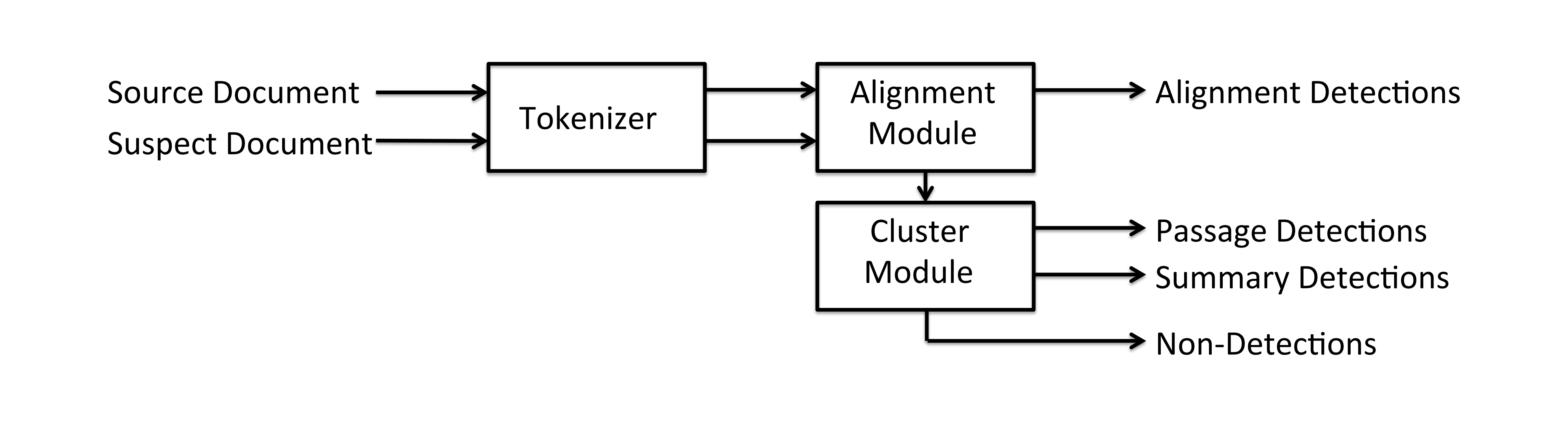
A Hybrid Architecture for Plagiarism Detection

Demetrios Glinos
University of Central Florida, Orlando, Florida USA
glinos@eecs.ucf.edu

Hybrid Design

- Combine different techniques for different types of text plagiarism
- Use text alignment method (extended Smith-Waterman dynamic programming algorithm) for order-based plagiarism
- Use concept clustering method (several variations) for non-order based plagiarism

Processing Flow



Order-Based Plagiarism

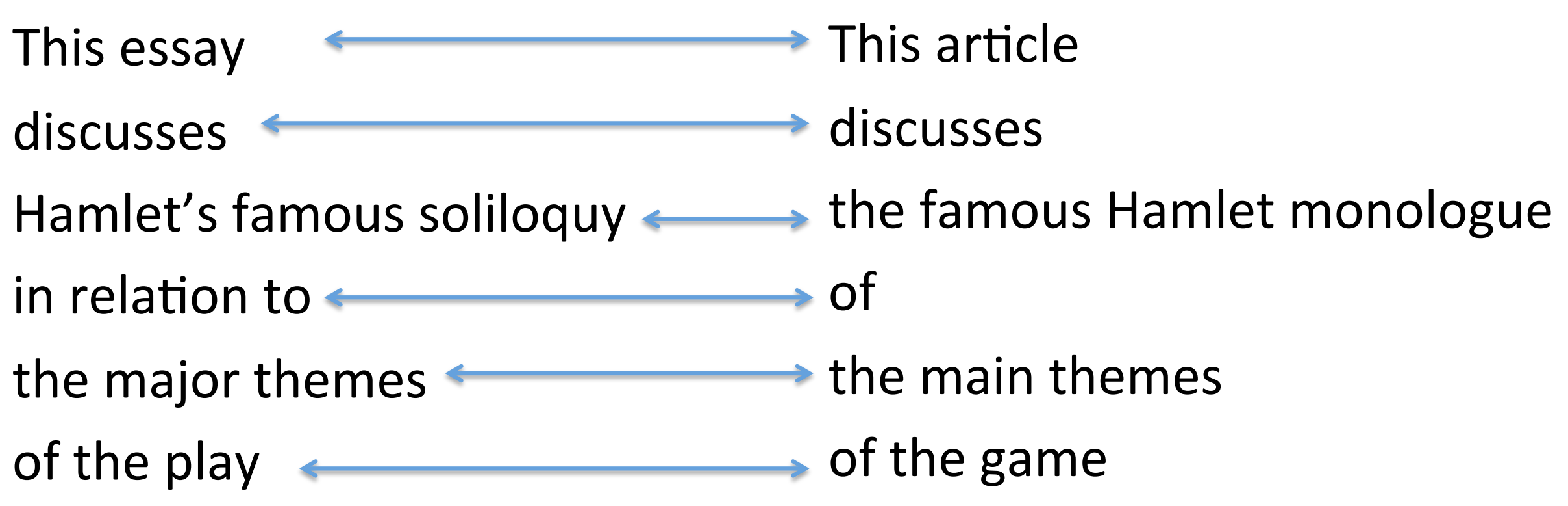
Key feature:
Concepts in both documents appear in substantially the same order, possibly with some additions, deletions and differences.

Source sentence:

This essay discusses Hamlet 's famous soliloquy in relation to the major themes of the play.

Suspect sentence:

This article discusses the famous Hamlet monologue of the main themes of the game.



This essay	This article
discusses	discusses
Hamlet's famous soliloquy	the famous Hamlet monologue
in relation to	of
the major themes	the main themes
of the play	of the game

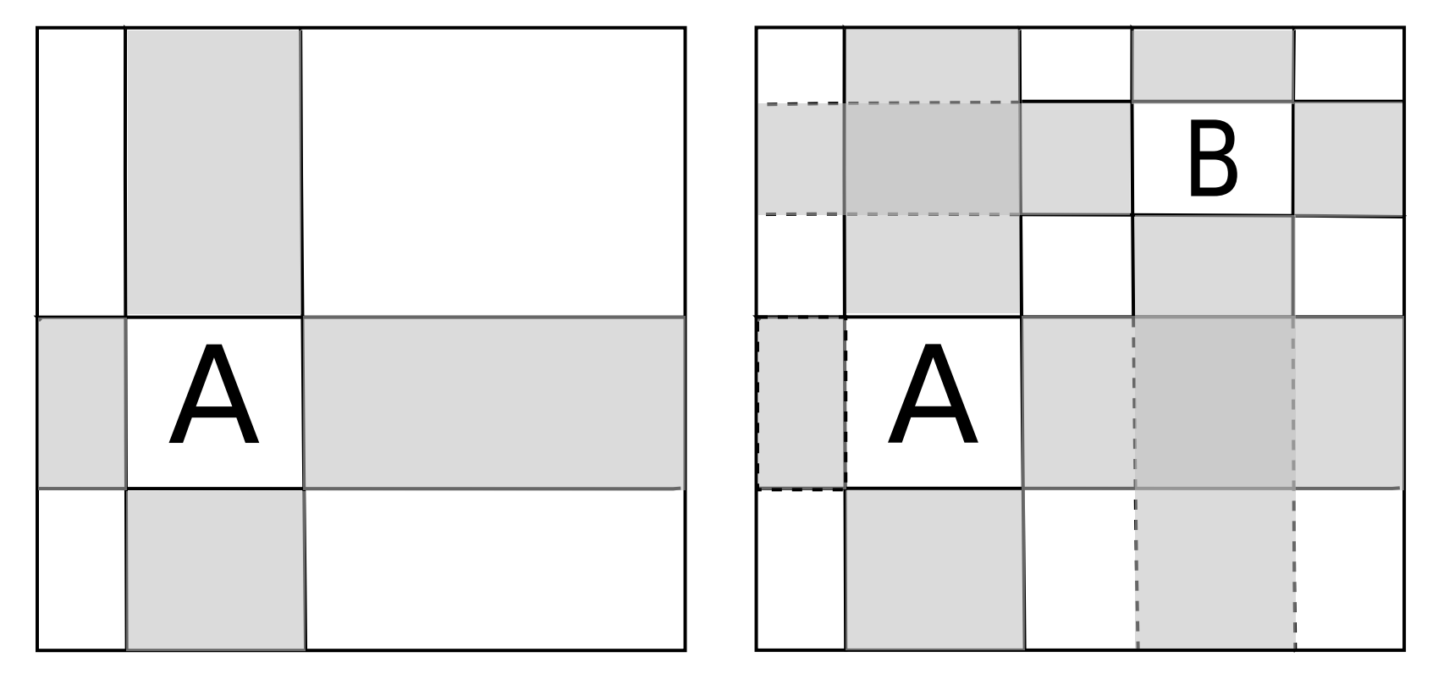
Text Alignment

$$M(i,j) = max \begin{cases} M(i-1,j-1) + match(a_i,b_j) \\ M(i-1,j) + gap \\ M(i,j-1) + gap \\ 0 \end{cases}$$

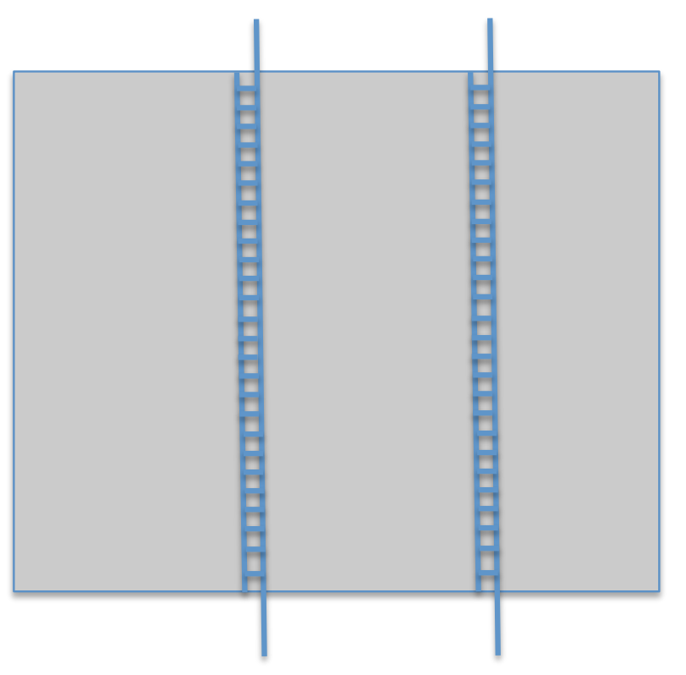
where $match(a_i,b_j) = +2$, if $a_i = b_j$; and -1 otherwise; and where $gap = -1$ is the gap penalty.

		This	essay	discusses	Hamlet	's	famous	soliloquy	in	relation	to	the	major	themes	of	the	play	.	tempus	fugit
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
This	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
article	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
discusses	0	0	0	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
the	0	0	0	2	2	1	0	0	0	0	0	2	1	0	0	2	1	0	0	0
famous	0	0	0	1	1	1	3	2	1	0	0	1	1	0	0	1	1	0	0	0
Hamlet	0	0	0	0	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0
monologue	0	0	0	0	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0
of	0	0	0	0	1	1	1	0	3	2	2	1	0	0	2	1	0	0	0	0
the	0	0	0	0	0	0	0	0	2	2	1	4	3	2	1	4	3	2	1	0
main	0	0	0	0	0	0	0	0	1	1	1	3	3	2	1	3	3	2	2	0
themes	0	0	0	0	0	0	0	0	0	0	0	2	2	5	4	3	2	2	1	0
of	0	0	0	0	0	0	0	0	2	1	2	1	1	4	7	6	5	4	3	2
the	0	0	0	0	0	0	0	0	1	1	1	4	3	3	6	9	8	7	6	5
game	0	0	0	0	0	0	0	0	0	0	0	3	3	2	5	8	8	7	6	5
.	0	0	0	0	0	0	0	0	0	0	0	2	2	2	4	7	7	10	9	8
carpe	0	0	0	0	0	0	0	0	0	0	0	1	1	1	3	6	6	9	9	8
diem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	5	8	8	8

Recursive Descent



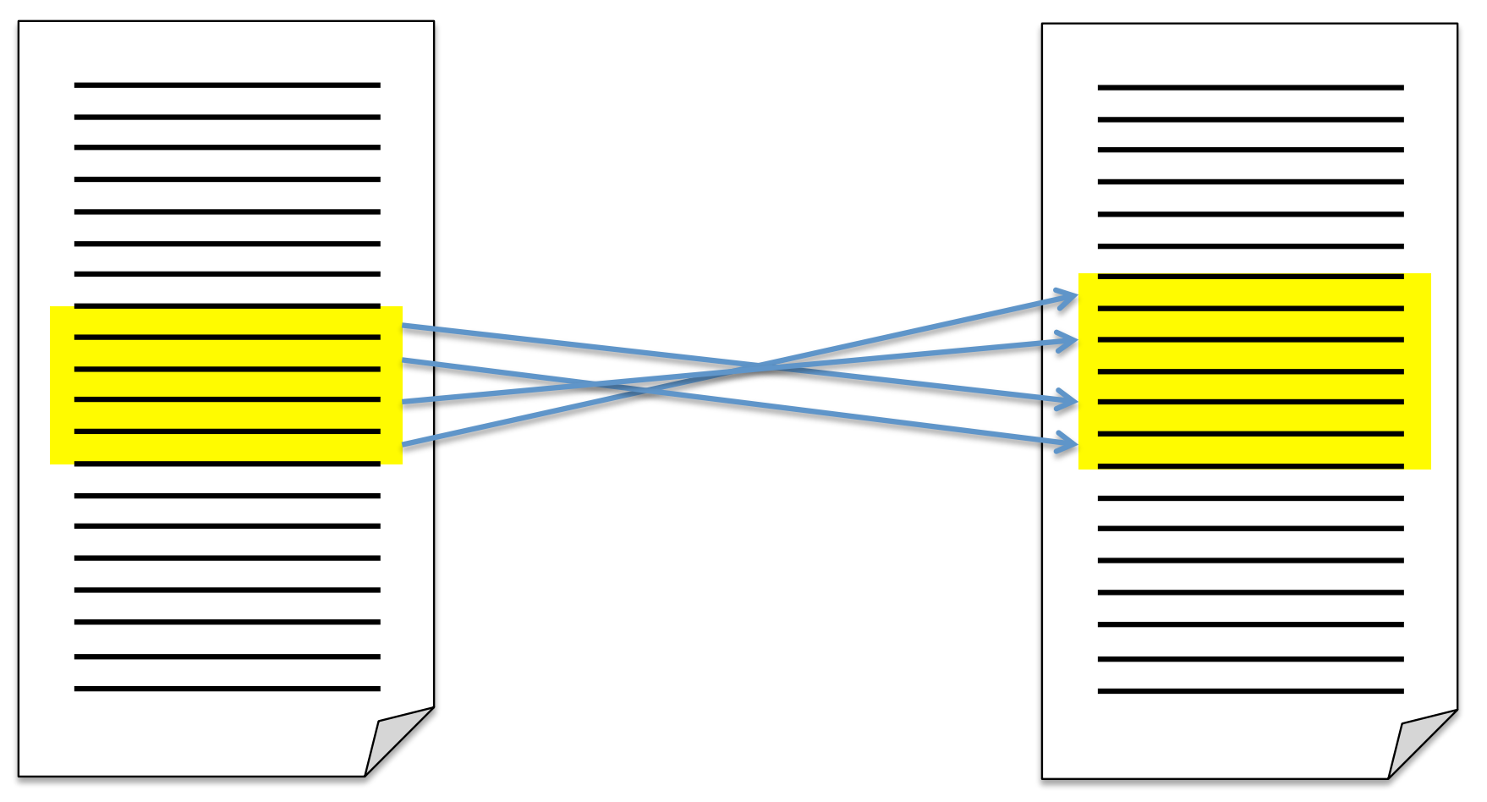
Matrix Splicing



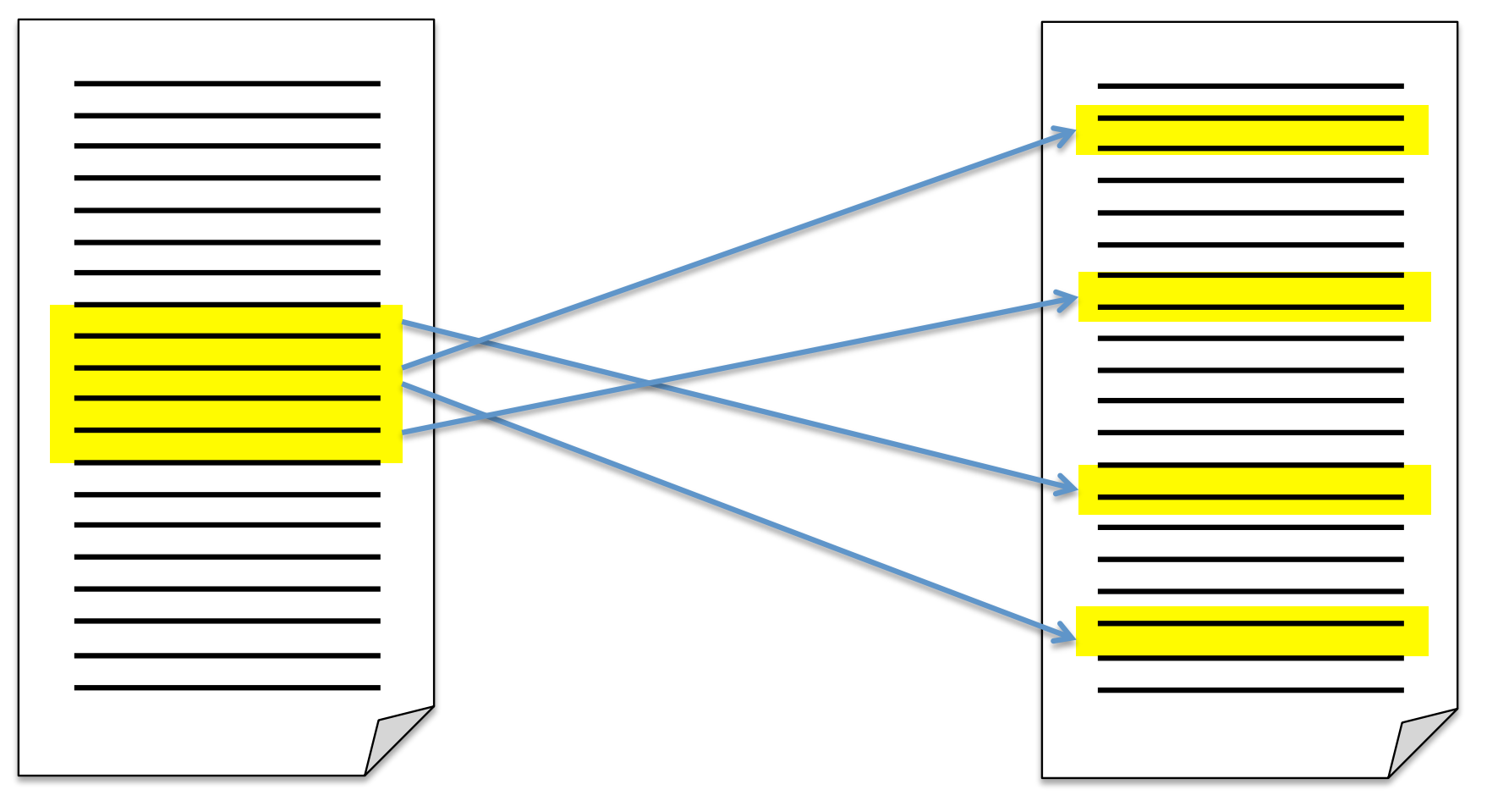
- Slice to fit segment within available memory
- Column to left preserves state, allowing chains to cross boundaries

Non-order Based Plagiarism

Passage Detection



Summary Detection



UNIVERSITY OF CENTRAL FLORIDA

Clustering Algorithms

Basic Clustering:

- Form trigrams centered on 30 most frequent tokens at least 5 characters long that start with letter in the source document.
- If trigram set spans at least 80% of source document, then search for occurrences in suspect document.
- Merge into clusters if within 20 tokens.
- Accept largest cluster at least 40 tokens in length, if any, and report summary detection.

Word Clustering:

- Form bigrams from 20 most frequent tokens and 20 most frequent tokens in source document that start with an uppercase letter of at least 5 characters in length.
- Find occurrences of bigrams in suspect document and merge into clusters if within 15 token.
- Keep clusters at least 40 tokens long that contain at least 8 source terms and choose maximal cluster that has Jaccard coefficient at least 0.65 computed on source concept words and content words in the suspect cluster, excluding stop words.
- Attempt to find a passage in the source document with Jaccard coefficient at least 0.50 for concept words in maximal suspect cluster.
- If source passage found, then report passage detection; otherwise, if only suspect cluster found, report summary detection.

Bigram Clustering:

- Apply word clustering as above, but use threshold of 4 source concepts instead of 8.
- If a maximal suspect passage is found, but no corresponding source passage is found, compute all bigrams for suspect passage.
- In such case, compute bigrams and Jaccard coefficients for all source clusters and report a passage detection if there is a maximal cluster with at least 0.25 Jaccard value; otherwise, report summary detection.

Test Data

Plagiarism type	*Test Corpus 1*	*Test Corpus 2*	*Test Corpus 3*
No plagiarism	90	1000	1600
No obfuscation	108	1000	1600
Random obfuscation	94	1000	1600
Cyclic translation	105	1000	0
Summary obfuscation	121	1185	0
Total document pairs	518	5185	4800

Test Results

System	*Measure*	*Test Corpus 1 (518 docs)*	*Test Corpus 2 (5185 docs)*	*Test Corpus 3 (4800 docs)*
Basic Clustering	Recall	0.77088	0.76389	0.83473
	Precision	0.96735	0.96726	0.96243
	Granularity	1.01479	1.01756	1.01783
	PlagDet Score	0.84899	0.84300	0.88274
Word Clustering	Recall	0.79327	0.79105	0.84248
	Precision	0.96524	0.96339	0.96022
	Granularity	1.01441	1.01700	1.01767
	PlagDet Score	0.86192	0.85827	0.88626
Bigram Clustering	Recall	0.79469	0.79331	0.84511
	Precision	0.96599	0.96253	0.96007
	Granularity	1.01437	1.01695	1.01761
	PlagDet Score	0.86309	0.85930	0.88770