Hierarchical Document Clustering Using Frequent Itemsets

SDM ’03

Present：Yao-Min Huang

Date：2004年6月10日

Reference：Benjamin C.M. Fung 、Ke Wang、Martin Esterz

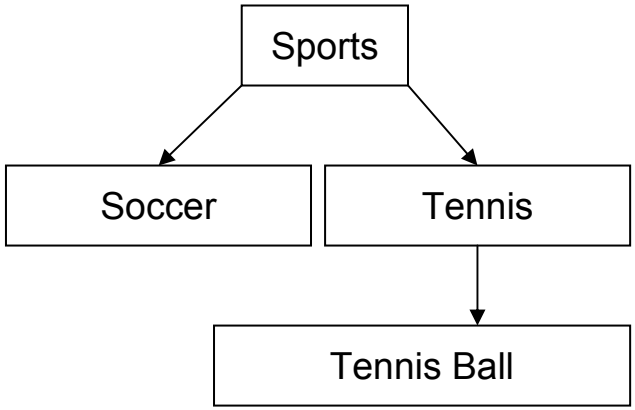
Hierarchical Document Clustering Using Frequent Itemsets

Outline

- What is hierarchical document clustering?
- Frequent Itemset Hierarchical Clustering (FIHC)
- Experimental Results
- Conclusions

Hierarchical Document Clustering

- Document Clustering
  - Automatic organization of documents into clusters so that documents within a cluster have high similarity in comparison to one another, but are very dissimilar to documents in other clusters.
- Approach
  - Agglomerative (Button-up)
  - Divisive (Top-down)
- Hierarchical Document Clustering:



Challenges in Hierarchical Document Clustering

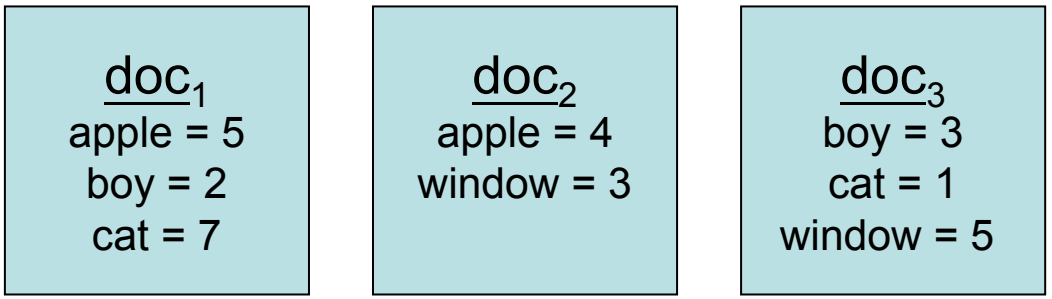
- High dimensionality.
- High volume of data.
- Consistently high clustering quality.
- Meaningful cluster description.

Preprocessing

- Remove stop words and Stemming.
- Construct vector model

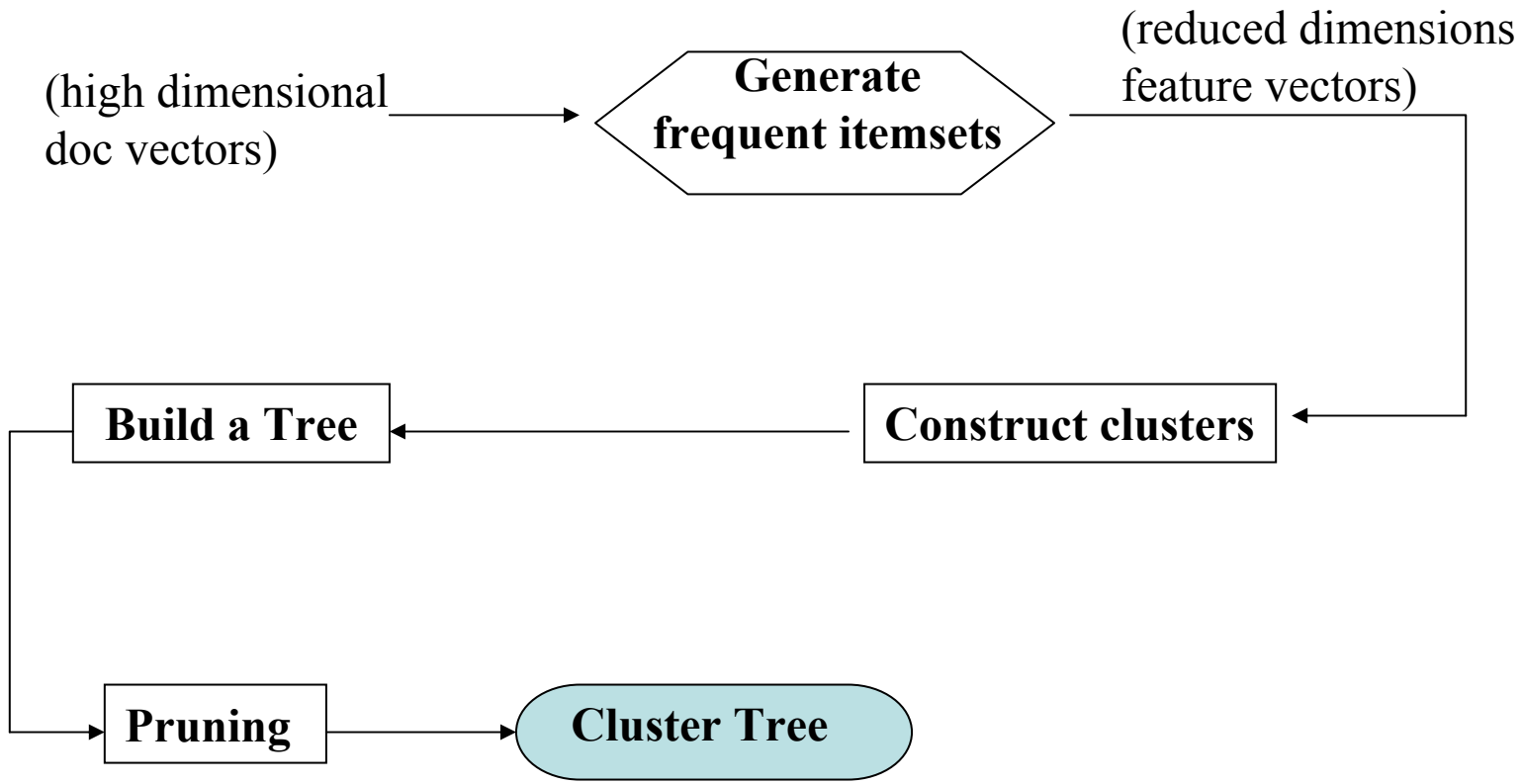
doc_i = (item $frequency_1$, if_2, if_3, …, if_m)

e.g.



	(apple,	boy,	cat,	window)	
doc_1 = (	5,	2,	7,	0	)
doc_2 = (	4,	0,	0,	3	)
doc_3 = (	0,	3,	1,	5	) ← **document vector**

Algorithm Overview of Our Method (FIHC)



Definition: Global Frequent Itemset

- A *global frequent itemset* refers to a set of items (words) that appear together in more than a user-specified fraction of the document set. (Minimal global support)

- A *global frequent item* refers to an item that belongs to some global frequent itemset, e.g., “apple”.

- The *global support* of an itemset is the percentage of documents containing the itemset.
  e.g. 7% of the documents contain both words.
  {apple, window} has global support 7%.

Reduced Dimensions Vector Model

- High dimensional vector model

(apple, boy, cat, window)

doc_1 = (5, 2, 1, 1)

doc_2 = (4, 0, 0, 3)

doc_3 = (0, 3, 1, 5) ← ***document vector***

doc_4 = (8, 0, 2, 0)

doc_5 = (5, 0, 0, 3)

- Suppose we set the minimum support to 60%. The global frequent itemsets are: {apple}, {cat}, {window}, {apple, window}
- Store the frequencies only for global frequent items.

(apple, cat, window)

doc_1 = (5, 1, 1)

doc_2 = (4, 0, 3) ← ***feature vector***

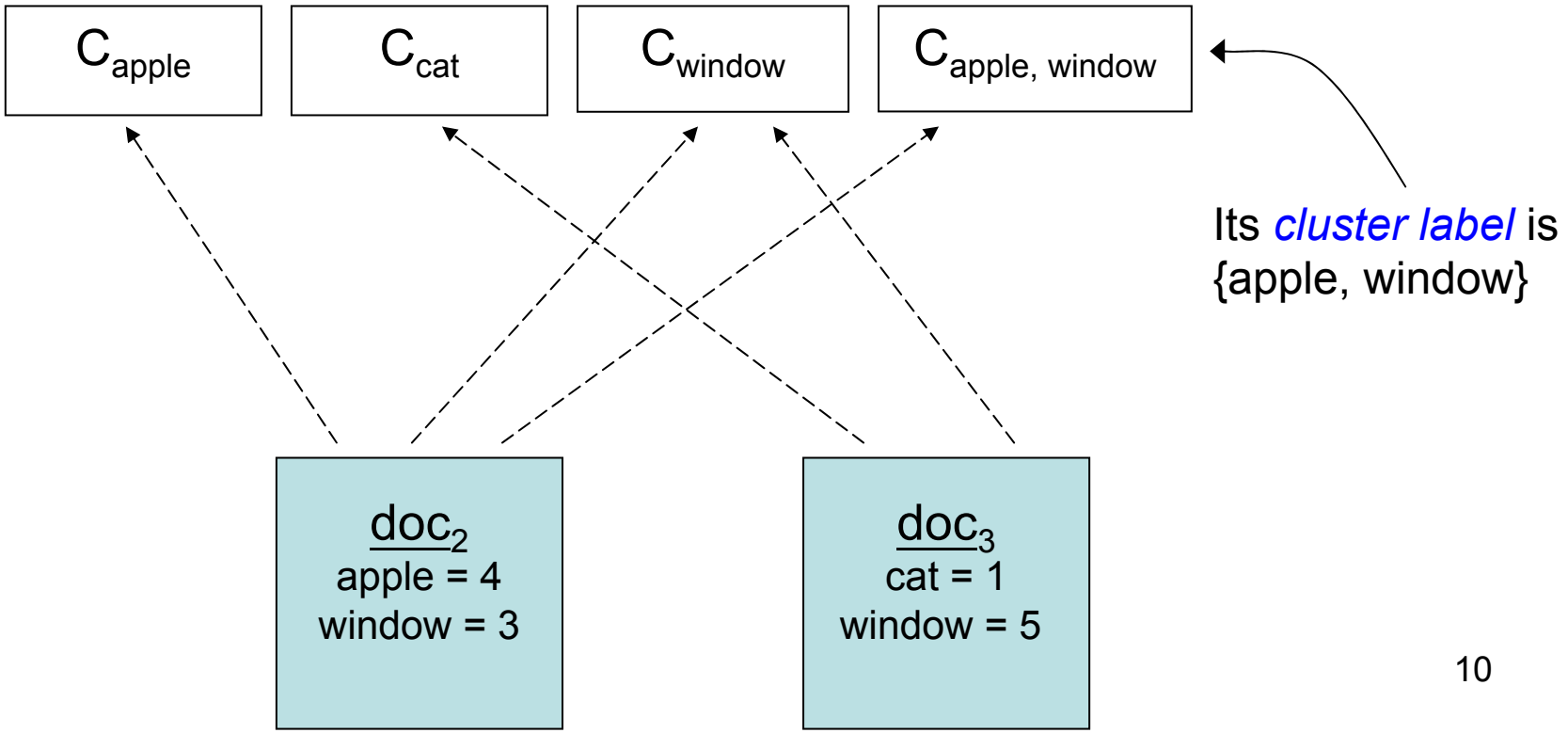
Intuition

- Why do we think frequent itemsets work for clustering?
  - different clusters share few frequent itemsets.
  - captures the relationship (or combination) among the words.
- Frequent itemsets → combination of words.
  - Ex. "apple" Topic: Fruits

    "apple, window" Topic: Computer

Construct Initial Clusters

- Construct a cluster for each global frequent itemset.
  Global frequent itemsets = {apple}, {cat}, {window}, {apple, window}
- All documents containing this itemset are included in the same cluster.



Making Clusters Disjoint

- Assign a document to the “best” initial cluster.

Intuitively, a cluster C_i is good for a document doc_j if there are many global frequent items in doc_j that appear in many documents in C_i.

Cluster Frequent Items

- A global frequent item is *cluster frequent* in a cluster C_i if the item is contained in some minimum fraction of documents in C_i.
- A cluster support of an item in Ci is the percentage of the doc. In Ci that contain the item.
- Suppose we set the minimum cluster support to 60%.

C_{apple}

	(apple,	cat,	window)
doc_1 = (	5,	1,	1)
doc_2 = (	4,	0,	3)
doc_4 = (	8,	2,	0)
doc_5 = (	5,	0,	3)

C_{apple}	
Item	**Cluster Support**
apple	100%
cat	50%
window	75%

{apple} and {window} are cluster frequent items.

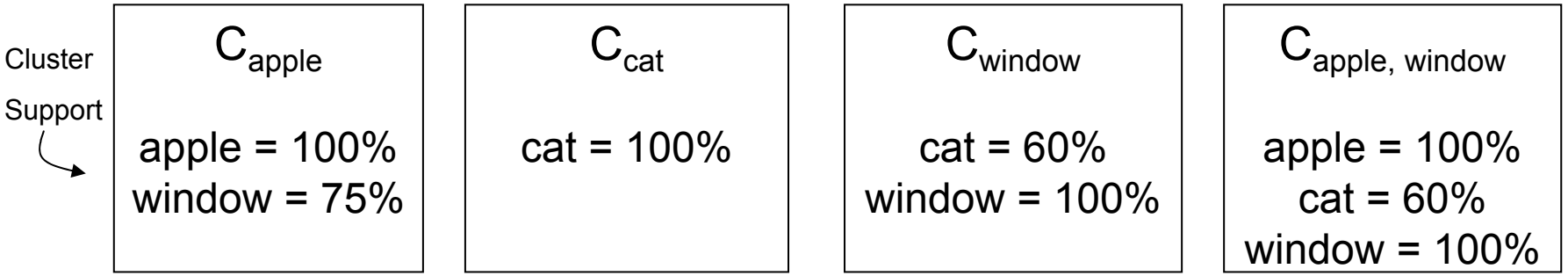
Score Function

- Assign each doc_j to the initial cluster C_i that has the highest $score_i$:

$$Score(C_i \leftarrow doc_j) = [\sum_x n(x) * cluster\_support(x)] - [\sum_{x'} n(x') * global\_support(x')]$$

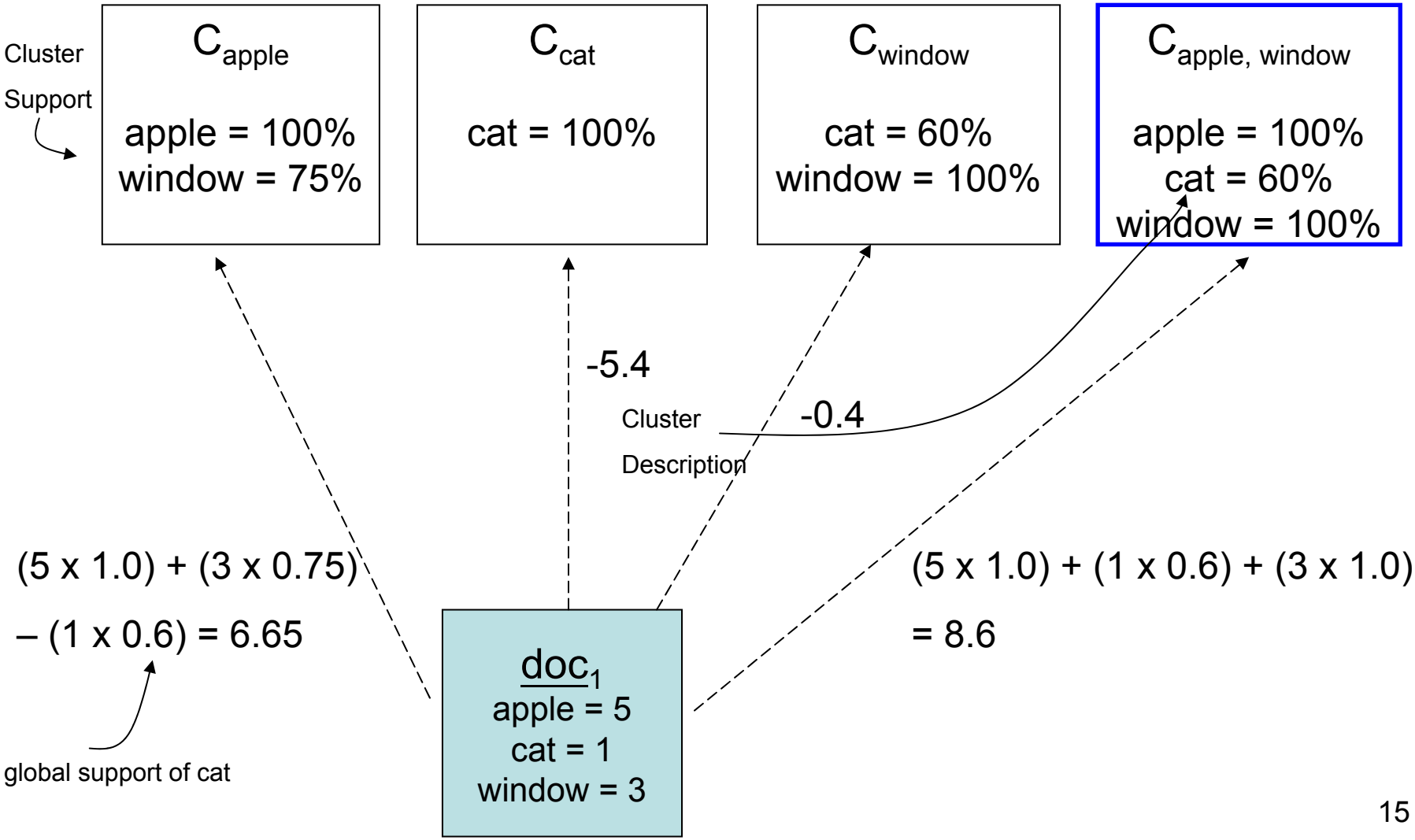
- x represents a global frequent item in doc_j and the item is also cluster frequent in C_i.
- x' represents a global frequent item in doc_j but the item is not cluster frequent in C_i.
- n(x) is the frequency of x in the feature vector of doc_j.
- n(x') is the frequency of x' in the feature vector of doc_j.

Score Function (Example)



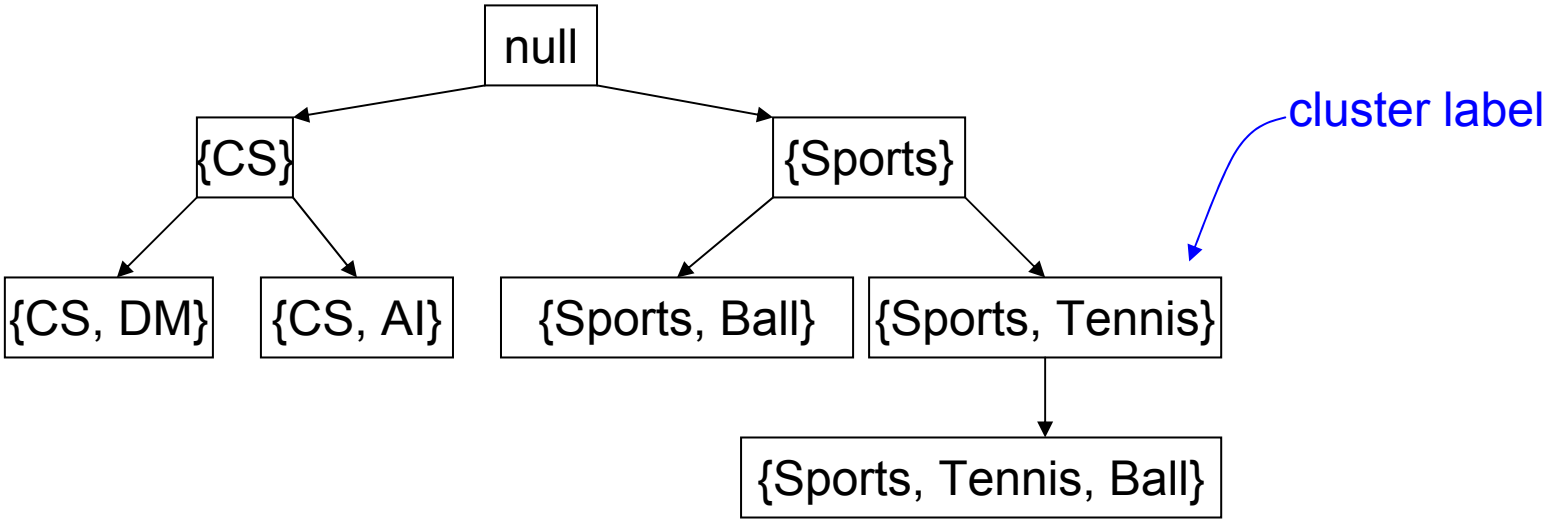
$\underline{doc_1}$
apple = 5
cat = 1
window = 3

Score Function (Example)



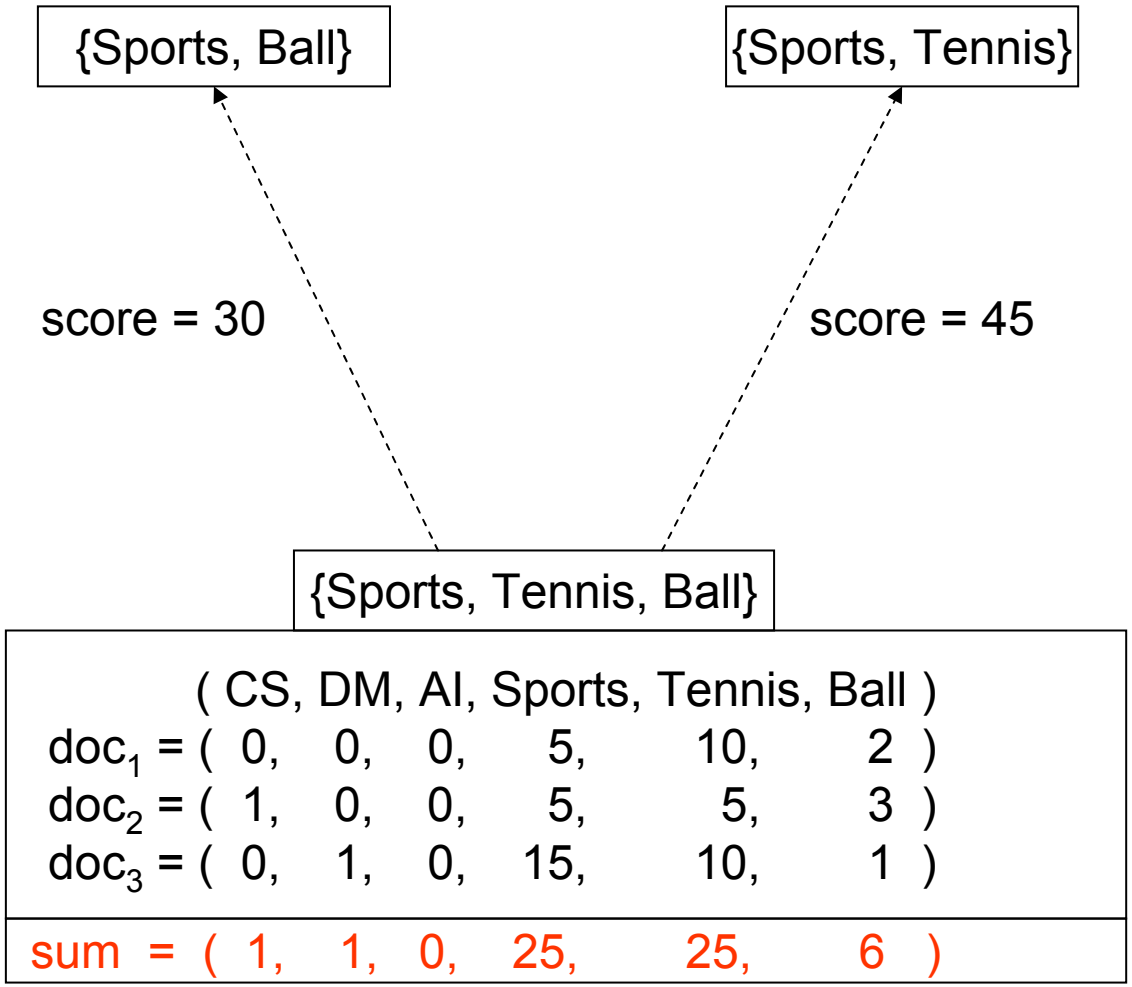
Tree Construction

- Put the more specific clusters at the bottom of the tree.
- Put the more general clusters at the top of the tree.



- Build a tree from bottom-up by choosing a parent for each cluster (start from the cluster with the largest number of items in its cluster label).

Choose a Parent Cluster (example)

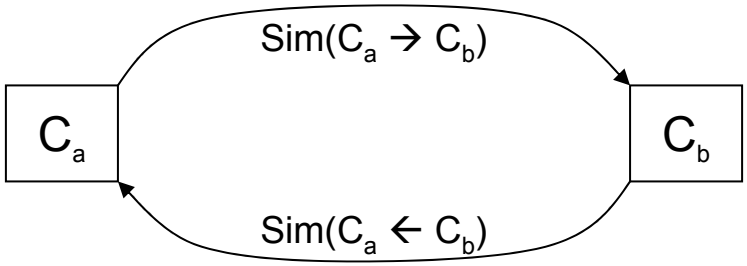


Prune Cluster Tree

- Why do we want to prune the tree?
  - Remove overly specific child clusters.
  - Documents of the same class (topic) are likely to be distributed over different subtrees, which would lead to poor clustering quality.

Inter-Cluster Similarity

- *Inter_Sim* of C_a and C_b:



but how?

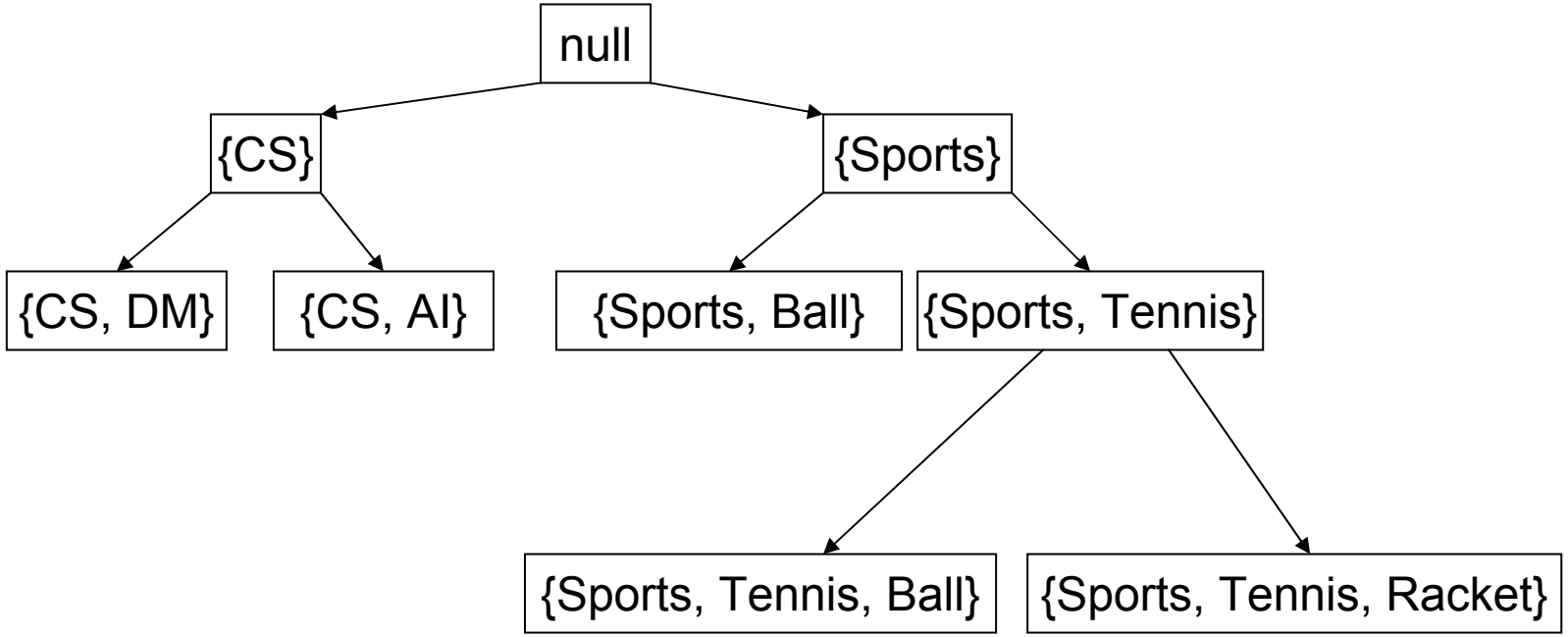
$$Inter\_Sim(C_a \leftrightarrow C_b) = [Sim(C_a \leftarrow C_b) * Sim(C_b \leftarrow C_a)]^{\frac{1}{2}}$$

- Reuse the score function to calculate Sim($C_i \leftarrow C_j$).

$$Sim(C_i \leftarrow C_j) = \frac{Score(\ C_i \leftarrow doc(C_j)\ )}{\sum_x n(x) + \sum_{x'} n(x')} + 1$$

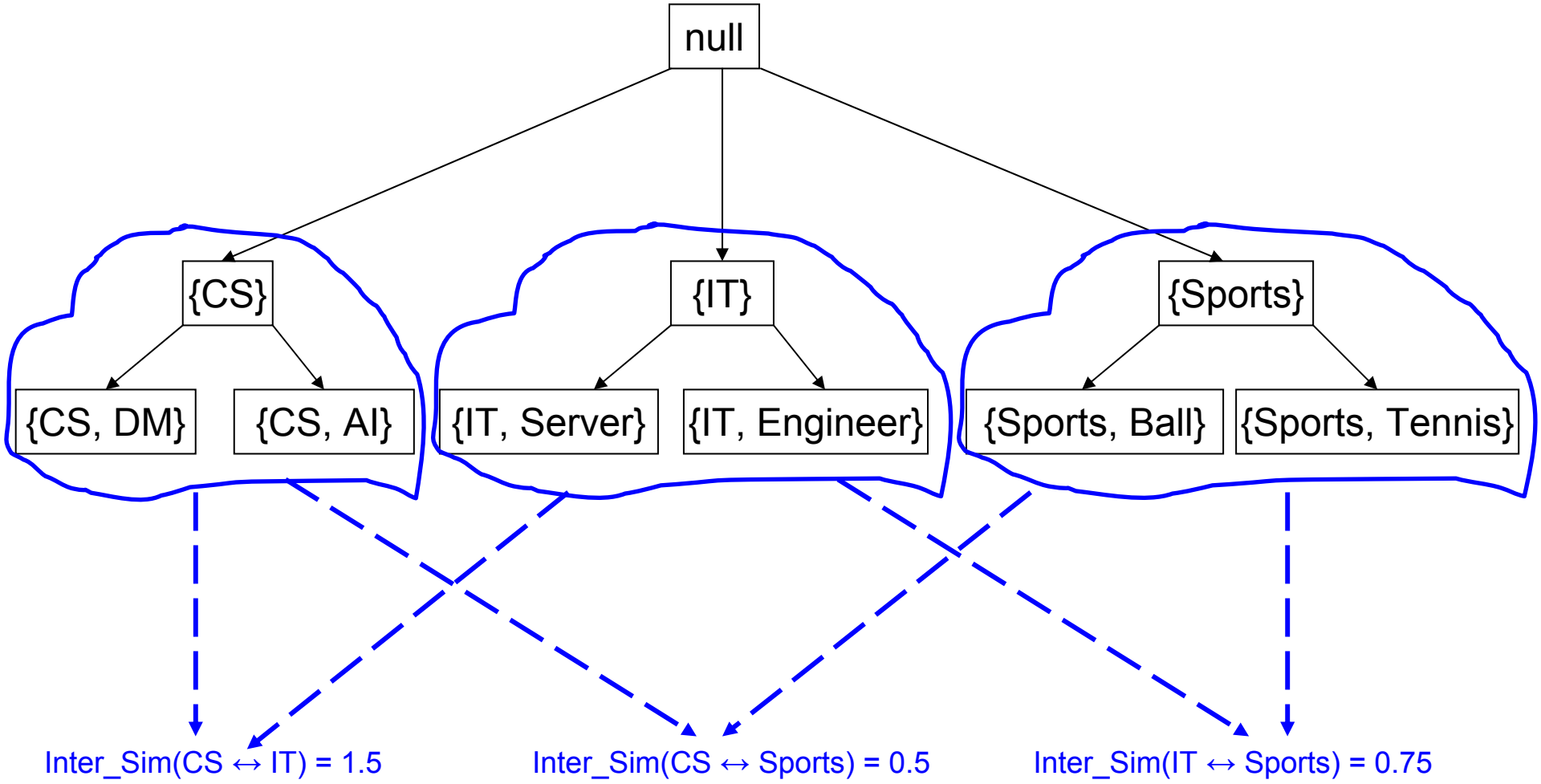
Child Pruning

- Efficiently shorten a tree by replacing child clusters by their parent.
- A child is pruned only if it is similar to its parent.
- Prune if Inter_Sim > 1

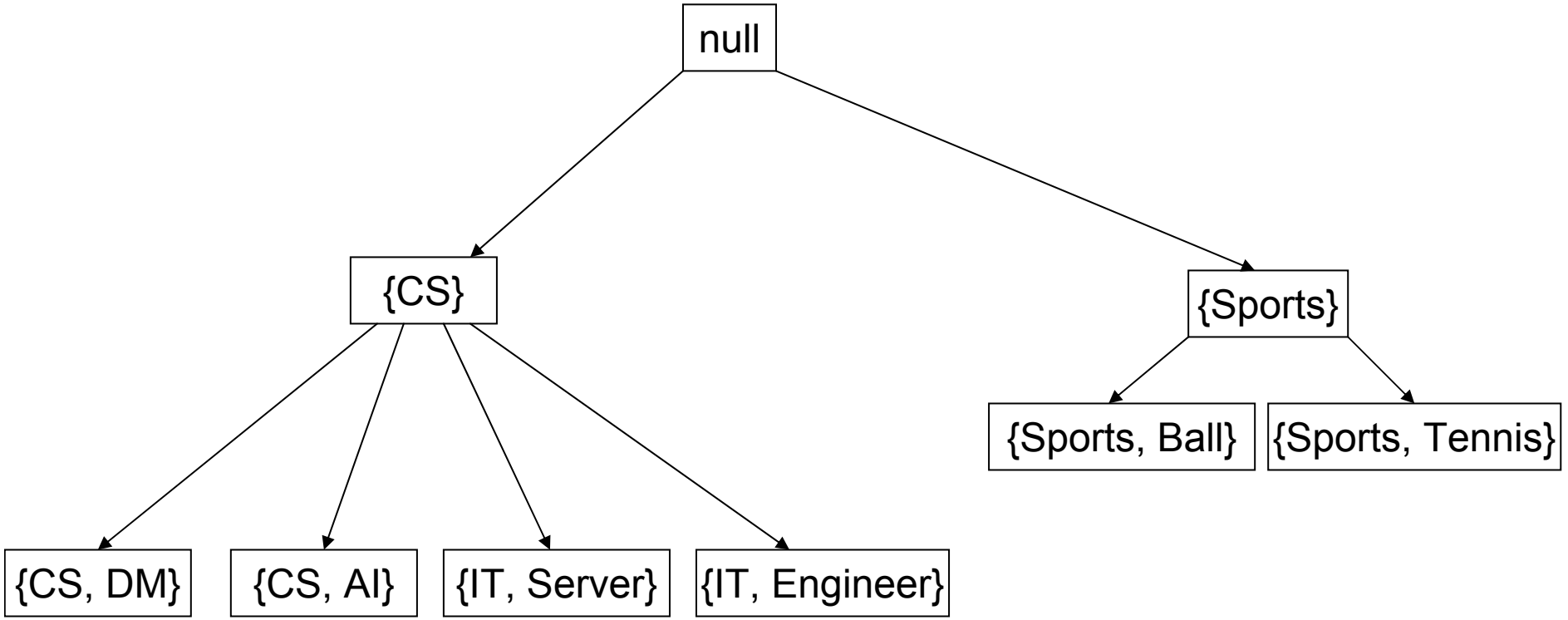


Sibling Merging

- Narrow a tree by merging similar subtrees at level 1.



Sibling Merging



Experimental Results

- Compare with state-of-the-art clustering algorithms:
  - Bisecting k-means (Cluto 2.0 Toolkit)
  - UPGMA (Cluto 2.0 Toolkit)
  - HFTC (Source code in Java from author)
- Clustering quality.
- Efficiency and Scalability.

Data Sets

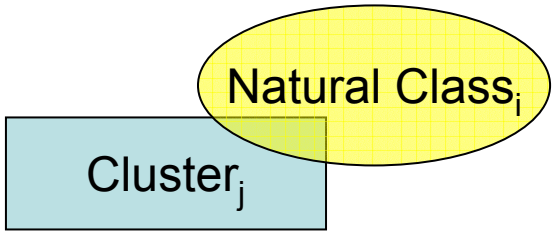
Data Set	Number of Documents	Number of Classes	Class Size	Average Class Size	Number of Terms
Classic4	7094	4	1033 – 3203	1774	12009
Hitech	2301	6	116 – 603	384	13170
Re0	1504	13	11 – 608	116	2886
Reuters	8649	65	1 – 3725	131	16641
Wap	1560	20	5 – 341	78	8460

Table 5.1: Summary descriptions of data sets

- Each document is pre-classified into a single natural class.

Clustering Quality (F-measure)

- Widely used evaluation method for clustering algorithms.



- Recall and Precision.
- F-measure {F(C) }: weighted average of recalls and precisions.
  - from 0 to 1. The higher values imply higher the accuracy

F-measure

$$\mathrm{Re}\,call(K_i, C_j) = \frac{n_{ij}}{|k_i|}$$

$$\mathrm{Pr}\,ecision(K_i, C_j) = \frac{n_{ij}}{|C_j|}$$

$$F(K_i, C_j) = \frac{2 * \mathrm{Re}\,call(K_i, C_j) * \mathrm{Pr}\,ecision(K_i, C_j)}{\mathrm{Re}\,call(K_i, C_j) + \mathrm{Pr}\,ecision(K_i, C_j)}$$

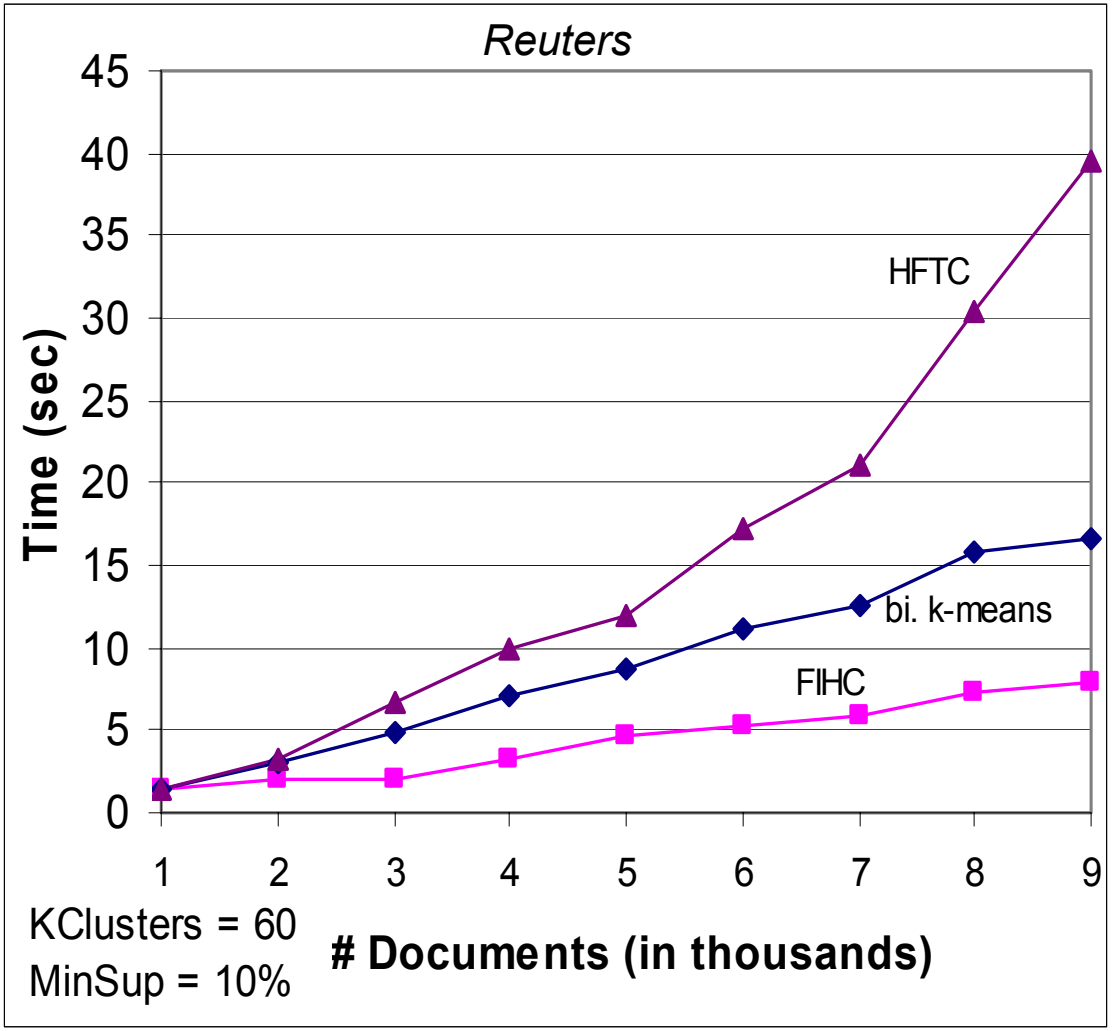
$$F(C) = \sum_{Ki \in K} \frac{|K_i|}{|D|} \max_{C_j \in C} \{F(K_i, C_j)\}$$

- Natural class Ki and Cluster Cj
- Nij is the number of members of natural class Ki in cluster Cj
- Cluster result C

Data Set (# of natural classes)	# of Clusters	F-measure			
		FIHC	**UPGMA**	**Bi. k-means**	**HFTC**
Classic4 (4)	3	0.62*	×	0.59	n/a
	15	0.52*	×	0.46	n/a
	30	0.52*	×	0.43	n/a
	60	0.51*	×	0.27	n/a
	Average	0.54	×	0.44	0.61*
Hitech (6)	3	0.45	0.33	0.54*	n/a
	15	0.42	0.33	0.44*	n/a
	30	0.41	0.47*	0.29	n/a
	60	0.41*	0.40	0.21	n/a
	Average	0.42*	0.38	0.37	0.37
Re0 (13)	3	0.53*	0.36	0.34	n/a
	15	0.45	0.47*	0.38	n/a
	30	0.43*	0.42	0.38	n/a
	60	0.38*	0.34	0.28	n/a
	Average	0.45*	0.40	0.34	0.43
Reuters (65)	3	0.58*	×	0.48	n/a
	15	0.61*	×	0.42	n/a
	30	0.61*	×	0.35	n/a
	60	0.60*	×	0.30	n/a
	Average	0.60*	×	0.39	0.49
Wap (20)	3	0.40*	0.39	0.40*	n/a
	15	0.56	0.49	0.57*	n/a
	30	0.57	0.58*	0.44	n/a
	60	0.55	0.59*	0.37	n/a
	Average	0.52*	0.51	0.45	0.35

Table 5.2: F-measure comparison
× = not scalable to run * = best competitor

Efficiency


Reuters
45
40
35
30
25
20
15
10
5
0
Time (sec)
HFTC
bi. k-means
FIHC
1
2
3
4
5
6
7
8
9
KClusters = 60
MinSup = 10%
# Documents (in thousands)

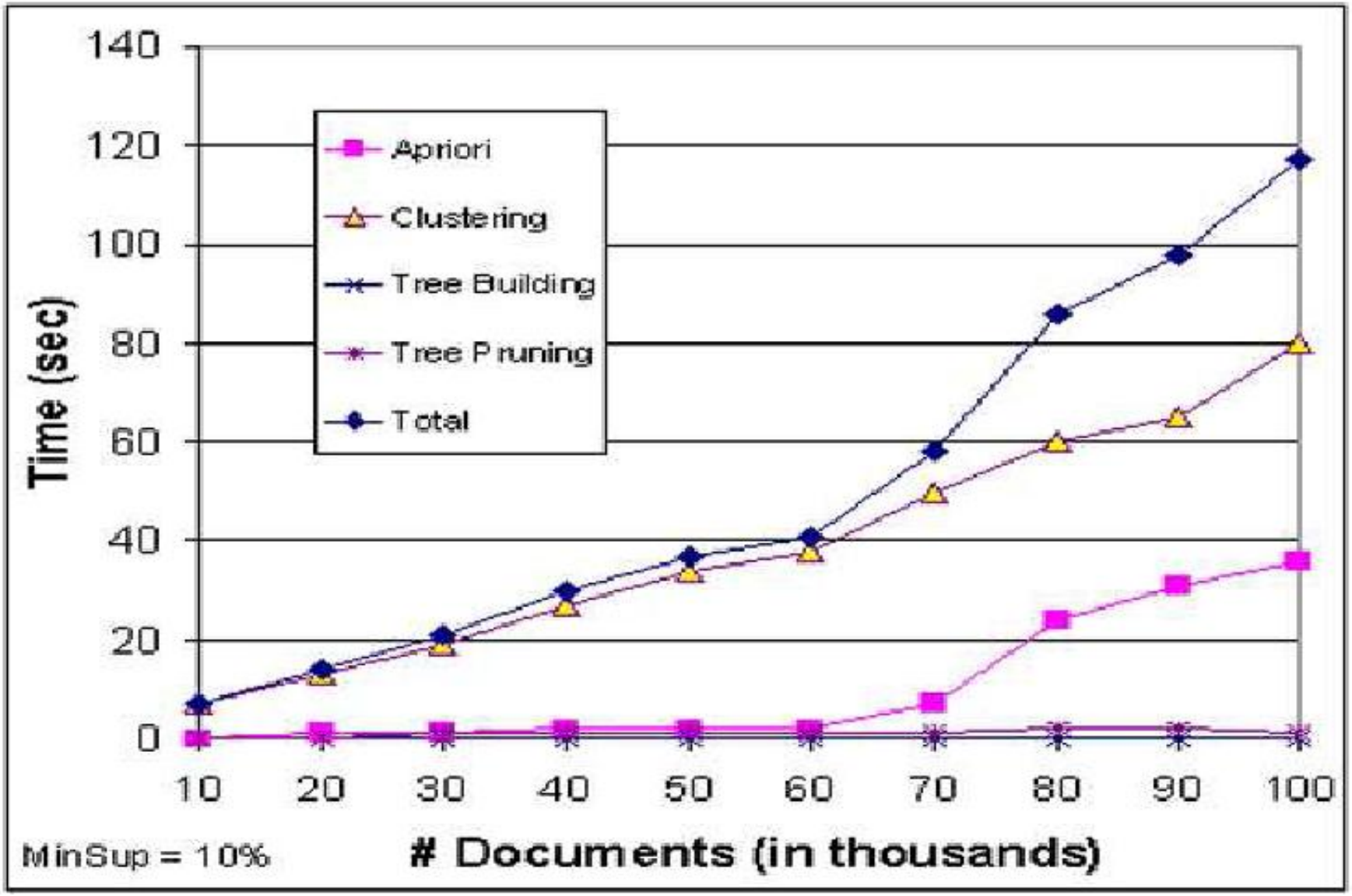


Figure 7: Scalability of our FIHC method
with the scale-up *Reuters* document set

Conclusions

- This research exploits frequent itemsets for:
  - defining a cluster.
  - organizing the cluster hierarchy.

- contributions:
  - Reduced dimensionality → efficient and scalable.
  - High clustering quality.
  - Number of clusters as optional input parameter.
  - Meaningful cluster description.

Q&A

Thanks!