TF-ICF – A New Document Encoding Scheme for Mining Data Streams

Dr. Yu (Cathy) Jiao

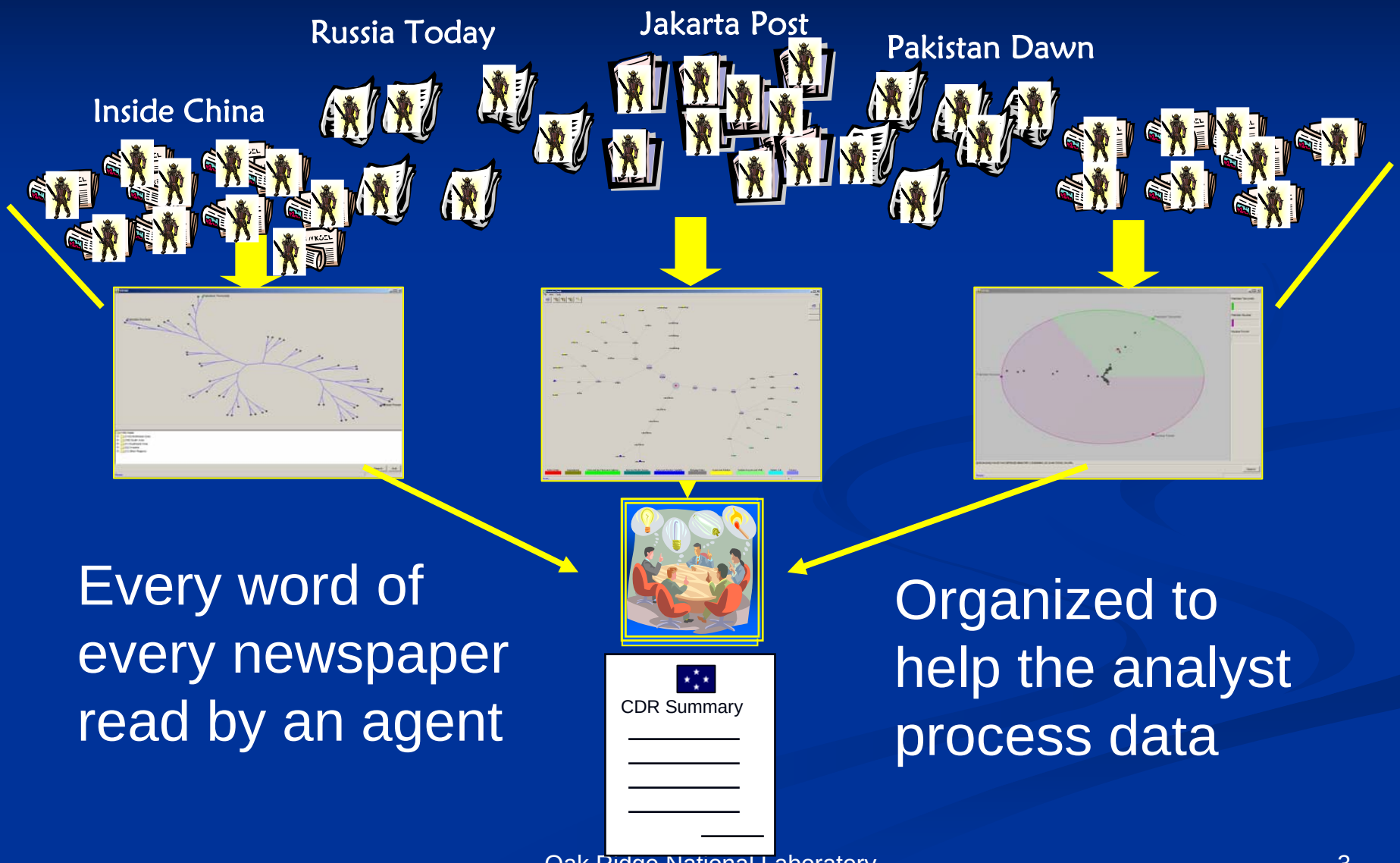
Applied Software Engineering Research Group

Oak Ridge National Laboratory

Agenda

- Introduction
- Background
- TF-ICF – A New Document Encoding Scheme
- Piranha – A TF-ICF Based KDD Tool
- Conclusions and Future Directions

Introduction



Every word of every newspaper read by an agent

Organized to help the analyst process data

Introduction

- Mining data streams is concerned with extracting knowledge structures represented in models and patterns in continuous streams of information.

- Major Challenges:
  - One-pass through data points
  - Memory constraints

Background – Document Clustering

Inverse Document Frequency

$$W_{ij} = \log_2(f_{ij}+1) * \log_2\left(\frac{N}{n}\right)$$

$$d(x_i, x_j) = \left(\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2\right)^{1/2}$$

$$O(n^2 \log n)$$

Vector Space Model

	Doc 1	Doc 2	Doc 3
Army	1	0	0
Sensor	1	1	1
Technology	1	1	0
Help	1	0	0
Find	1	0	0
Improvise	1	0	0
Explosive	1	0	1
Device	1	0	1
ORNL	0	1	0
develop	0	1	1
homeland	0	1	1
Defense	0	1	1
Mitre	0	0	1
won	0	0	1
contract	0	0	1

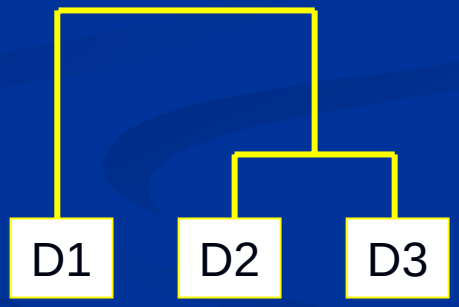
Words to Documents

Similarity Matrix

	Doc 1	Doc 2	Doc 3
Doc 1	100%	17%	21%
Doc 2		100%	36%
Doc 3			100%

Documents to Documents

Cluster Analysis



Most similar documents

Cannot work with data streams!
Need a new encoding scheme!

The TF-ICF Encoding Scheme

- Term Frequency – Inverse Corpus Frequency Encoding Scheme

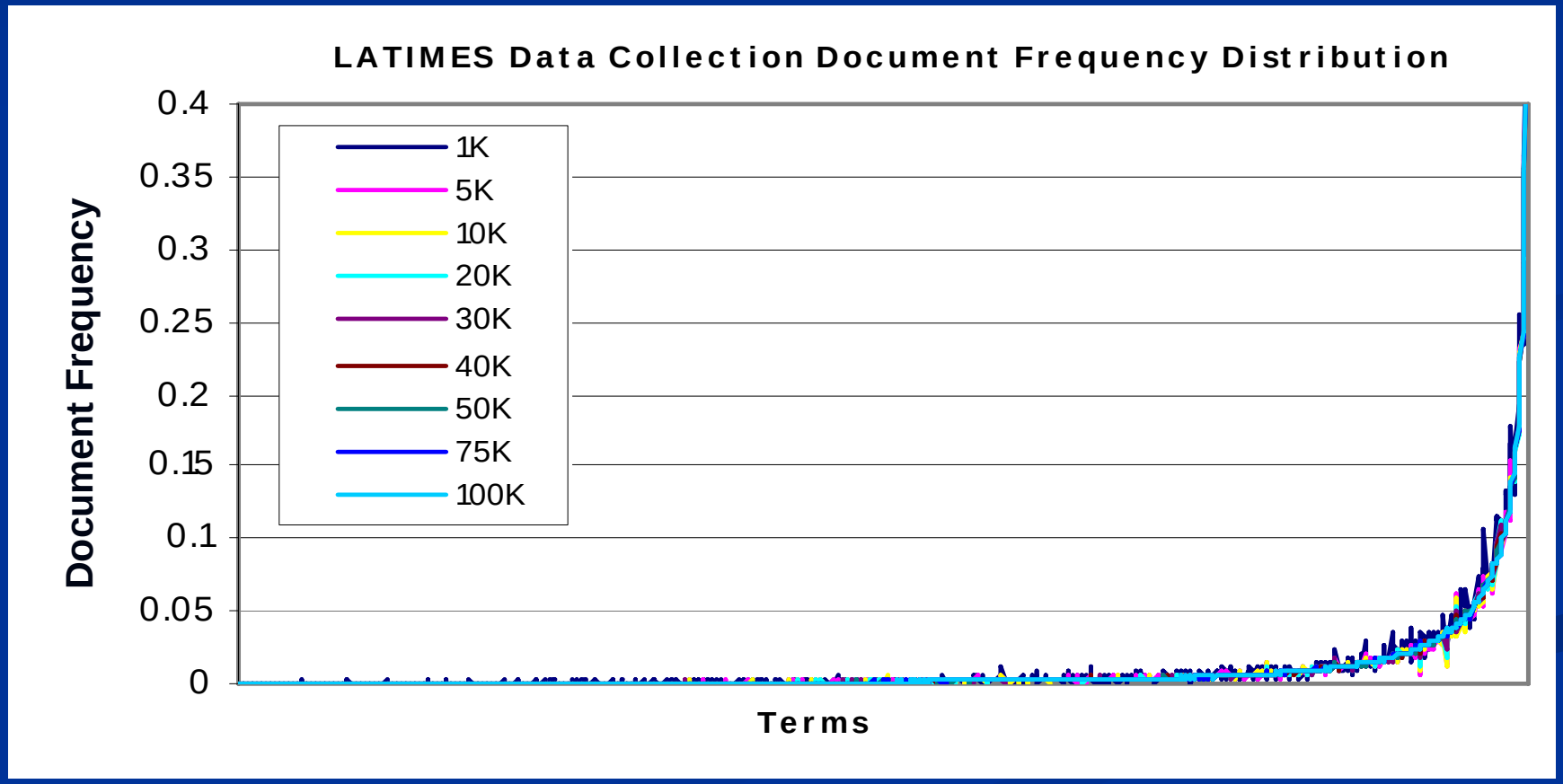
- Objective
  - Remove the dependency on document frequency.

The TF-ICF Encoding Scheme

- Questions
  - Can we use the document frequency distribution of a smaller data set to approximate that of a larger data set?
  - Can we use the document frequency of a known corpus to approximate that of an unknown data stream?
  - Can we compose a corpus that covers most of the commonly used words?

The TF-ICF Encoding Scheme

- Can we use the document frequency distribution of a smaller data set to approximate that of a larger data set?



The TF-ICF Encoding Scheme

- Can we use the document frequency of a known corpus to approximate that of an unknown data stream?

Reference Corpora	Document Count
TREC-AP	239,055
TREC-FBIS	130,392
TREC-LATIMES	127,732
TREC-SJM	73,748
TREC-WSJ	161,576
RSS News Feeds	202,991
Total	**935,494**

Statistics:

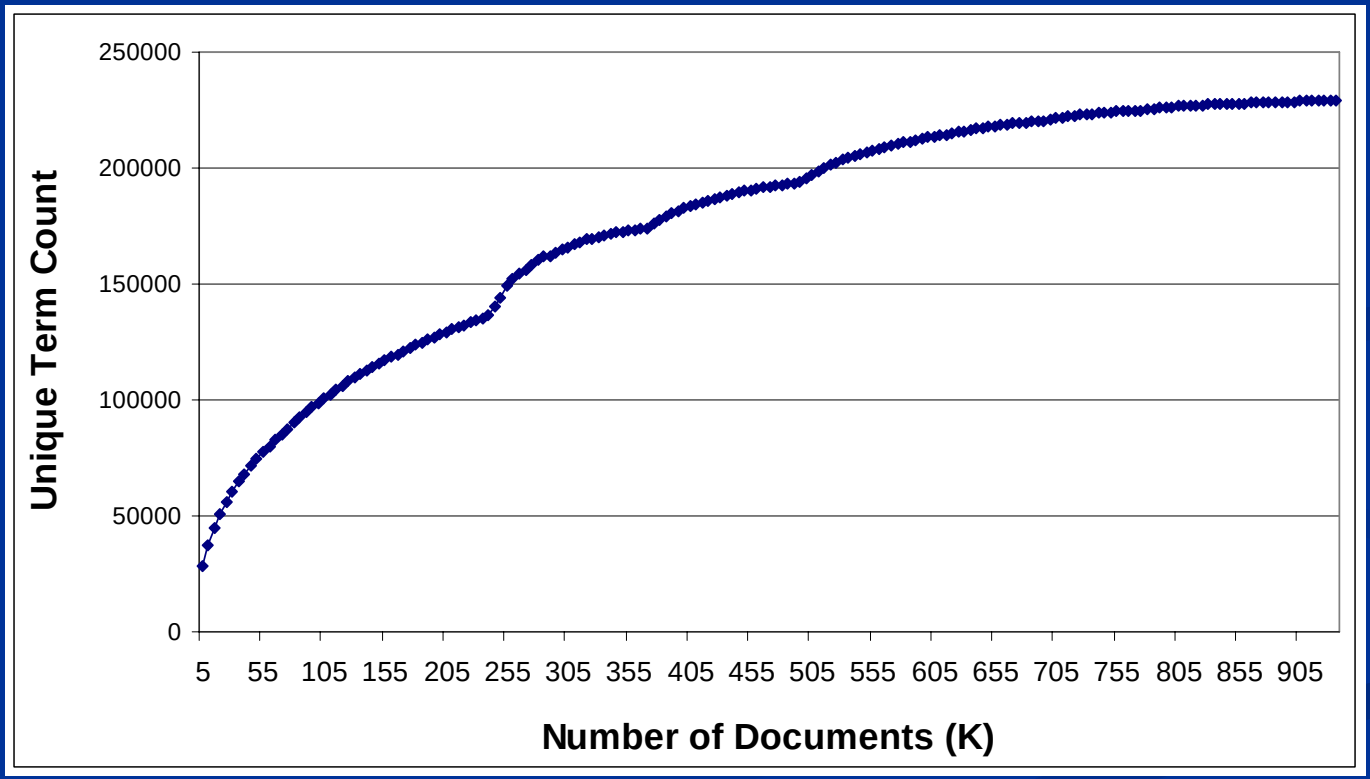
- # of Unique Terms: 229,023
- # of Common Terms: 28,872
- Average Standard Deviation of the Document Frequency (n/N) of the Six Corpora: < 0.01%

$$W_{ij} = \log_2\left(f_{ij}+1\right) * \log_2\left(\frac{C+1}{c+1}\right)$$

Inverse Corpus Frequency

The TF-ICF Encoding Scheme

- Can we compose a corpus that covers most of the commonly used words?



The TF-ICF Encoding Scheme

- Meet the Challenges
  - One-pass through data points
  - Memory constraints
- Moreover
  - Reduces the computational complexity of encoding N dynamic documents from $O(N^2)$ to $O(N)$.
  - Provides the technical foundation for parallel data stream clustering.

TF-ICF Performance Evaluation

Term Weighting Schemes

Name	Term Weighting Scheme
TF-IDF	$w_{ij} = \log(f_{ij}) \times \log(N / n_j)$
MI	$w_{ij} = \log \dfrac{\frac{f_{ij}}{N}}{\frac{\sum_{i=1}^{N} f_{ij}}{N} \times \frac{\sum_{j=1}^{M} f_{ij}}{N}}$
ATC	$w_{ij} = \dfrac{\left(0.5 + 0.5 \times \frac{f_{ij}}{max_f}\right) \log\left(\frac{N}{n_j}\right)}{\sqrt{\sum_{i=1}^{N} \left[\left(0.5 + 0.5 \times \frac{f_{ij}}{max_f}\right) \log\left(\frac{N}{n_j}\right)\right]^2}}$
Okapi	$w_{ij} = \left(\dfrac{f_{ij}}{0.5 + 1.5 \times \frac{dl}{avg_dl} + f_{ij}}\right) \log\left(\dfrac{N - n_j + 0.5}{f_{ij} + 0.5}\right)$
LTU	$w_{ij} = \dfrac{(\log(f_{ij}) + 1.0) \log\left(\frac{N}{n_j}\right)}{0.8 + 0.2 \times \frac{dl}{avg_dl}}$

Test Data Sets

Data Set	# of Docs	# of Classes	Largest Class	Smallest Class
Reuters	2349	58	1041	1
SMART	3891	3	1460	1033
20 News	4650	12	399	385

Test Algorithms

Basic K-Means

Single Link (HAC)

Complete Link (HAC)

UPGMA (HAC)

TF-ICF Performance Evaluation

Similarity Measures

$$EuclideanDistance(X,Y) = \sqrt{\sum (x_i - y_i)^2}$$

$$CosineSimilarity(X,Y) = \frac{X \bullet Y}{\| X \| \times \| Y \|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \times \sqrt{\sum y_i^2}}$$

Performance Metrics

Let the correct classification of documents be $C_1 \ldots C_k$, where each C_i is a class. Let $\tilde{C}_1 \ldots \tilde{C}_l$ denotes the clusters.

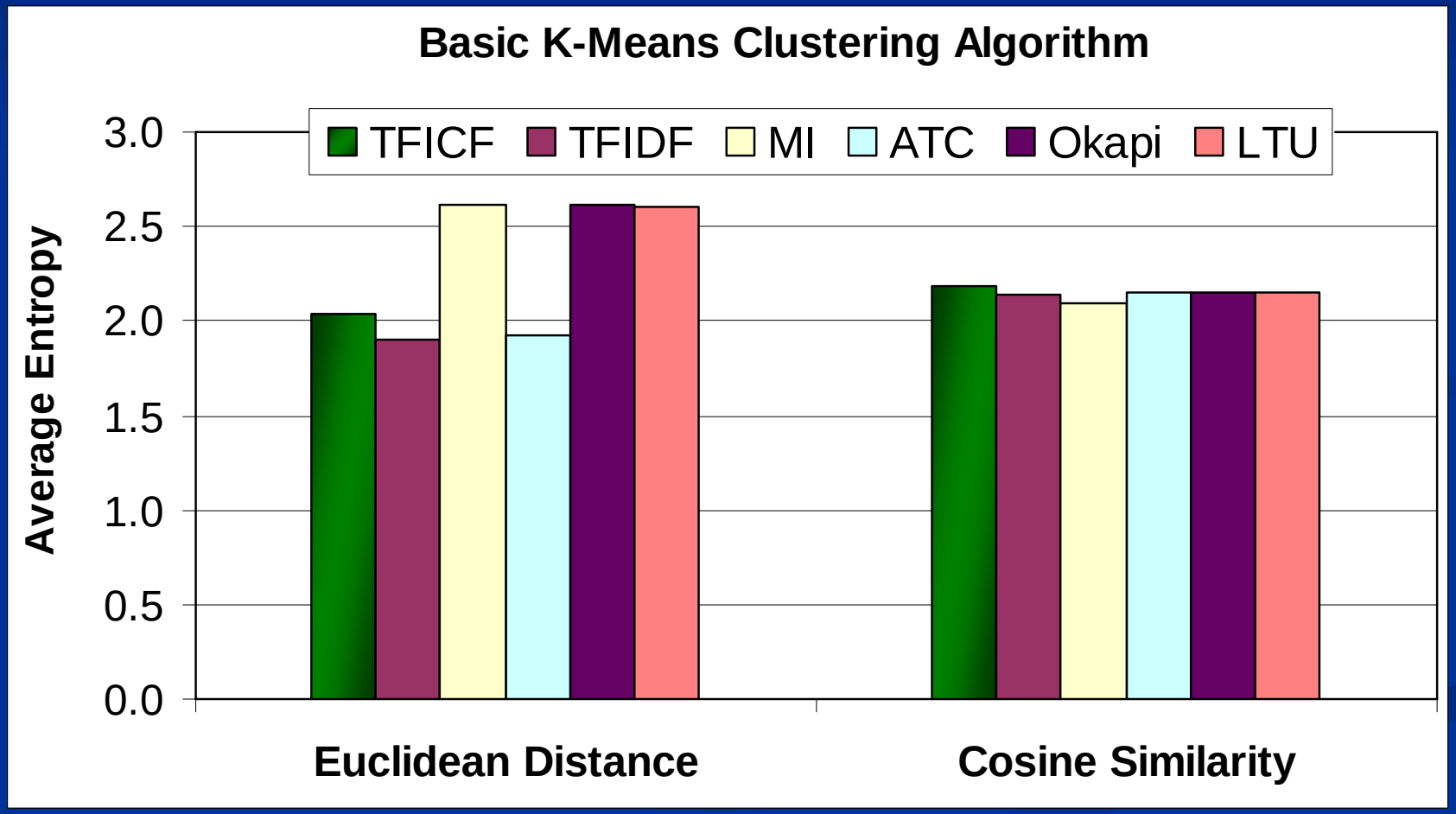
For Partitional Algorithms like the Basic K-Means, we use *Entropy*.

$$Entropy(\tilde{C}j) = \sum_{i=1}^{k} -\left(\frac{|C_i \cap \tilde{C}_j|}{|\tilde{C}_j|}\right) \log\left(\frac{|C_i \cap \tilde{C}_j|}{|\tilde{C}_j|}\right) \qquad Entropy = \sum_{j=1}^{k}\left(Entropy(\tilde{C}_j) \times \frac{|\tilde{C}_j|}{N}\right)$$

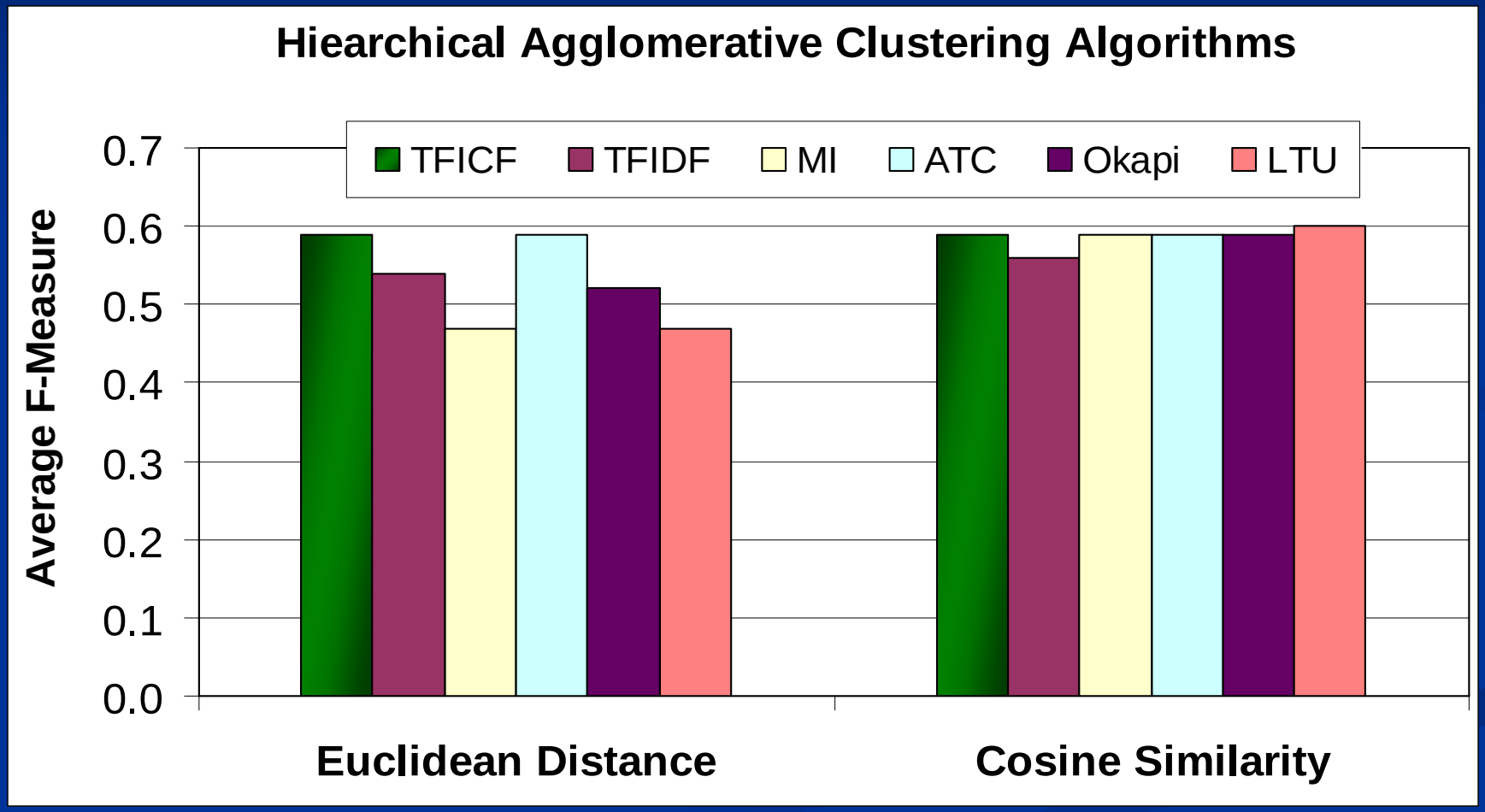
For Hierarchical Agglomerative Algorithms, we use *F-Measure*.

$$P_j = \frac{|C_i \cap \tilde{C}_j|}{|\tilde{C}_j|}, Rj = \frac{|C_i \cap \tilde{C}_j|}{|C_i|} \qquad F(C_i) = \max_{i=1}^{l} \frac{2P_j R_j}{P_j + R_j}$$

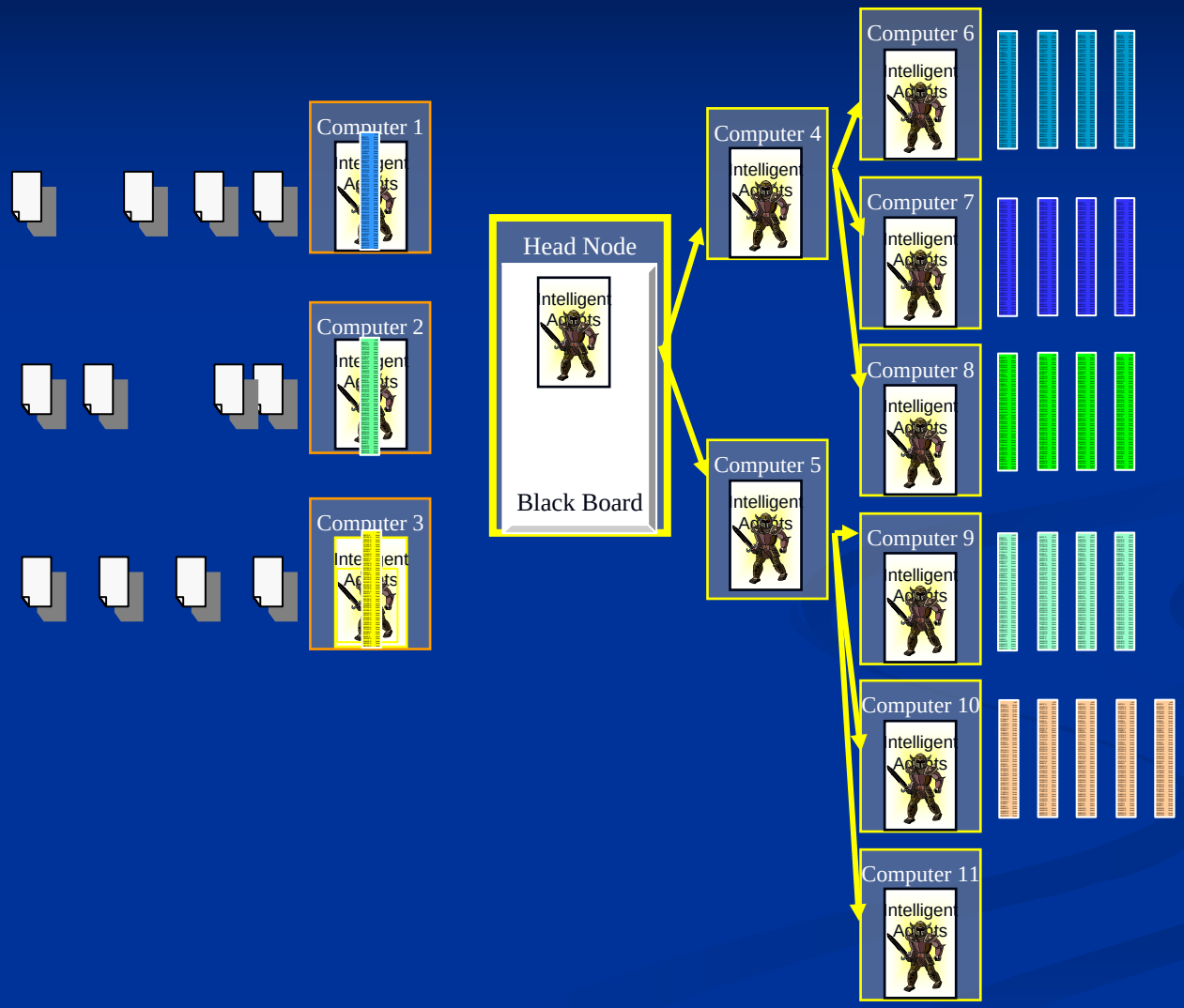
TF-ICF Performance Evaluation



TF-ICF Performance Evaluation



Piranha – A TF-ICF Based KDD Tool



Conclusions and Future Directions

$$W_{ij} = \log_2\left(f_{ij}+1\right) * \log_2\left(\frac{N}{n}\right)$$

Vector Space Model

	Doc 1	Doc 2	Doc 3
Army	1	0	0
Sensor	1	1	1
Technology	1	1	0
Help	1	0	0
Find	1	0	0
Improvise	1	0	0
Explosive	1	0	1
Device	1	0	1
ORNL	0	1	0
develop	0	1	1
homeland	0	1	1
Defense	0	1	1
Mitre	0	0	1
won	0	0	1
contract	0	0	1

Words to Documents

TF-ICF

$$d(x_i, x_j) = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2\right)^{1/2}$$

Dissimilarity Matrix

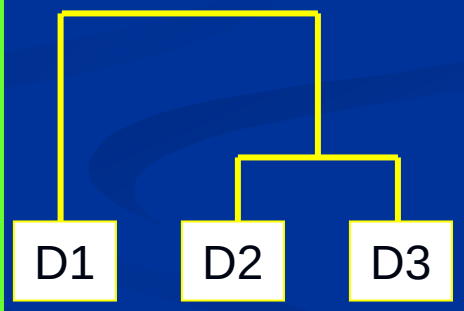
	Doc 1	Doc 2	Doc 3
Doc 1	100%	17%	21%
Doc 2		100%	36%
Doc 3			100%

Documents to Documents

Dynamic Dimensionality Reduction

$$O(n^2 \log n)$$

Cluster Analysis



Most similar documents

Parallel Algorithms

Conclusions and Future Directions

Every word of every newspaper read by an agent

Organized to help the analyst process data

Q & A

- Contact Information

Dr. Yu (Cathy) Jiao

jiaoy@ornl.gov