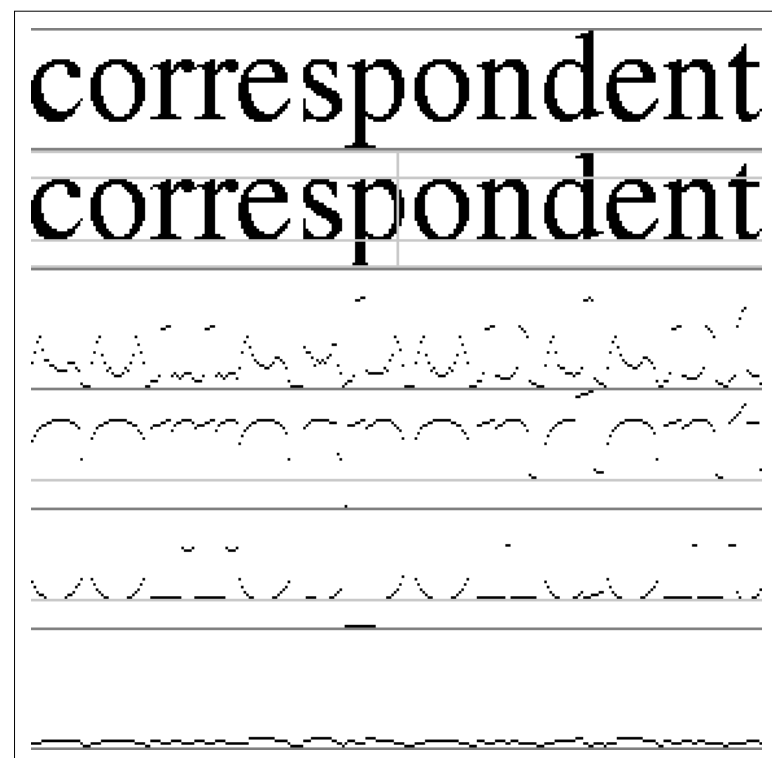
Features are adapted from set used by Manmatha and Rath for offline handwriting.*

Black pixel density →

Upper text contour →

Lower text contour →

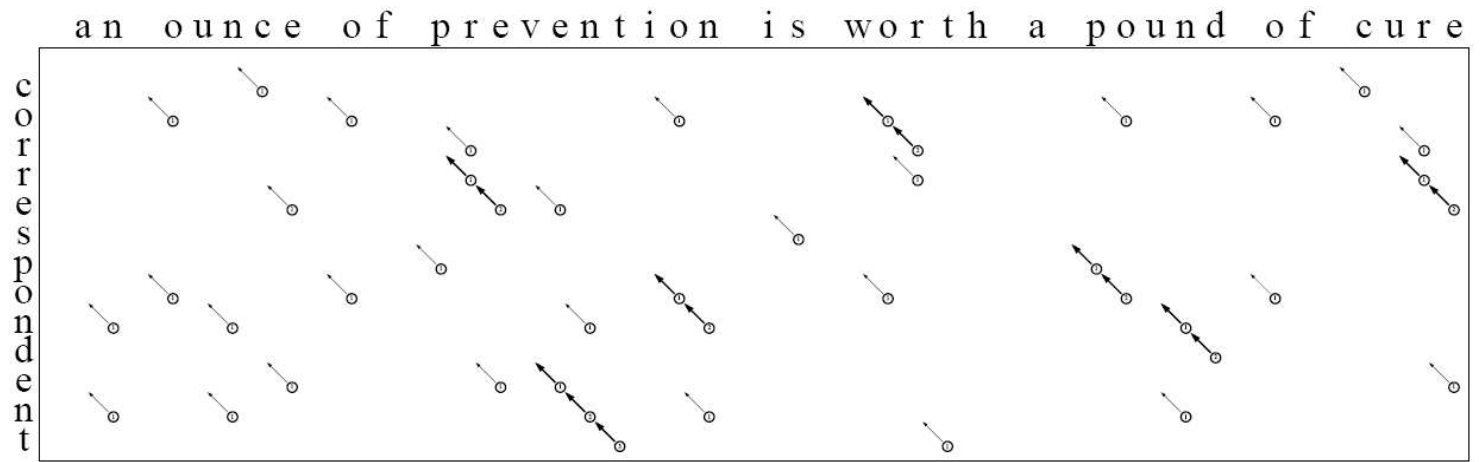
0-1 transitions →



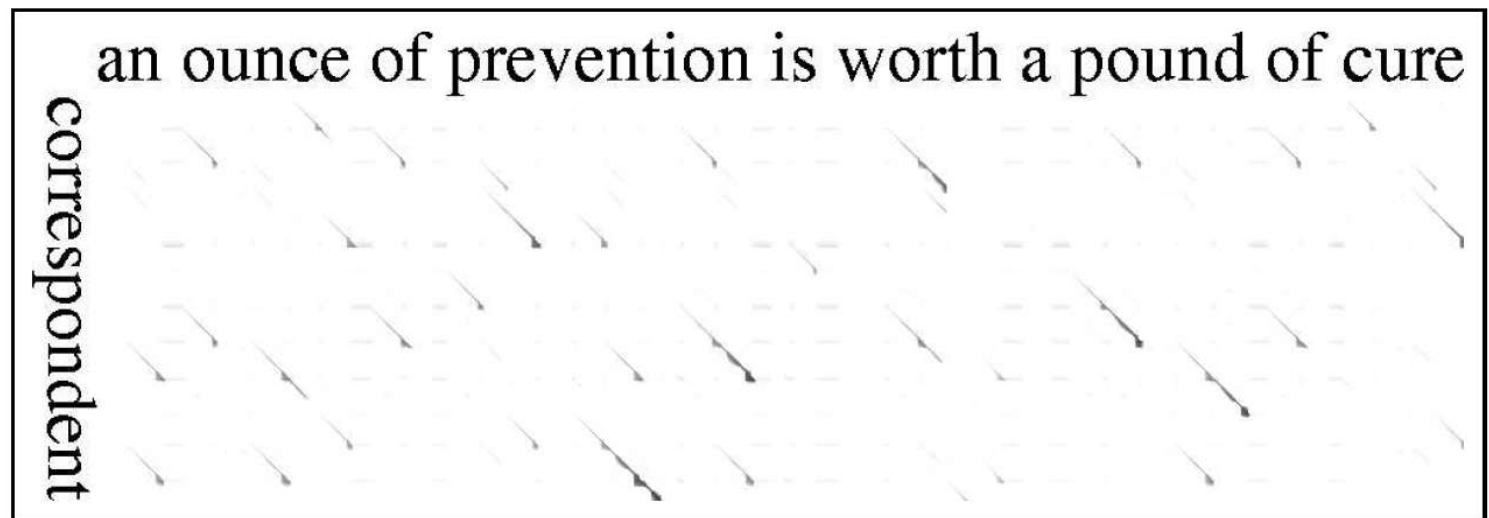
* "Indexing Handwritten Historical Documents – Recent Progress," R. Manmatha and T. Rath, *Proceedings of the Symposium on Document Image Understanding*, pp. 195-197, 2003.

Visualization of Distance Matrices

Result of
lexical
comparison:

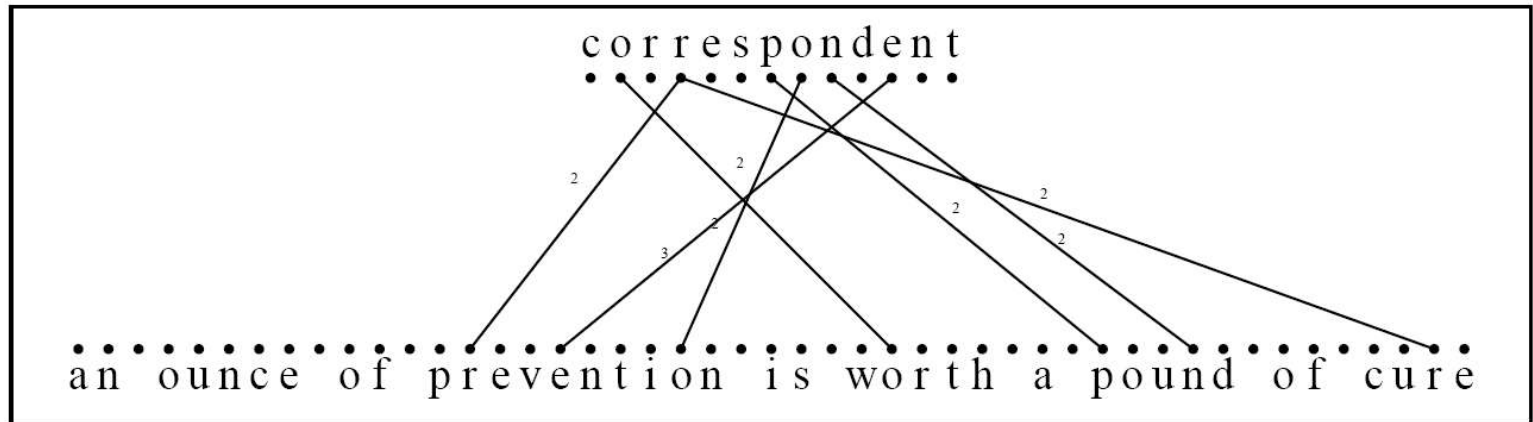


Result of
signal
comparison:

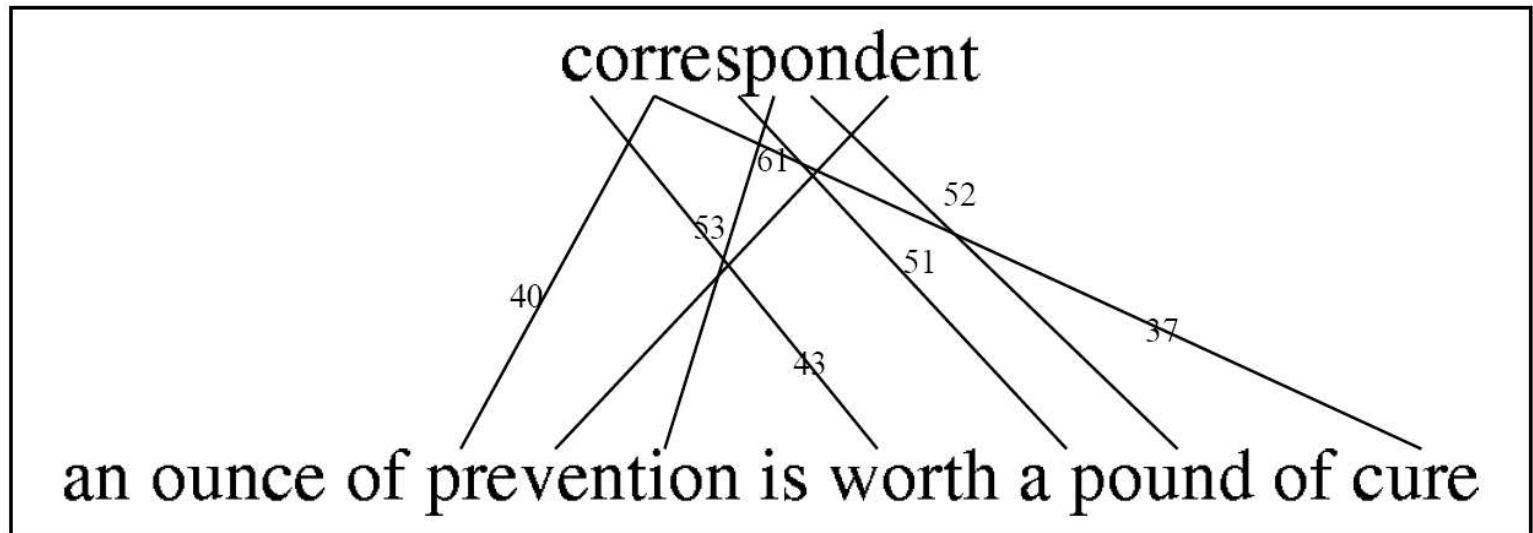


Resulting Match Graphs

Lexical
domain:



Signal
domain:

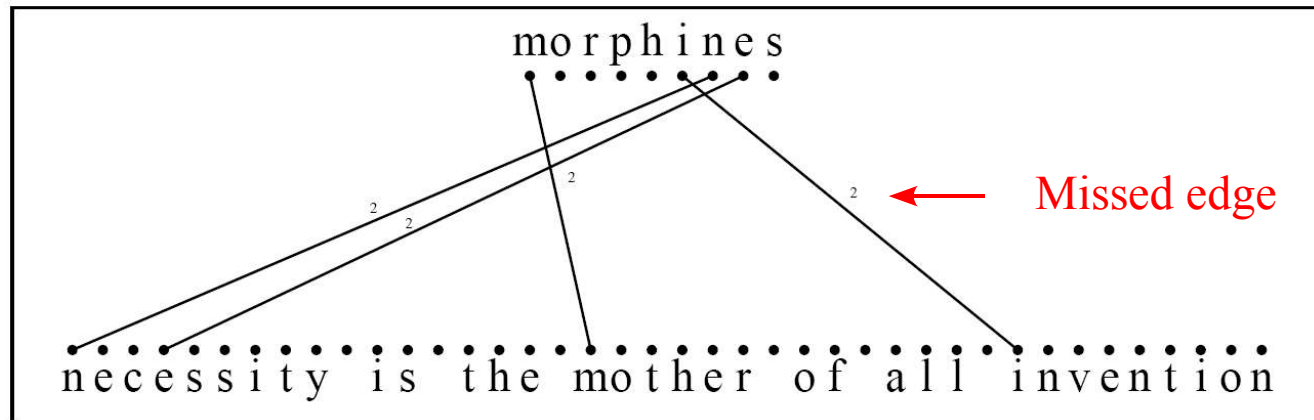


Note that these match graphs correspond perfectly.

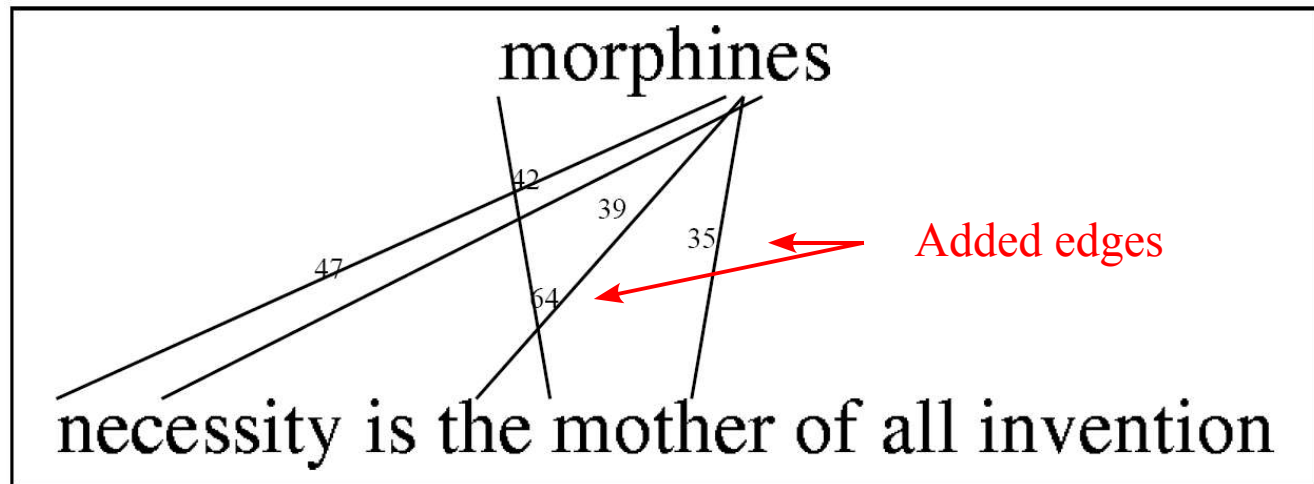
Match Graph Errors

The real world is rarely so cooperative, however.

Lexical
domain:



Signal
domain:

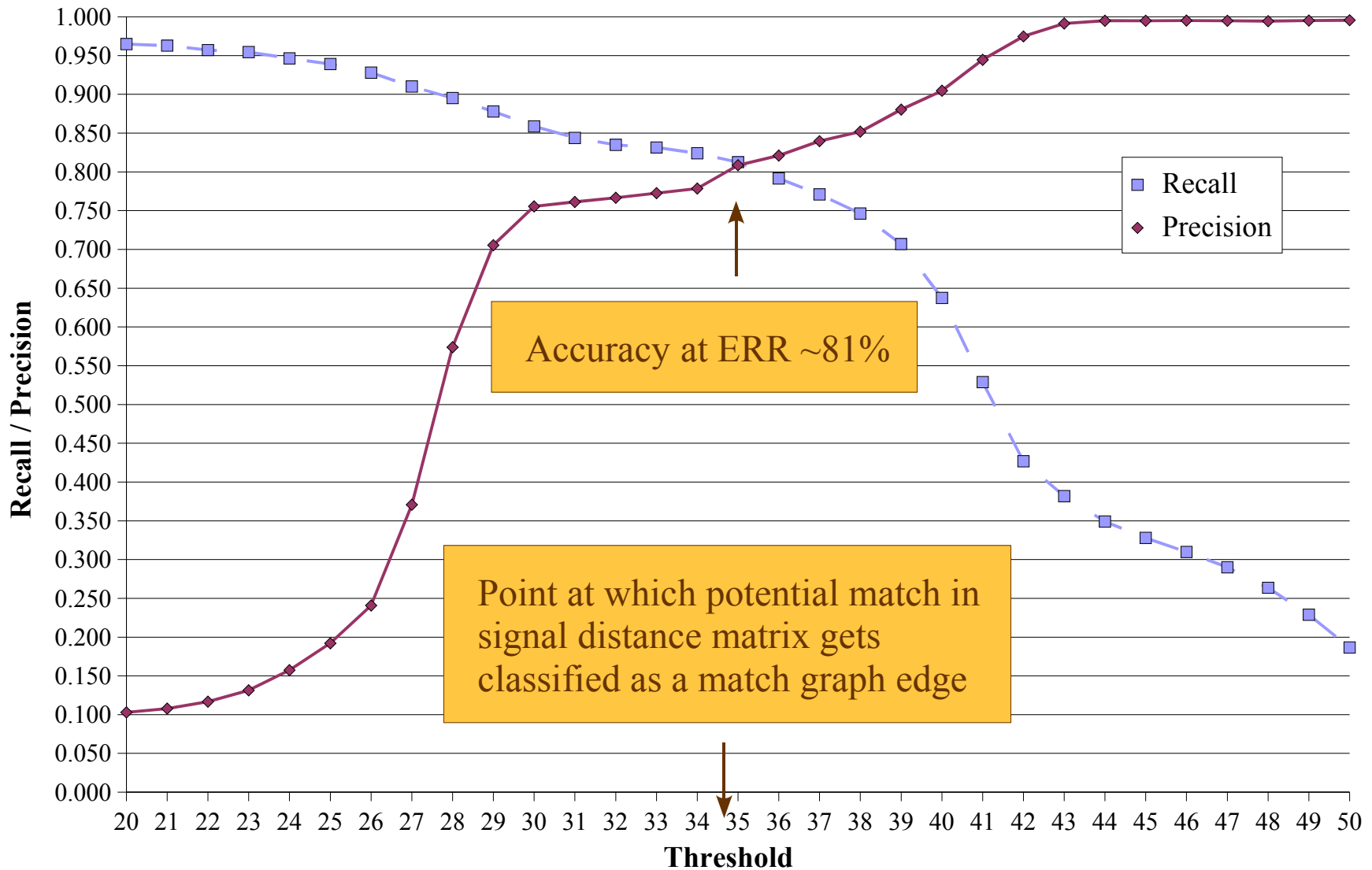


SIC Evaluation

- Employ synthesized TIF bitmaps of known strings.
- Reference strings = 100 random proverbs.
- Query strings = 100 random words from YAWL*.
- Compare match graphs, count missing/added edges.
- Recall = percentage of lexical match graph edges correctly represented in signal match graph.
- Precision = percentage of signal match graph edges truly present in lexical match graph.
- Total match graphs tested = 10,000 (= 100 × 100).

* “Yet Another Word List,” <http://www.ibiblio.org/pub/linux/libs/>.

SIC Results



Most Frequent Edge Effects

Tabulate various effects we saw at optimal threshold:

Rank	Correct Edges		Missed Edges		Spurious Edges	
	Count	Pattern(s)	Count	Pattern(s)	Count	Pattern(s)
1	1,056	“es”	331	“it”, “es”	37	“nes” ↔ “her”
2	971	“th”	219	“st”	34	“me” ↔ “ma”
3	957	“in”	209	“ti”	32	“nes” ↔ “he m”
4	827	“er”	199	“li”	30	“mo” ↔ “ma”
5	537	“re”	178	“is”	25	“mo” ↔ “me”
6	471	“at”	141	“re”	24	“me” ↔ “mo”
7	426	“on”	124	“ll”	22	“nes” ↔ “he b”
8	394	“an”	95	“te”	21	“nes” ↔ “he h”
9	391	“he”	82	“er”, “at”	19	“nes” ↔ “he d”, “owe” ↔ “wi”, “mo” ↔ “mi”
10	380	“nt”	57	“en”	18	“me” ↔ “mi”

- Missed edges due largely to thin characters (e.g., i).
- Spurious edges due to feature similarity, including character prefixes and suffixes (e.g., h ↔ n).

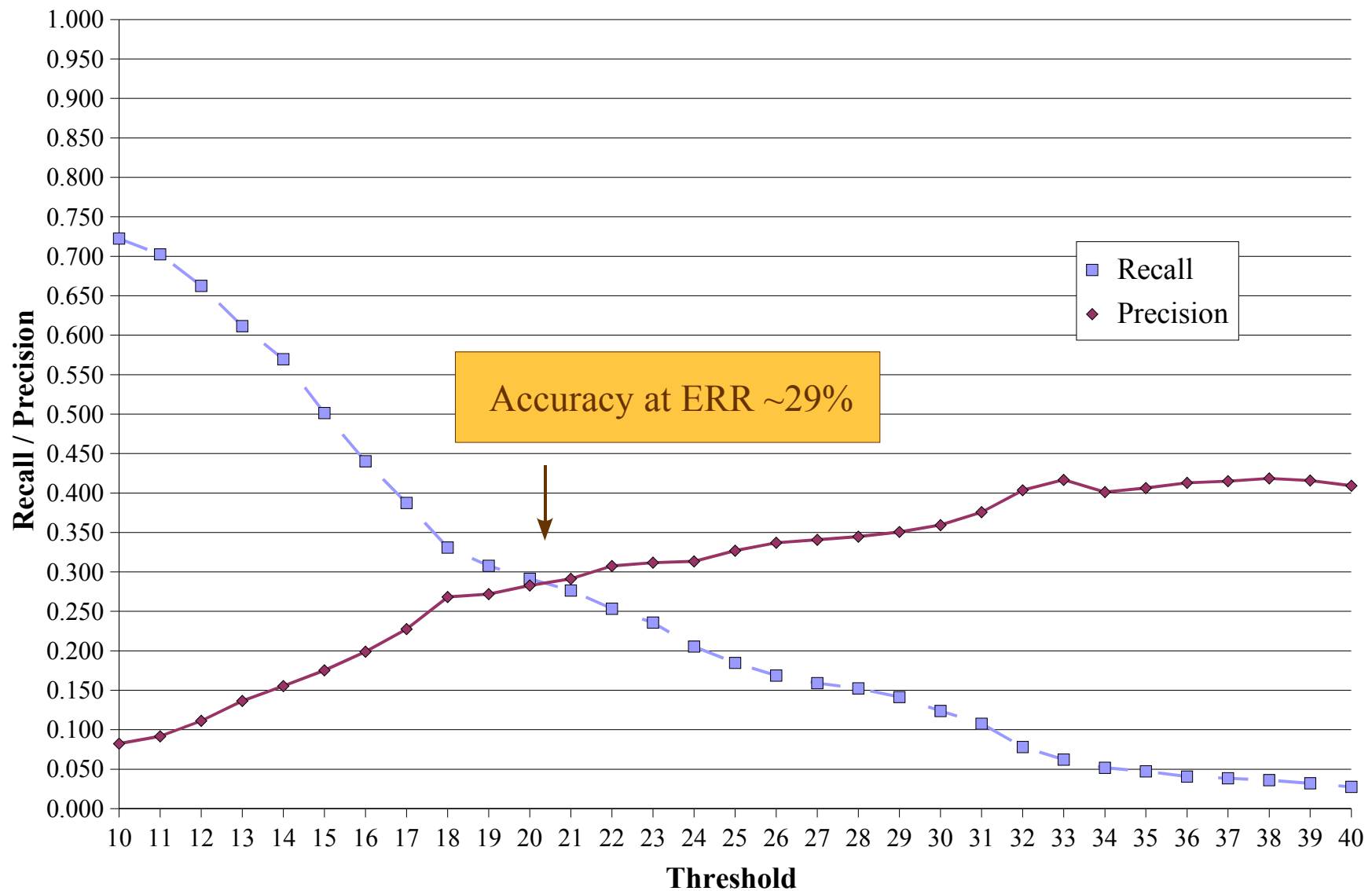
More Challenging Evaluation

- SIC proposed for handling hard-to-segment inputs.
- Repeat exact same experiment, only this time using highly condensed text strings.

correspondent

correspondent

SIC Results (Condensed Text)



Conclusions

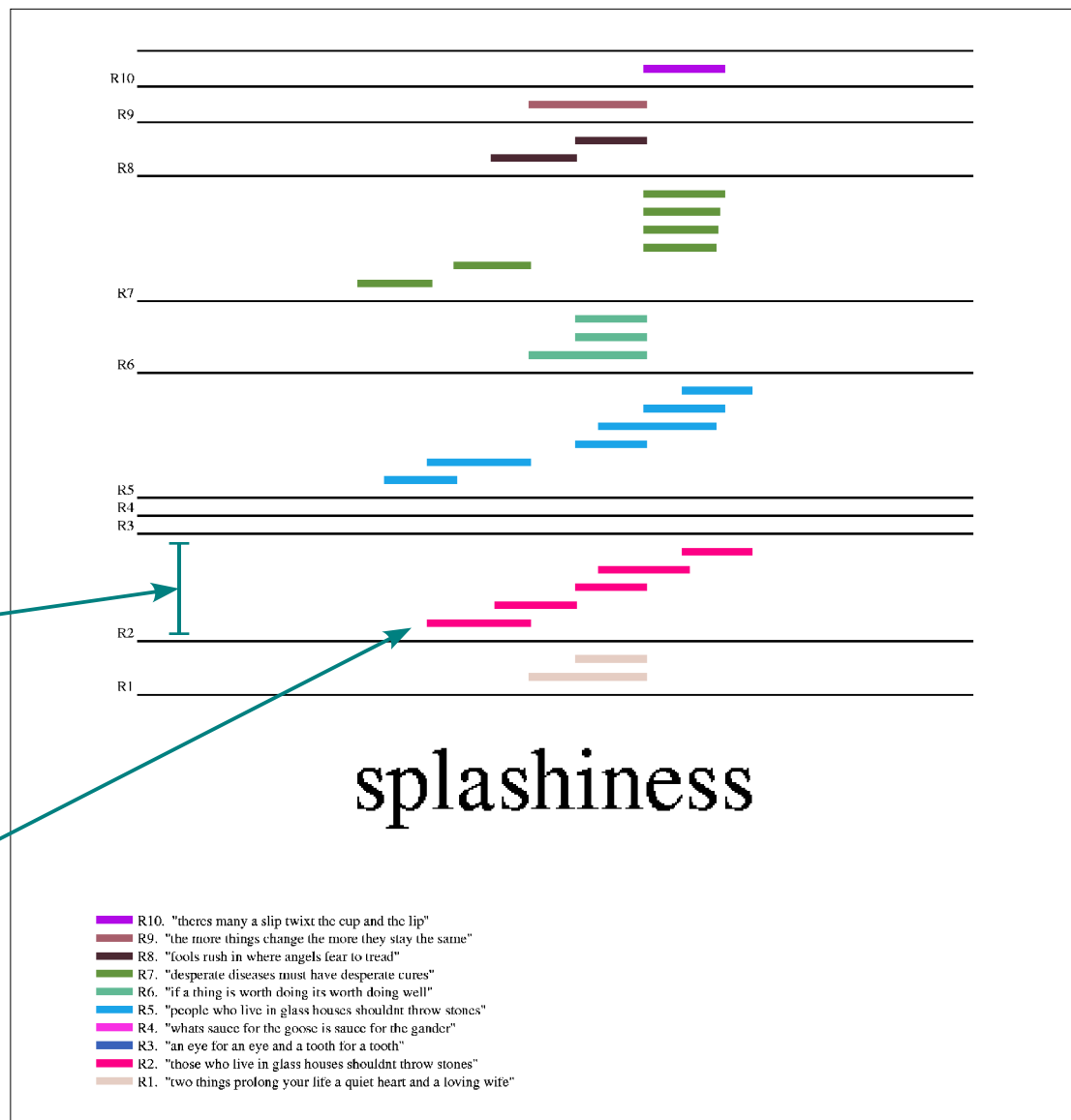
- Smith-Waterman approach appears to be right model for building match graphs.
- Current problems lie with feature representation. Some issues may be challenging to surmount (e.g., suffix of “h” will always resemble suffix of “n”).
- On the other hand, final stage of SIC has ability to overcome a certain number of errors.
- Future work includes exploring connection between match graph errors and overall SIC error rate, as well as extending evaluation to real handwriting and scanned text inputs (appropriately ground-truthed).

Visualizing Multiple Matching Results

Results of comparing signal input “splashiness” to 10 different reference strings:

Each reference string corresponds to a set of colored bars

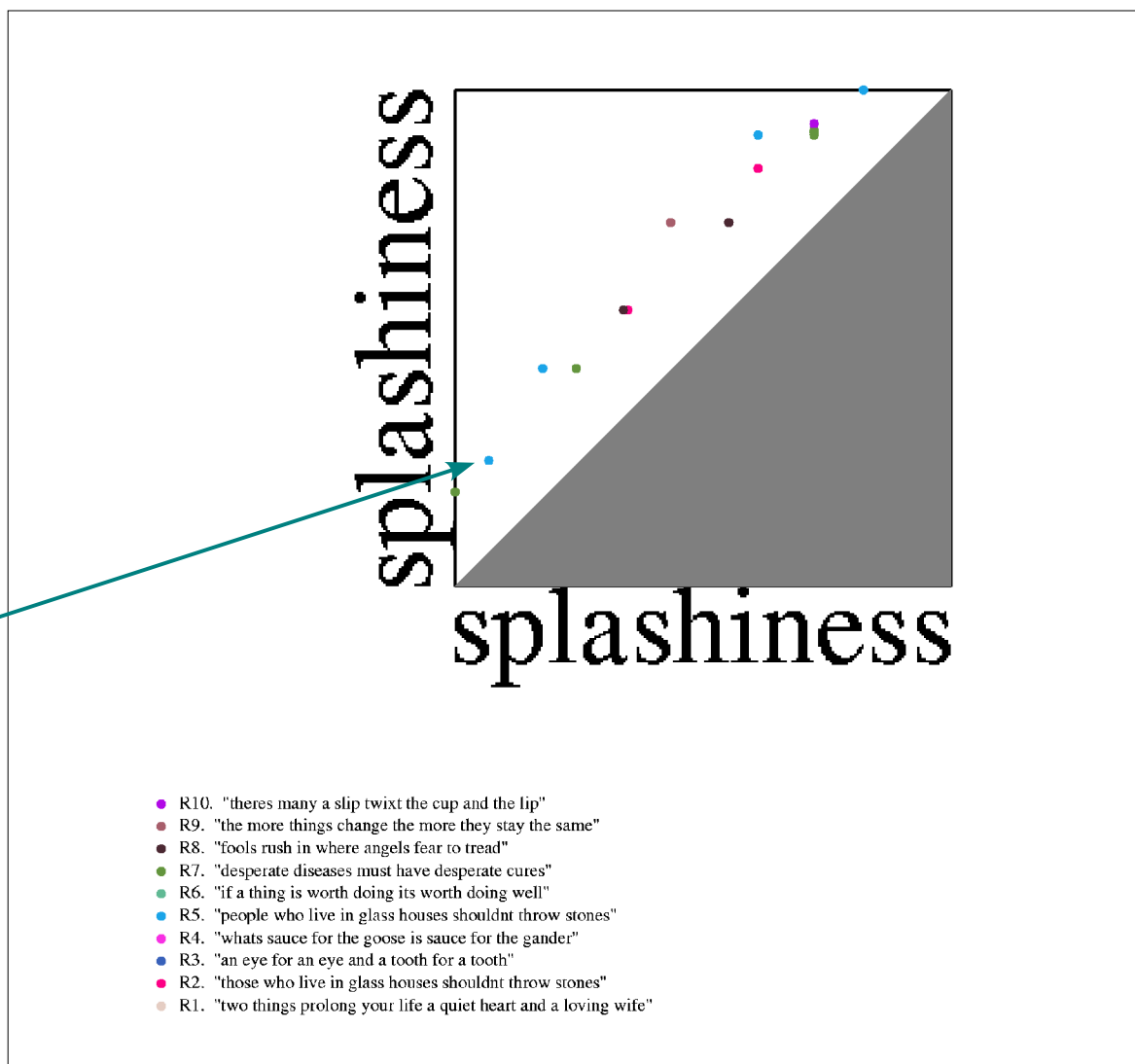
Each colored bar records starting and ending positions of one match along signal input





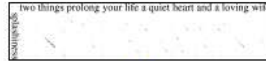
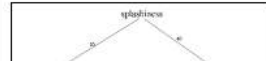
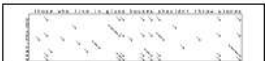

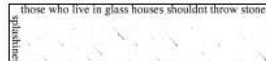

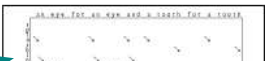
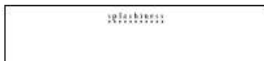
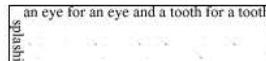
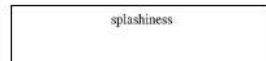


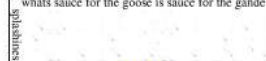

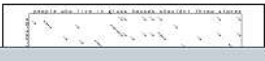
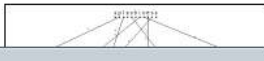
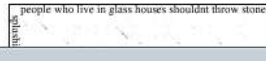

Visualizing Multiple Matching Results

Results of comparing signal input “splashiness” to 10 different reference strings:

Each match corresponds to a single datapoint. x-coordinate records starting position, y-coordinate records ending position.



Web Browser Interface to SIC Results

Query String	Reference String	Lexical Distance Matrix	Lexical Match Graph	Signal Distance Spectrum	Signal Match Graph
<i>plashiness</i> features: GIF char positions: TXT	<i>two things prolong your life a quiet heart and a loving wife</i> features: GIF char positions: TXT	 GIF PS	 GIF PS	 GIF	 GIF PS
<i>plashiness</i> features: GIF char positions: TXT	<i>those who live in glass houses shoudnt throw stones</i> features: GIF char positions: TXT	 GIF PS	 GIF PS	 GIF	 GIF PS
<i>plashiness</i> features: GIF char positions: TXT	<i>an eye for an eye and a tooth for a tooth</i> features: GIF char positions: TXT	 GIF PS	 GIF PS	 GIF	 GIF PS
<i>plashiness</i> features: GIF char positions: TXT	<i>whats sauce for the goose is sauce for the gander</i> features: GIF char positions: TXT	 GIF PS	 GIF PS	 GIF	 GIF PS
	<i>people who live in glass houses</i>	 GIF PS	 GIF PS	 GIF	 GIF PS

Each table row corresponds to one query-reference comparison

Thumbnail images are clickable to hires versions