Discovering Similar Passages Within Large Text Documents

Demetrios Glinos
glinos@eecs.ucf.edu

The Problem Domain

- The task is to find one or more passages in one document that are the same or closely similar to passages in another document.
- There can be more than one matching set of passages in a given document pair.
- Corresponding passages may not be in the same order in each document.
- Corresponding passages need not be identical, only similar:
  - Additions or deletions of words and phrases
  - Use of synonyms
  - Alternate grammatical constructions
- Each passage pair, however, presents a text alignment problem.

UNIVERSITY OF CENTRAL FLORIDA

Application Areas

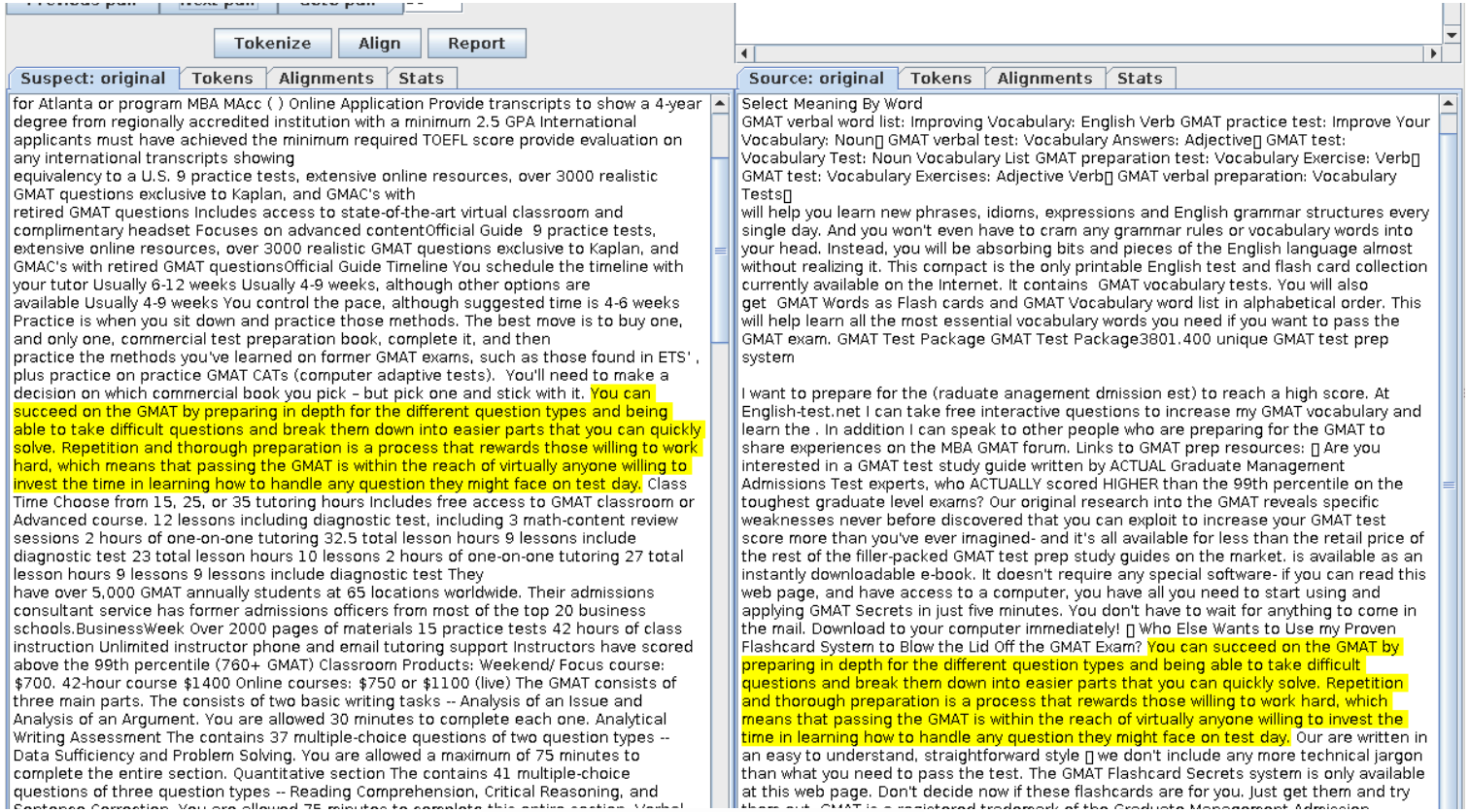
- Document deduplication
  - ***Example:*** Recognizing that two documents represent the same content when building a database of medical journal articles or abstracts retrieved from different online sources.

- Textual Entailment Determination
  - ***Example:*** Recognizing that two sentences mean the same thing despite different grammatical constructions and that can spoof deep parsers.

- Plagiarism Detection
  - ***Example:*** Recognizing that one document contains substantial passages that have been copied, perhaps modified, from another.

A Simple Example of Cut-and-Paste

- Here, the task is simply to *find* the coresponding passage(s), if any.



How Difficult Can This Be?

- Consider two 5,000-word documents that contain a common passage (i.e., no differences), but we don't know anything about it, not even its length.

- An exhaustive search must test:
  - Every valid length from 1 to 5,000
  - Every shingle of each length in each document
  - Average number of shingles is 2,500

- Result is approx. (5000)(2500)(2500) = over 30 billion passage comparisons.

- This is $O(n^3)$ complexity. If differences are allowed, search is $O(n^4)$.

Our Approach

- Take advantage of the fact that, despite differences, similar passages tend to have aligned concepts.

- We borrow the **Smith-Waterman** dynamic programming algorithm from the bioinformatics community.

- We extend it for large document text similarity applications by specifying:
  - **Recursive descent** – to support discovery of multiple passage pairs
  - **Matrix splicing** – for handling large documents
  - **Chaining** – for connecting passage components
  - **Relaxed similarity measure** – for identifying token matches

A simple (but actual) example

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

```
(ROOT
 (S
  (NP (DT This) (NN essay))
  (VP (VBZ discusses)
   (NP
    (NP
     (NP (NNP Hamlet) (POS 's))
     (JJ famous) (NN soliloquy))
    (PP (IN in)
     (NP (NN relation))))
   (PP (TO to)
    (NP
     (NP (DT the) (JJ major) (NNS themes))
     (PP (IN of)
      (NP (DT the) (NN play))))))
  (. .)))
```

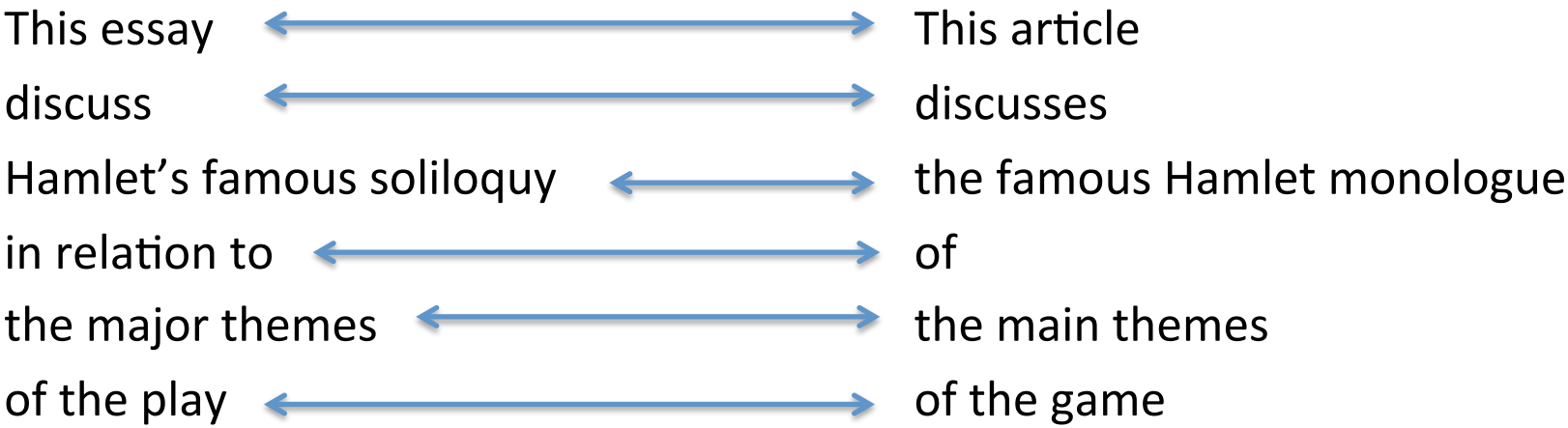
This article discusses the famous Hamlet monologue of the main themes of the game.

```
(ROOT
 (S
  (NP (DT This) (NN article))
  (VP (VBZ discusses)
   (NP
    (NP (DT the) (JJ famous) (NNP Hamlet) (NN monologue))
    (PP (IN of)
     (NP
      (NP (DT the) (JJ main) (NNS themes))
      (PP (IN of)
       (NP (DT the) (NN game)))))))
  (. .)))
```

Concept Alignment

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

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



The Smith-Waterman Algorithm

- Uses dynamic programming to build a match matrix for the two input documents
- Finds the maximal length alignment
- The algorithm:

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

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

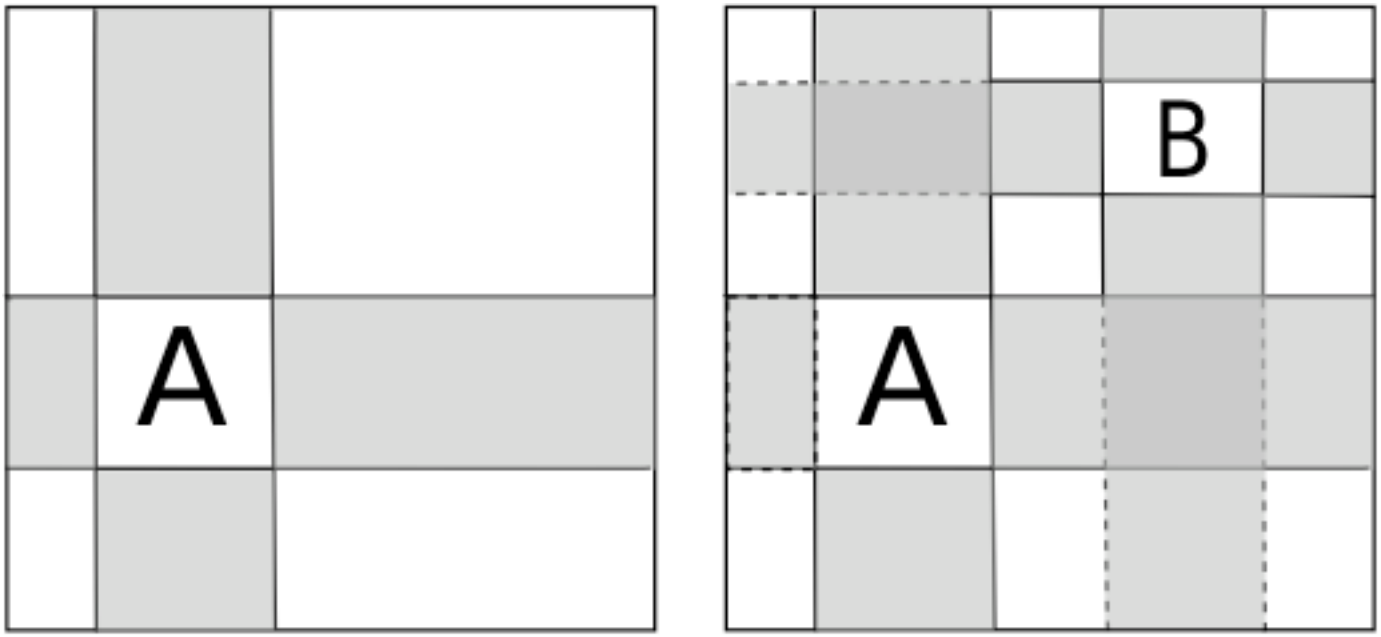
UNIVERSITY OF CENTRAL FLORIDA

The Match Matrix

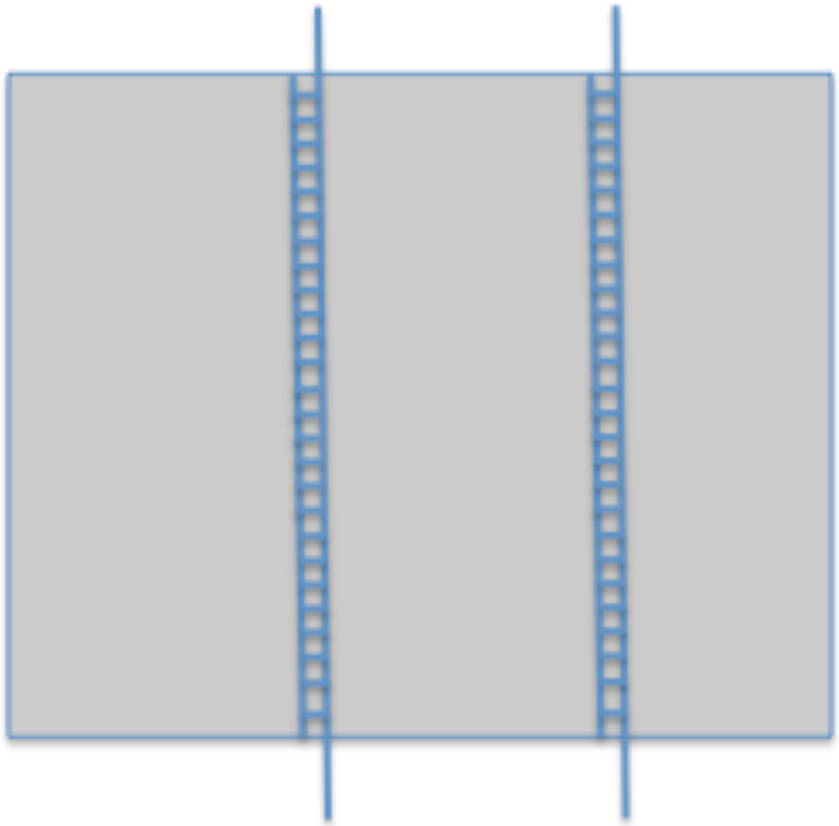
		This	essay	discusses	Hamlet	's	famous	soliloquy	in	relation	to	the	major	themes	of	the	play	.	tempus	fugit
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
This	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
article	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
discusses	0	0	0	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
the	0	0	0	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

- Apply algorithm recursively to unused regions of document space



Matrix Splicing



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

Chaining

- Bridge gaps along diagonals if continue on both sides
- Limit 2 gaps bridged per chain

		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

Relaxed Similarity Measure

- Different authors and speakers often use different articles and prepositions when expressing the same concept.

- When testing for matches while building up the match matrix:

  - Equate determiners: *a, an, the*

  - Also equate common prepositions:

    of, in, to, for, with, on, at, from, by, about, as, into, like, through, after, over, between, out, against, during, without, before, under, around, among

UNIVERSITY OF CENTRAL FLORIDA

Test Data

- Although not a perfect match for this algorithm, we chose the 2013 PAN text alignment test corpus, comprising
  - 5,185 document pairs from 3,169 source and 1,826 suspect documents
    - 1,000 pairs each involving ***no plagiarism***, ***no obfuscation***, ***random obfuscation***, and ***cyclic translation*** plagiarism
    - 1,185 pairs involving ***summary*** plagiarism
  - Source documents:
    - min/mean/max: 104 / 914 / 12,277 words
  - Suspect documents:
    - min/mean/max: 131 / 2,930 / 20,297 words

Aggregate Performance

Target Corpus	PlagDet	Recall	Precision	Granularity	Runtime
No plagiarism	undefined	undefined	0.0000	undefined	1:41.2013
No obfuscation	0.9624	0.9603	0.9644	1.0000	6:47.717
Random obfuscation	0.7958	0.7073	0.9732	1.0413	5:09.562
Cyclic obfuscation	0.8441	0.7506	0.9730	1.0056	6:55.170
Summary obfuscation	0.0984	0.0560	0.9794	1.1099	2:52.770
Overall	0.8404	0.7588	0.9690	1.0177	22:23.481

- Precision uniformly high
- Recall for summary near nil
  - Understandable, since summaries inherently do not preserve order of concepts

UNIVERSITY OF CENTRAL FLORIDA

Detection Counts

Target Corpus	Document Pairs	Reports	Detections	Cases	Cases Detected
No plagiarism	1,000	5	0	0	0
No obfuscation	1,000	1,160	1,159	1,206	1,159
Random	1,000	1,114	1,109	1,292	1,065
Cyclic	1,000	1,093	1,084	1,308	1,078
Summary	1,185	103	101	236	91
Overall	5,185	3,475	3,453	4,042	3,393

- Low false alarm rate overall
- Manual examination of a number of summary cases detected indicate that the summaries that were detected were largely cut-and-paste excerpts (for which concepts are aligned)

Conclusions and Improvements

- Conclusions

  1. Works well for detecting similar texts whose concepts are more-or-less aligned.

  2. Not well when concepts are not aligned.

  3. Can be a valuable component of a larger system for plagiarism detection (e.g., our entry in PAN 2014, which performed well)

- Improvements

  1. Explicitly include synonymy in similarity determinations

  2. Fine tune treatment of beginnings and endings of chains

UNIVERSITY OF CENTRAL FLORIDA