IDENTIFYING SIMILAR LEARNING OBJECTS INCREMENTALLY

NAVEEN NELAPUDI

COMMITTEE:

Dr. SUSAN GAUCH (CHAIR)
Dr. JERZY GRZYMALA BUSSE
Dr. XUE-WEN CHEN

OUTLINE OF THE PRESENTATION

- Introduction
- IKME
- Goals
- Overview
- Implementation
- Evaluation
- Screen Shots
- Conclusions
- Future Work

Introduction

- According to Institute of Higher Education Policy, 85% of four year colleges offer online courses.
- Many institutions offer similar courses, so there are probably hundreds of descriptions of similar topics on the Internet.
- Educational content is expensive to produce.
- Courses are not very flexible and cannot be easily re-purposed.

LEARNING OBJECTS

- Definition of Learning Object.
- Characteristics of a Learning Object.
  - Smaller Units of Learning.
  - Self Contained.
  - Reusable.
  - Can be aggregated.
  - Tagged with meta data.
- Reusable and very flexible when compared to a course as a learning unit.

IKME

- Intelligent Knowledge Management Environment (*IKME*) is an ongoing project at the *University of Kansas* aimed at assisting the *Defense Information Technology Test bed (DITT)/University After Next (UAN)* by providing an advanced reach-back capability for commanders, staff, and other users who have time-critical needs.
- A knowledge management environment would facilitate the creation of extensible and reusable learning objects that would lead to faster delivery of content to knowledge users.
- XML is used as the data format for publishing. Knowledge creators use the environment to create learning objects which are stored as XML documents.

Why XML?

- Advantages of XML
  - Structured - Capable of representing an object hierarchy.
  - Machine-readable and writeable.
  - Separates content from presentation.

- So, Learning Objects can be easily integrated into online courses.

Goals

- To help the user find the related learning object content for the creation of lesson objects and manuals.
- To use a Memory based approach for indexing as opposed to File based approach used for the earlier version.
- To incorporate Incremental Indexing into the similarity search instead of batch processing which required the algorithm to be re-run every time a new document is added to the collection.

Overview

- Problems with Earlier Version
  - File based Indexing.
  - Need to re-run the whole algorithm when a new document is added.

- My enhancements.
  - Memory based.
  - Incremental Indexing.

Formula

Formula:

$$Similarity(d1, d2) = \sum_{i=1}^{N} wt_{id1} * wt_{id2}$$

where

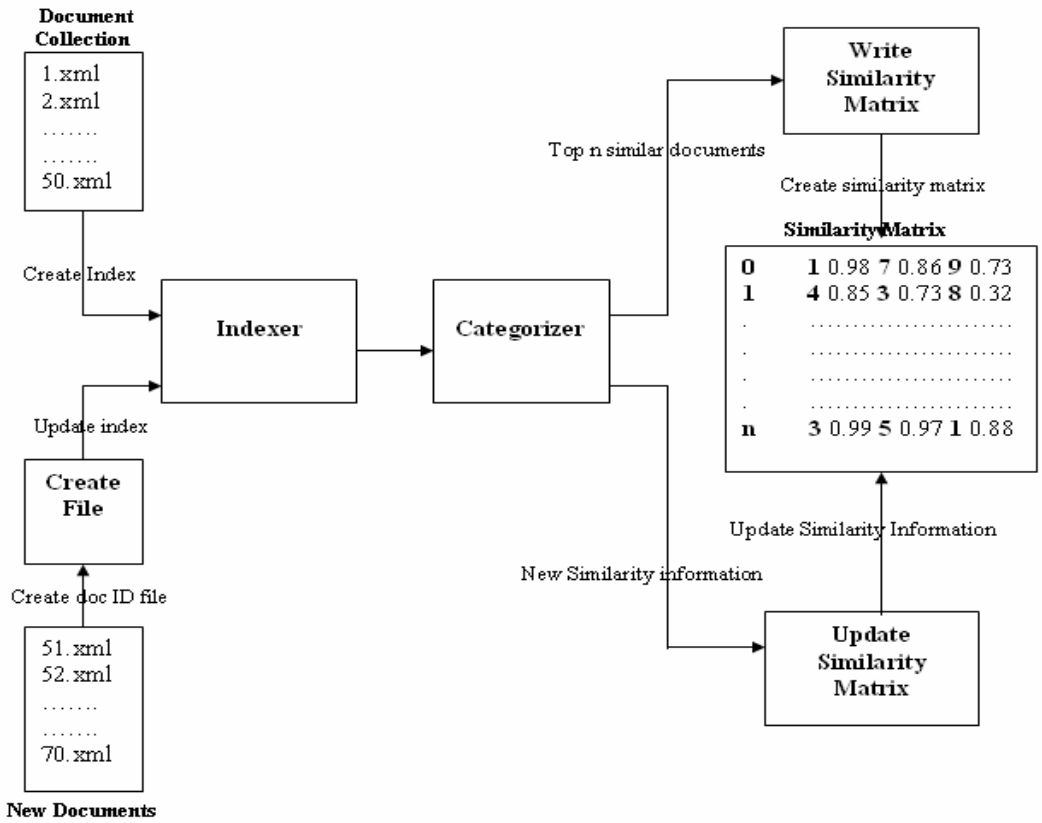
N = *number of tokens in the vocabulary*

$wt_{id1} = tf_{id1} * idf_i$

tf_{id1} = *(frequency of token i in d1 / total number of unique tokens in d1)*

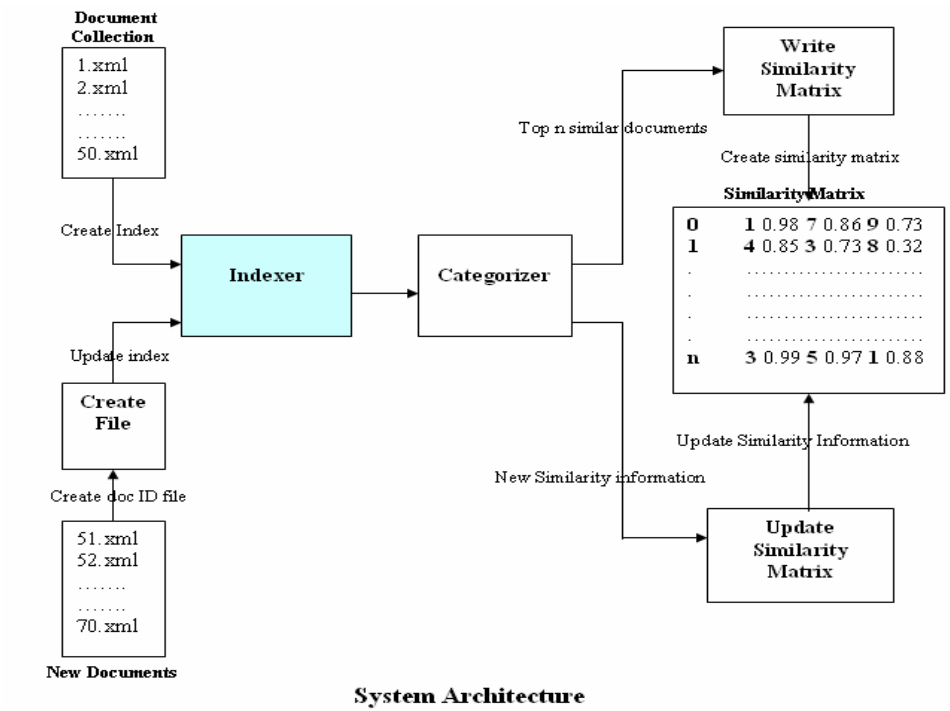
$idf_i = \log_2$ *(total number of documents/number of documents in which token i appeared)*

System Architecture



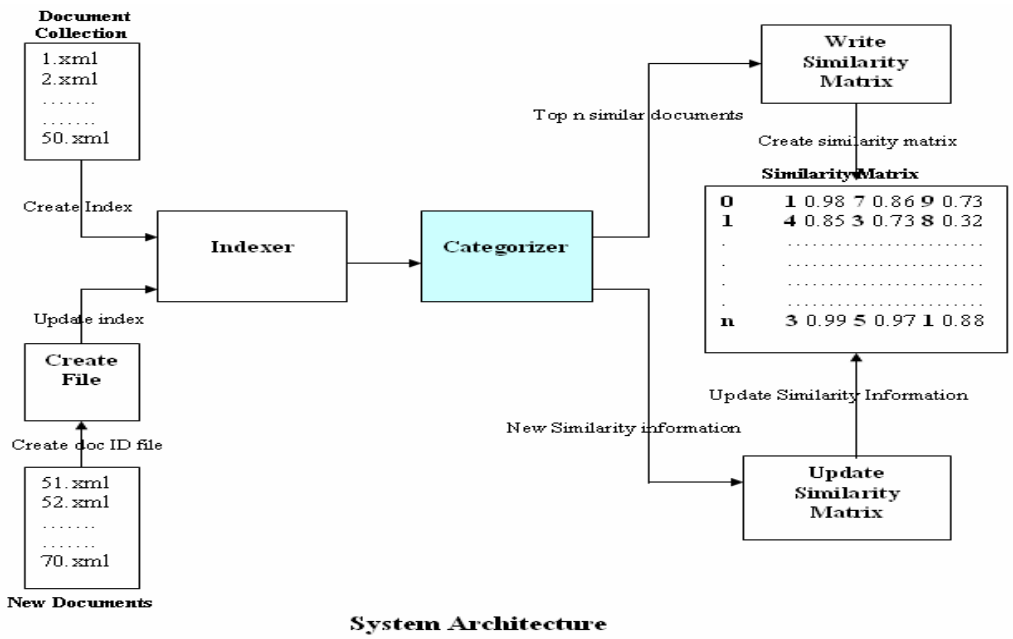
System Architecture

Indexer



Indexes the documents using standard vector space model (tf-idf) approach.

Categorizer



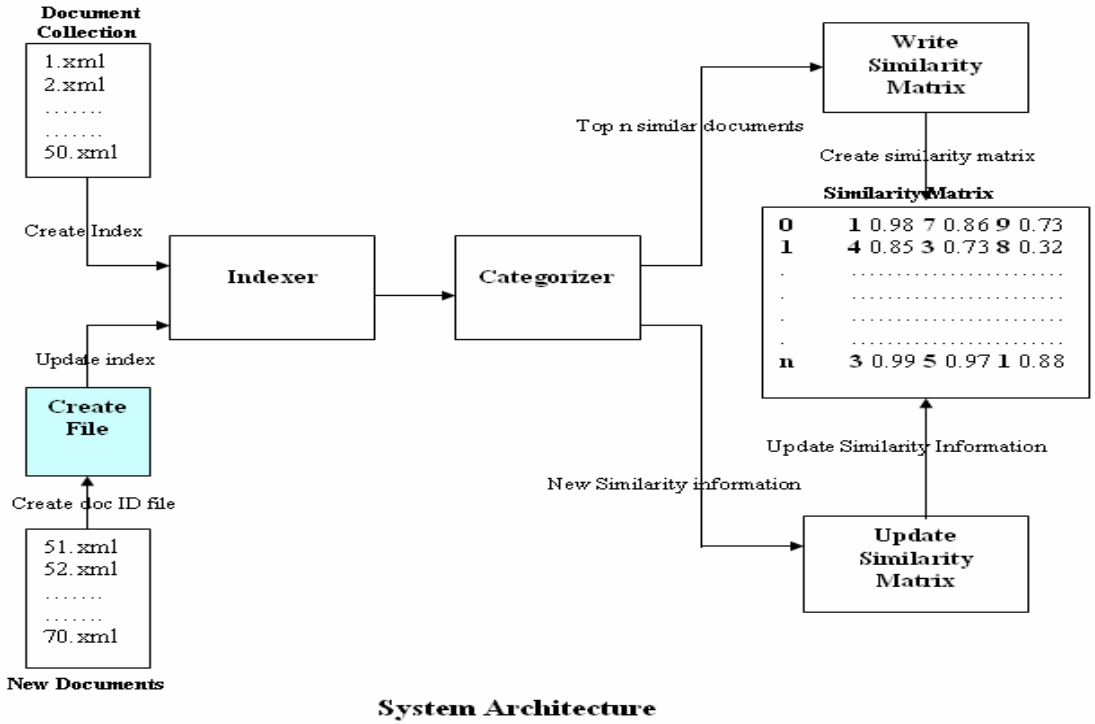
Returns the top ‘n’ similar documents to a given document in the decreasing order of similarities.

Categorizer Output

Example of Categorizer Output

Doc Id	Weight
10	0.981034
8	0.851032
6	0.752106
3	0.433210
11	0.121012
23	0.110012
1	0.090124
33	0.085291
42	0.067102
29	0.005128

CreateFile



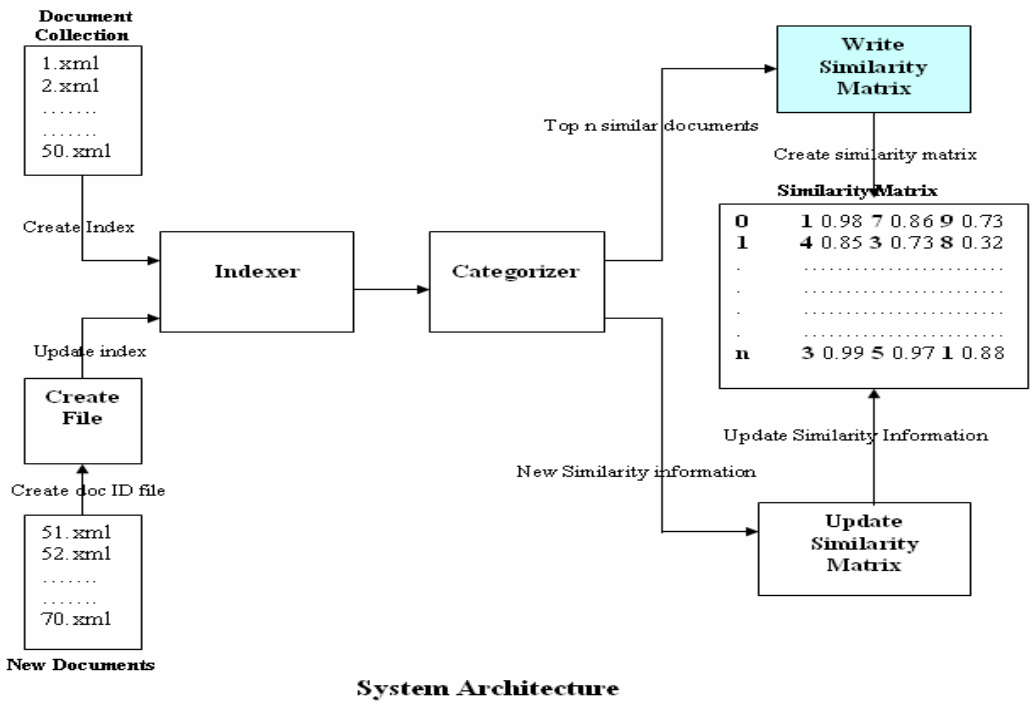
Creates a docID file which has document names and IDs. This file is needed for incremental indexing.

Sample docID file

Example of docID file

Doc Name	Doc Id
../inputfiles/urban_energy.xml	51
../inputfiles/urban_environment.xml	52
../inputfiles/urban_finance.xml	53
../inputfiles/urban_infrastructure.xml	54
../inputfiles/urban_population.xml	55
../inputfiles/urban_network.xml	56
../inputfiles/urban_satellite.xml	57
../inputfiles/urban_segments.xml	58
../inputfiles/urban_supersurface.xml	59
../inputfiles/urban_systems.xml	60

WriteSimilarity

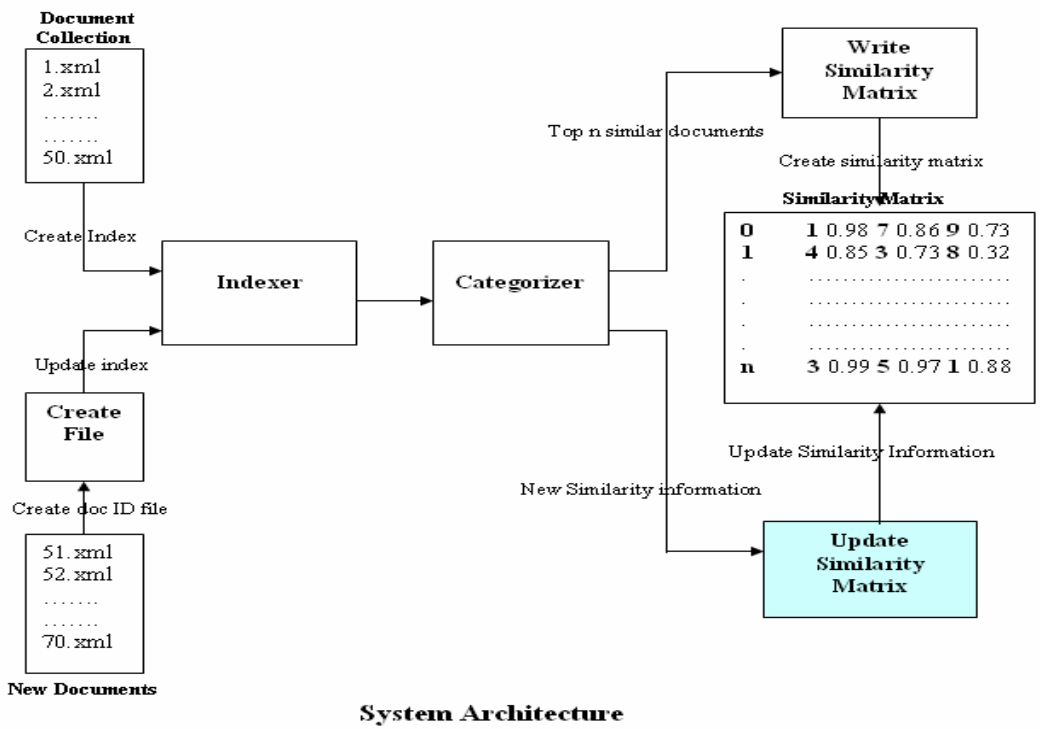


Generates the Similarity Matrix.

Format of Similarity Matrix

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	31	0.231273	13	0.214116	46	0.078542	5	0.058139	7	0.052780
2	7	0.537214	13	0.448968	9	0.226499	46	0.223909	25	0.188427
3	49	0.128824	17	0.128532	13	0.092650	12	0.080430	50	0.075375
...	...	.....	...	.....	...	.....	...	.....	...	.....
...	...	.....	...	.....	...	.....	...	.....	...	.....
49	47	0.478480	7	0.405528	42	0.300528	42	0.269646	43	0.253155
50	12	0.248636	47	0.188142	46	0.180350	7	0.158472	48	0.134855

UpdateSimilarity



System Architecture

To update the similarity information, because of the newly added documents to the collection.

STEP 1: Adding the new record(s) to the Similarity Matrix

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	31	0.231273	13	0.214116	46	0.078542	5	0.058139	7	0.052780
2	7	0.537214	13	0.448968	9	0.226499	46	0.223909	25	0.188427
3	49	0.128824	17	0.128532	13	0.092650	12	0.080430	50	0.075375
...	...	.....	...	.....	...	.....	...	.....	...	.....
...	...	.....	...	.....	...	.....	...	.....	...	.....
49	47	0.478480	7	0.405528	42	0.300528	42	0.269646	43	0.253155
50	12	0.248636	47	0.188142	46	0.180350	7	0.158472	48	0.134855
51	1	0.422716	49	0.412513	31	0.331092	42	0.290143	28	0.271103
52	2	0.610218	33	0.531274	41	0.442091	27	0.301208	8	0.200172
53	36	0.312082	3	0.229142	24	0.220081	46	0.193842	5	0.164032

STEP 2: Updating the corresponding records' similarity information

51	1	0.422716	49	0.412513	53	0.331092	42	0.290143	28	0.271103

Similarity (1,51) = 0.422716

Similarity (49,51) = 0.412513

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	31	0.231273	13	0.214116	46	0.078542	5	0.058139	7	0.052780
2	7	0.537214	13	0.448968	9	0.226499	46	0.223909	25	0.188427
3	49	0.128824	17	0.128532	13	0.092650	12	0.080430	50	0.075375
...	...	.....	...	.....	...	.....	...	.....	...	.....
...	...	.....	...	.....	...	.....	...	.....	...	.....
49	47	0.478480	7	0.405528	42	0.300528	42	0.269646	43	0.253155
50	12	0.248636	47	0.188142	46	0.180350	7	0.158472	48	0.134855
51	1	0.422716	49	0.412513	31	0.331092	42	0.290143	28	0.271103
52	2	0.610218	33	0.531274	41	0.442091	27	0.301208	8	0.200172
53	36	0.312082	3	0.229142	24	0.220081	46	0.193842	5	0.164032

Updating Document 1's Similarity Information

OLD RECORD OF DOC 1

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	31	0.231273	13	0.214116	46	0.078542	5	0.058139	7	0.052780

With the addition of new document 51 with Similarity (1,51) = 0.422716

NEW RECORD OF DOC 1

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	51	0.422716	31	0.231273	13	0.214116	46	0.078542	5	0.058139

Updating Document 49's Similarity Information

OLD RECORD OF DOC 49

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
49	47	0.478480	7	0.405528	42	0.300528	42	0.269646	43	0.253155

With the addition of new document 51 with Similarity (49,51) = 0.412513

NEW RECORD OF DOC 49

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
49	47	0.478480	51	0.412513	7	0.405528	42	0.300528	42	0.269646

Updating the corresponding records' similarity information (contd')

52	2	0.610218	33	0.531274	41	0.442091	27	0.301208	8	0.200172

Similarity (2,52) = 0.610218

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	31	0.231273	13	0.214116	46	0.078542	5	0.058139	7	0.052780
2	7	0.537214	13	0.448968	9	0.226499	46	0.223909	25	0.188427
3	49	0.128824	17	0.128532	13	0.092650	12	0.080430	50	0.075375
...	...	.....	...	.....	...	.....	...	.....	...	.....
...	...	.....	...	.....	...	.....	...	.....	...	.....
49	47	0.478480	7	0.405528	42	0.300528	42	0.269646	43	0.253155
50	12	0.248636	47	0.188142	46	0.180350	7	0.158472	48	0.134855
51	1	0.422716	49	0.412513	31	0.331092	42	0.290143	28	0.271103
52	2	0.610218	33	0.531274	41	0.442091	27	0.301208	8	0.200172
53	36	0.312082	3	0.229142	24	0.220081	46	0.193842	5	0.164032

New Similarity Matrix

Doc Id	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight	Doc Id	Weight
1	51	0.422716	31	0.231273	13	0.214116	46	0.078542	5	0.058139
2	52	0.610218	7	0.537214	13	0.448968	9	0.226499	46	0.223909
3	53	0.229142	49	0.128824	17	0.128532	13	0.092650	12	0.080430
...	...	.....	...	.....	...	.....	...	.....	...	.....
...	...	.....	...	.....	...	.....	...	.....	...	.....
49	47	0.478480	51	0.412513	7	0.405528	42	0.300528	42	0.269646
50	12	0.248636	47	0.188142	46	0.180350	7	0.158472	48	0.134855
51	1	0.422716	49	0.412513	31	0.331092	42	0.290143	28	0.271103
52	2	0.610218	33	0.531274	41	0.442091	27	0.301208	8	0.200172
53	36	0.312082	3	0.229142	24	0.220081	46	0.193842	5	0.164032

Evaluation

- <u>Incremental Updation of Similarity Information:</u>

Number of Documents	New Version Time (in seconds)	Earlier Version Time (in seconds)
54	26 (7.31u, 12.28s)	52
72 (18 documents added)	11 (2.48u, 3.98s)	62
172 (100 documents added)	52 (14.13u, 23.10s)	220

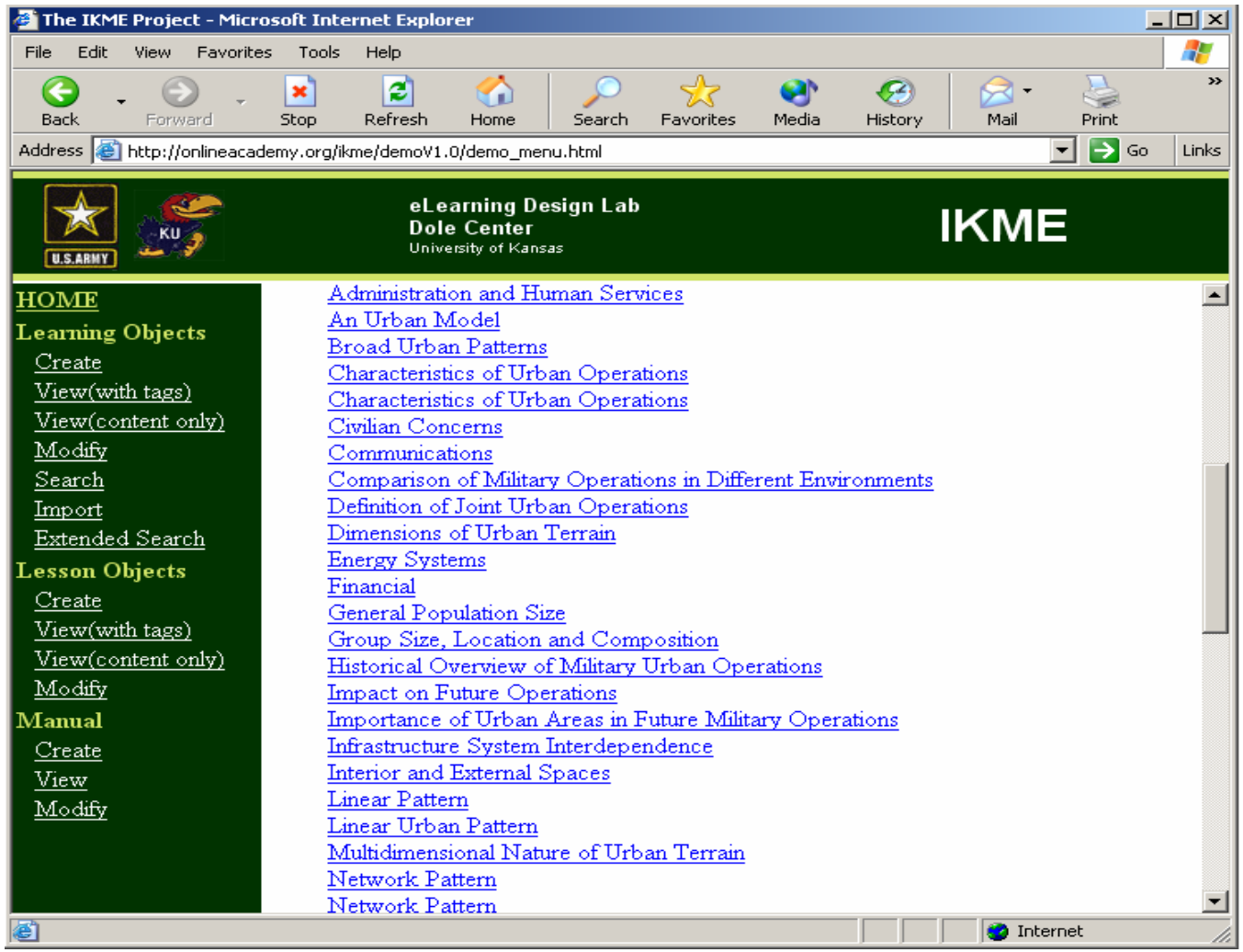
For 500 documents (450 added to an initial collection of 50), it took around 243 seconds (68.34u, 124.36s) for the new version.

Evaluation (cont'd)

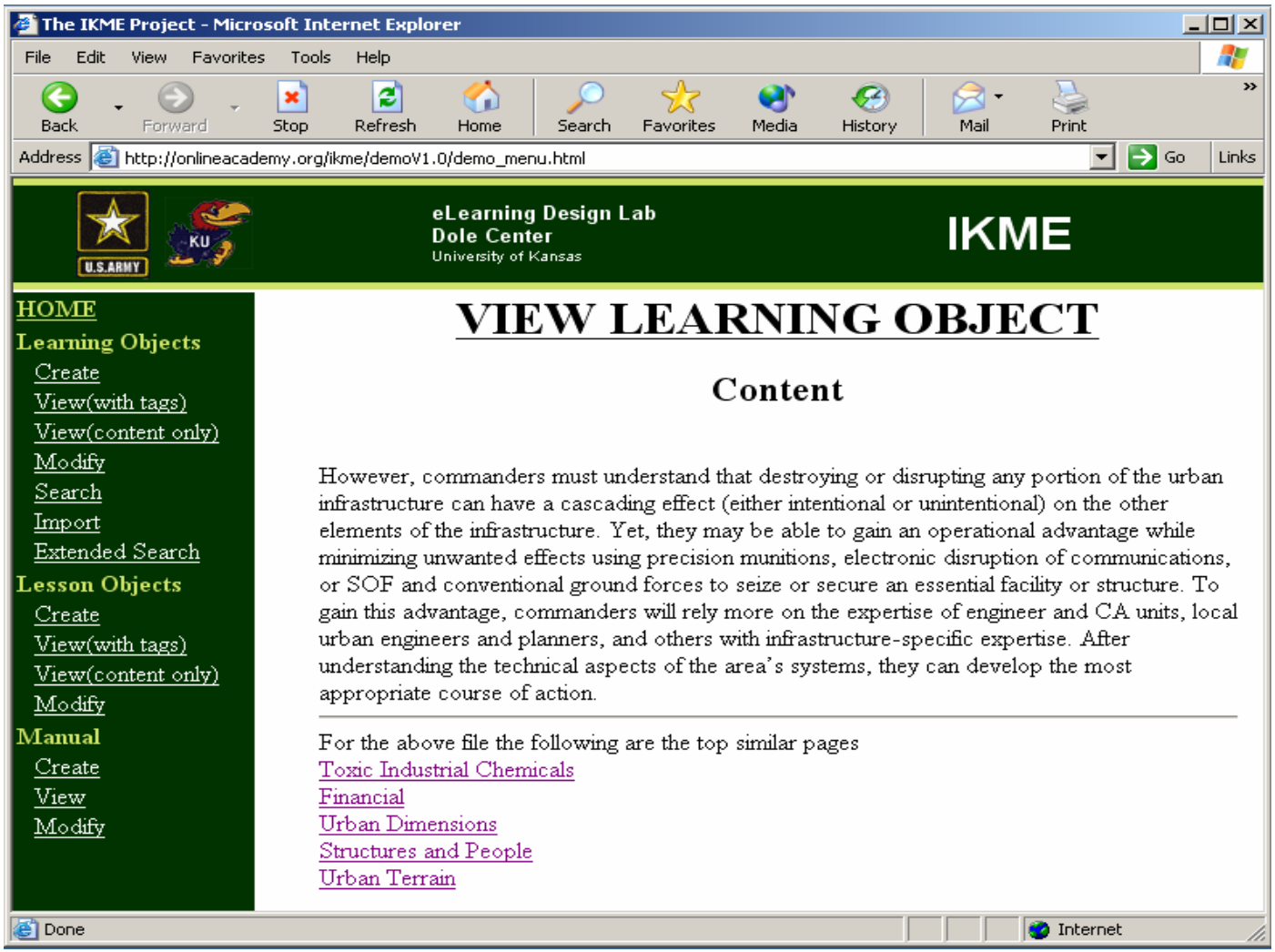
- Calculating Similarity Information from the scratch:

Number of Documents	New Version Time (in seconds)	Earlier Version Time (in seconds)
54	26 (6.98u, 12.09s)	52
72	37 (9.418u, 15.85 s)	62
172	91 (24.21u, 38.09s)	220

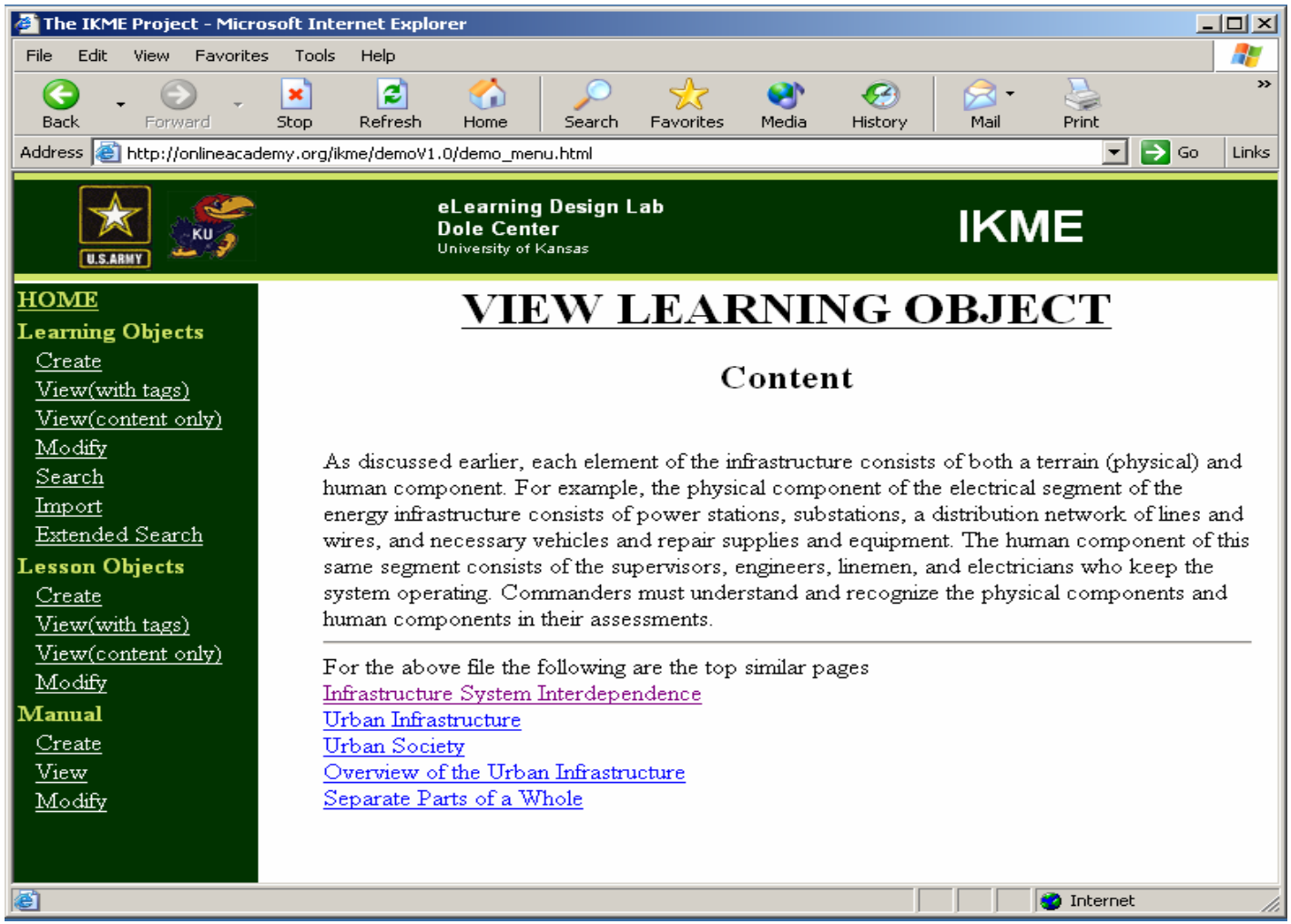
Screenshots – List of Learning Objects



Displaying the top 5 similar learning objects at the bottom



Screenshots – Displaying the top similar learning object



Conclusions

- The primary goal of incorporating Incremental Indexing into the similarity search has been achieved.
- The algorithm need not be re-run even if a single document is added to the collection. Only the required parts of the index and the similarity matrix are updated.
- This will provide a faster way to search for similar learning objects and help the educators in creating new lessons using existing learning objects rapidly and inexpensively.

Future Work

- Investigating similarity formula (i e., weighting different fields differently when calculating the match between the objects)
- Learning best differential weighting scheme.

References

- All about Learning Objects
  http://www.eduworks.com/LOTT/tutorial/learningobjects.html
- Learning Objects 101 : A Primer for Neophytes
  http://online.bcit.ca/sidebars/02november/inside-out-1.htm
- Introducing Reusable Learning Objects
  http://media.wiley.com/product_data/excerpt/56/07879649/0787964956.pdf
- "Automatically Identifying Related Learning Objects" Mahesh Vulpala, Masters Project. University of Kansas 2003.

Questions

- ???