XML Document Classification

Computational Intelligence and Data Mining 2007

April 1-5, Honolulu, HI, USA

Abdelhamid Bouchachia and *Marcus Hassler*

ALPEN-ADRIA
UNIVERSITÄT
KLAGENFURT

Introduction I

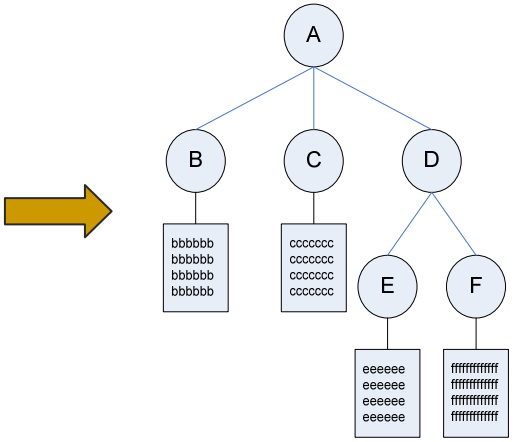
- XML as a standard for data exchange
- Many web applications are adopting XML documents
  - e.g., digital libraries, eLearning, RSS-Feeds
- Need to search and organize such XML repositories

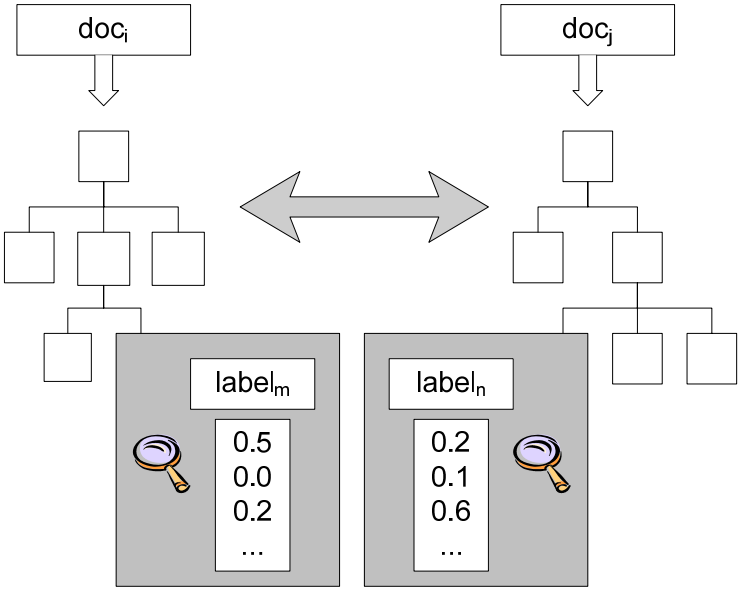
➔ Relevance of **classification** and clustering mechanisms

Introduction II

- Semi-structured XML documents
  - Content + Structure
  - Representation: ordered tree

```
<A>
  <B>bbb…bbb</B>
  <C>ccc…ccc</C>
  <D>
    <E>eee…eee</E>
    <F>fff…fff</F>
  </D>
</A>
```





Introduction III

- Traditional classification task
  - Assign new documents to predefined categories
  - Each document is a single *n*-dimensional data point
- XML-based classification task
  - Hierarchically structured documents
    - Multiple components (*n*-dimensional data points)
    - Different depth levels
- Matching problem
  - Appropriate similarity measures (to apply standard classification)
  - Must take content and structure into account

Outline

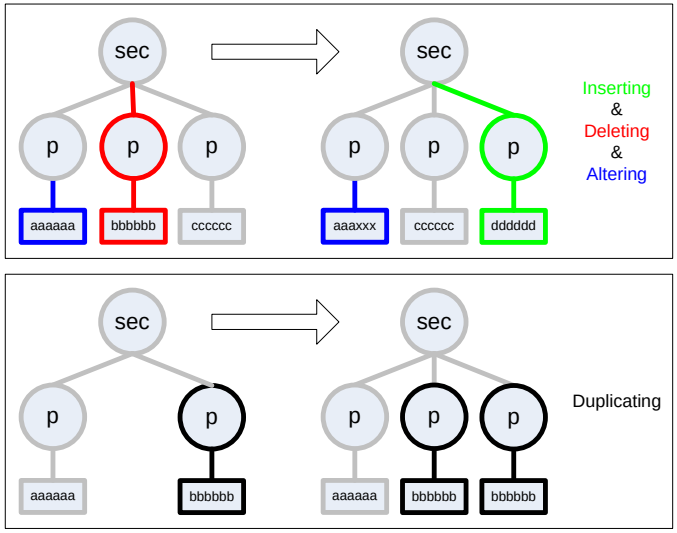
- XML Document Similarity
  - Tree Edit Distance
  - Content Matrix Matching
- Evaluation
  - Settings
  - Results

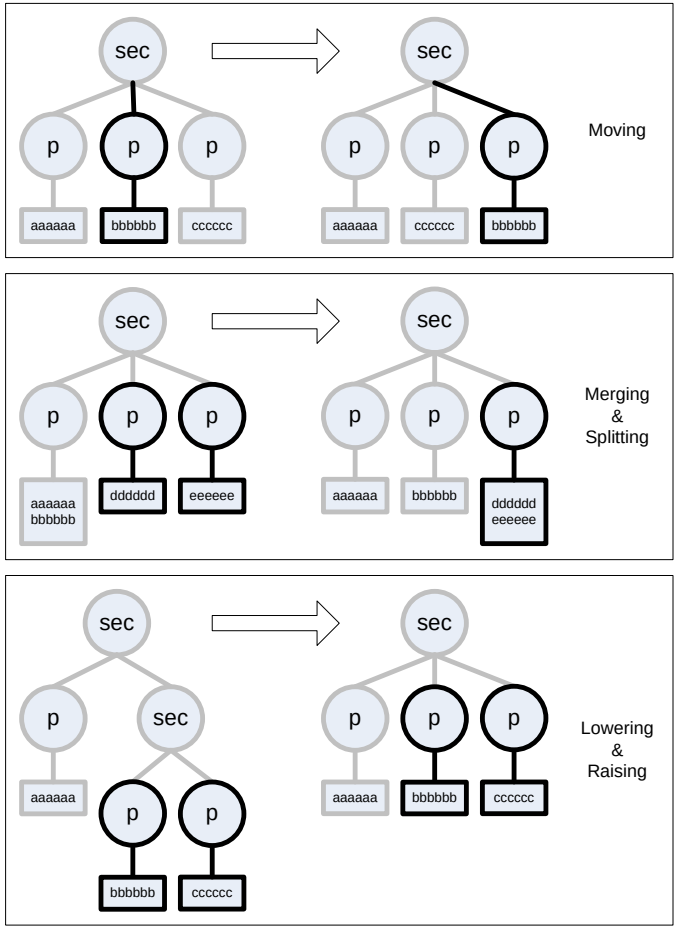
XML Document Similarity I

- How to compare two XML document trees?
  - Given an XML node similarity function, how is the overall similarity between two documents computed?
  - Is the order of XML nodes important?
  - Which “standard” document editing operations have to be considered? How?
    - Moving (raising / lowering) of XML elements
    - Merging / splitting of contents
    - Duplicating XML elements

XML Document Similarity II

"Standard" document editing operations



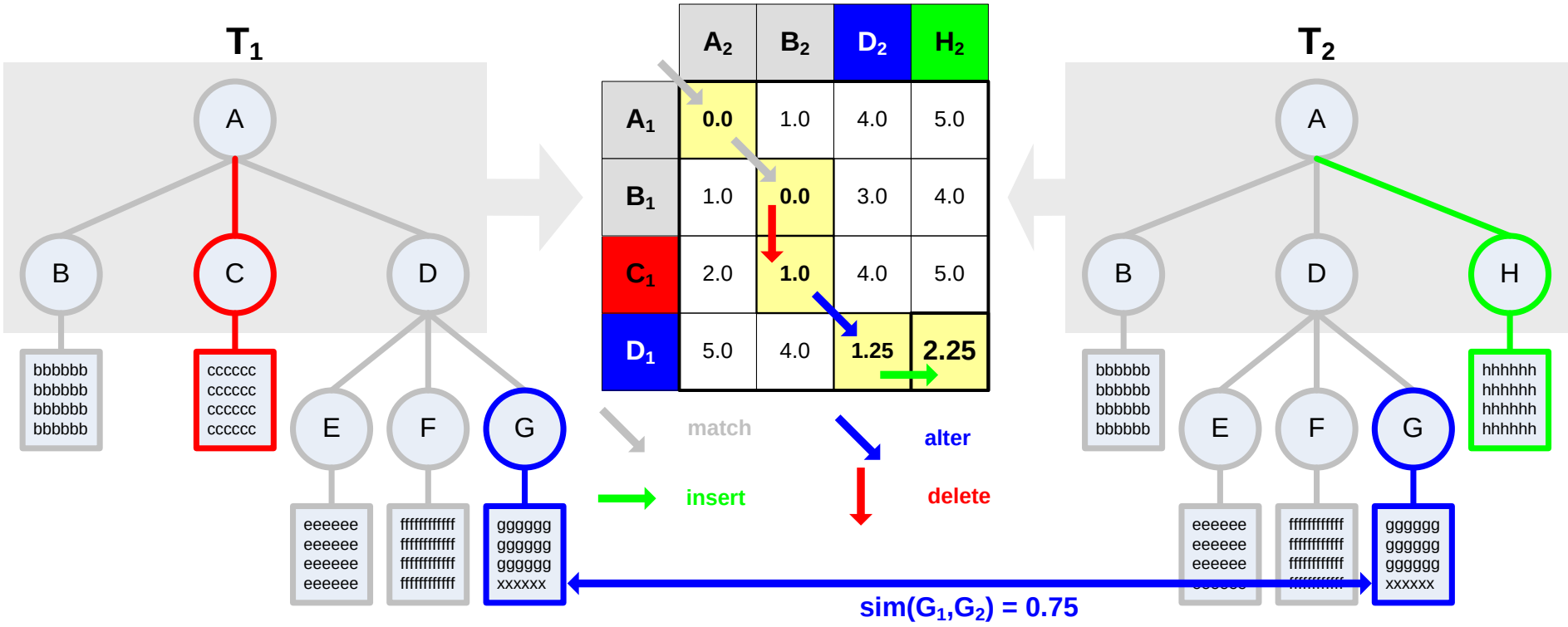


- Two possible solutions
  - Tree Edit Distance
  - Content Matrix Matching

Tree Edit Distance I

- Idea
  - TED(T_1,T_2) ➔ minimum costs of transforming T_1 into T_2
- Cost functions for basic operations
  - Inserting, Deleting, and Altering of nodes
  - Recursive computation complex operations
- Focusing on strutcure

Tree Edit Distance II

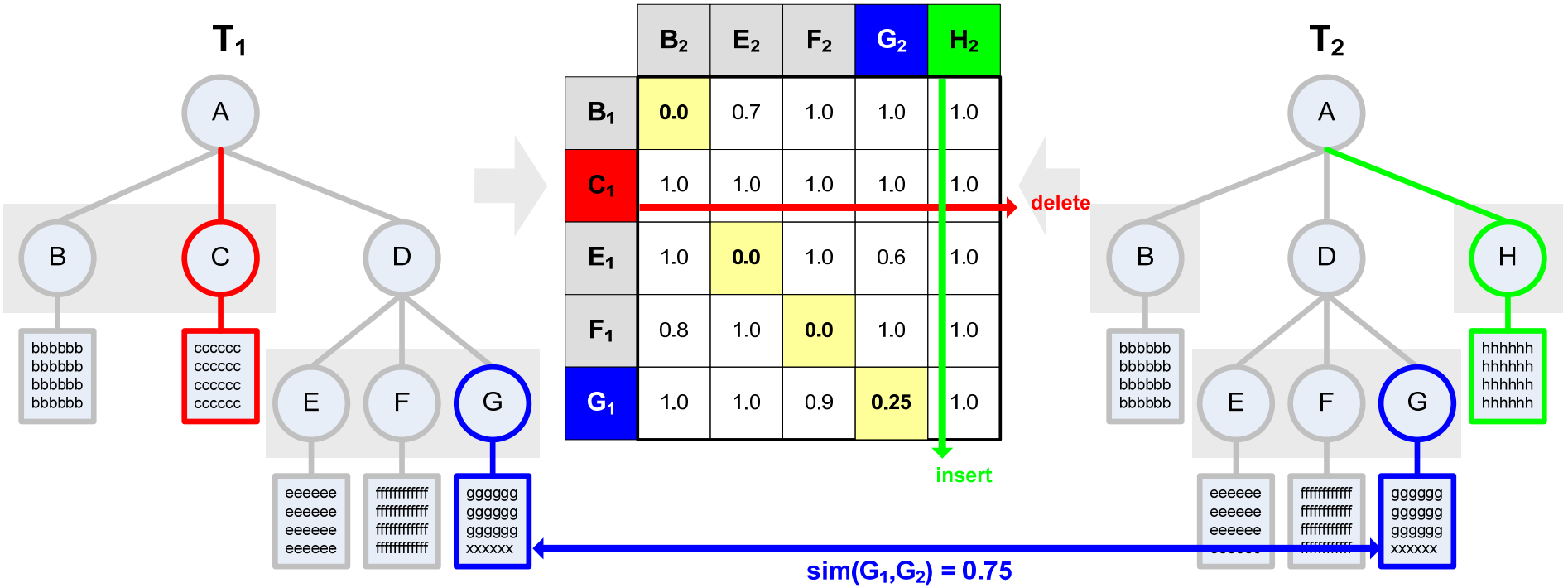


$TED(T_1,T_2) = 1 + 1 + (1-0.75) = 2.25$

Content Matrix Matching I

- Idea
  - CM(T_1,T_2) ➔ minimum costs of transforming content nodes of T_1 into content nodes of T_2
- Cost functions for basic operations
  - Inserting, Deleting, and Altering of nodes
  - No recursive computation
- Focusing on content

Content Matrix Matching II



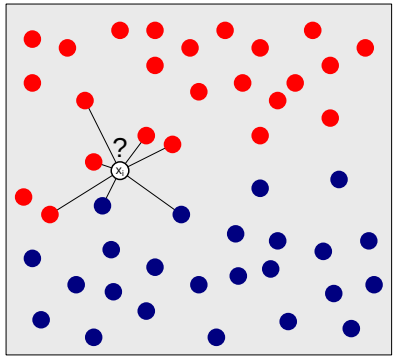
$$CM(T_1,T_2) = 1 + 1 + (1-0.75) = 2.25$$

Outline

- XML Document Similarity
  - Tree Edit Distance
  - Content Matrix Matching
- Evaluation
  - Settings
  - Results

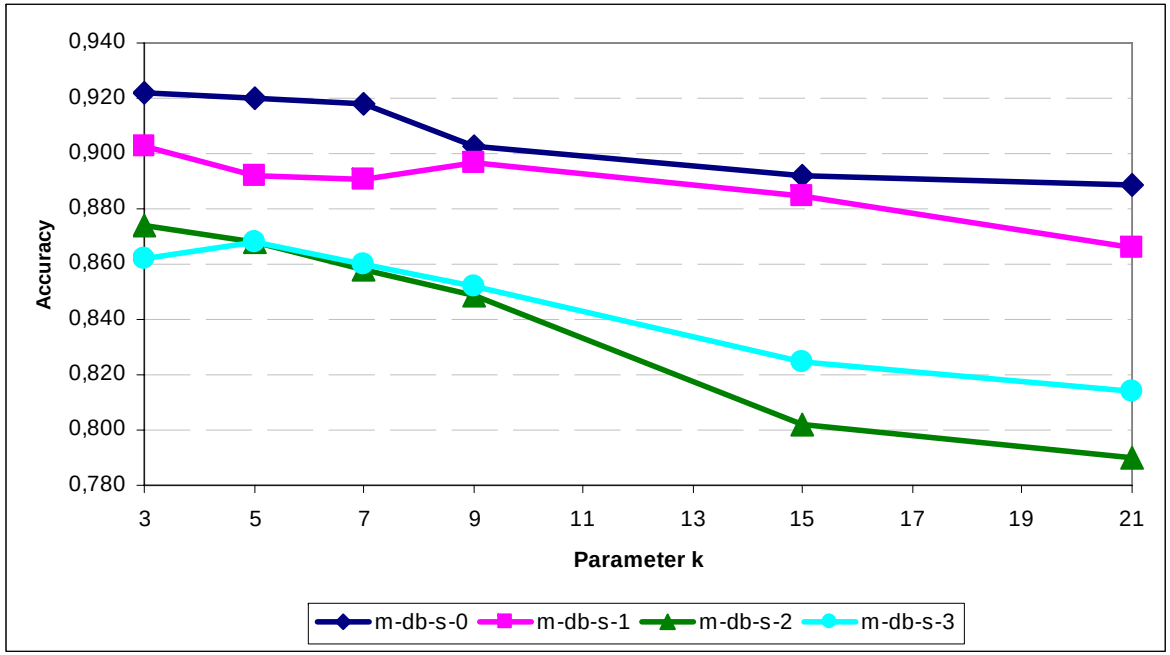
Evaluation – Settings

- Official INEX 2005 dataset (based on MovieDB)
  - Structure only corpora (5000 train, 5000 test, 11 classes)
    - **`m-db-s-0, m-db-s-1, m-db-s-2, m-db-s-3`**
  - Content and structure corpus (2500 train, 2500 test, 11 classes)
    - **`m-ds-cs-1`**
- Evaluation metrics
  - Accuracy, recall, precision
- k Nearest Neighbor (kNN) classification
  - Lazy learning
  - Strongly dependent on a distance measure
  - Simple to implement



Evaluation – # Neighbors (SO)

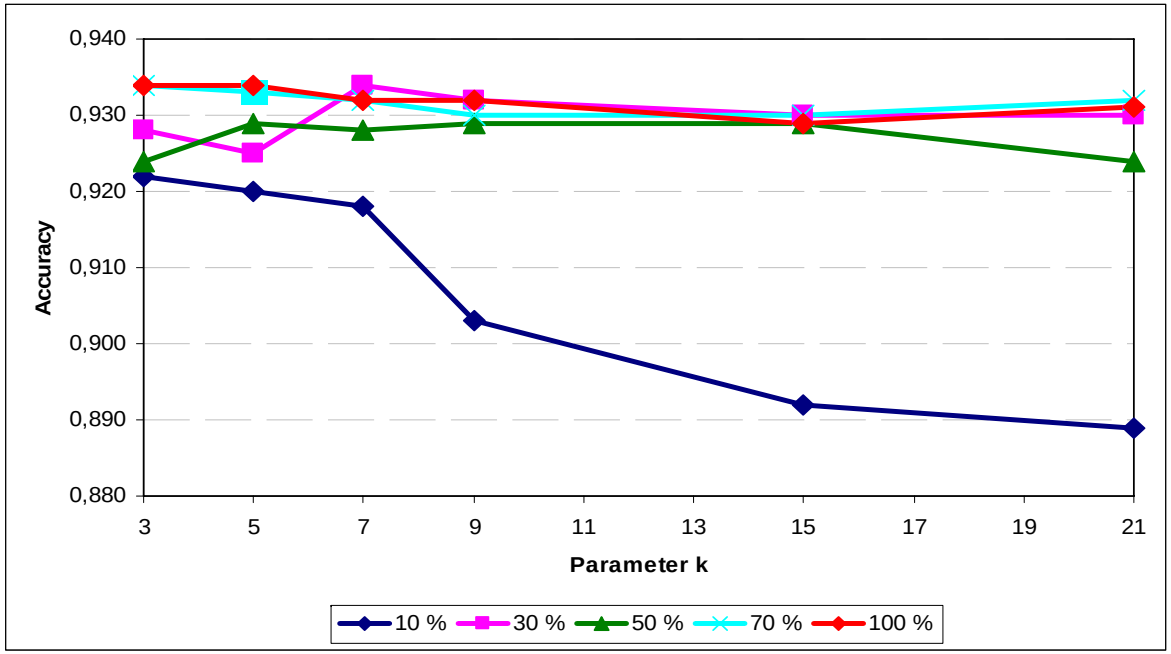
- How do *k* (number of neighbors) affect the accuracy?



- Findings
  - Higher *k* values decrease monotonically the accuracy
  - Increasing noise in **`m-db-s-0/1/2/3`** decreases accuracy
  - Maximum drop <5%

Evaluation – TR reduction (SO)

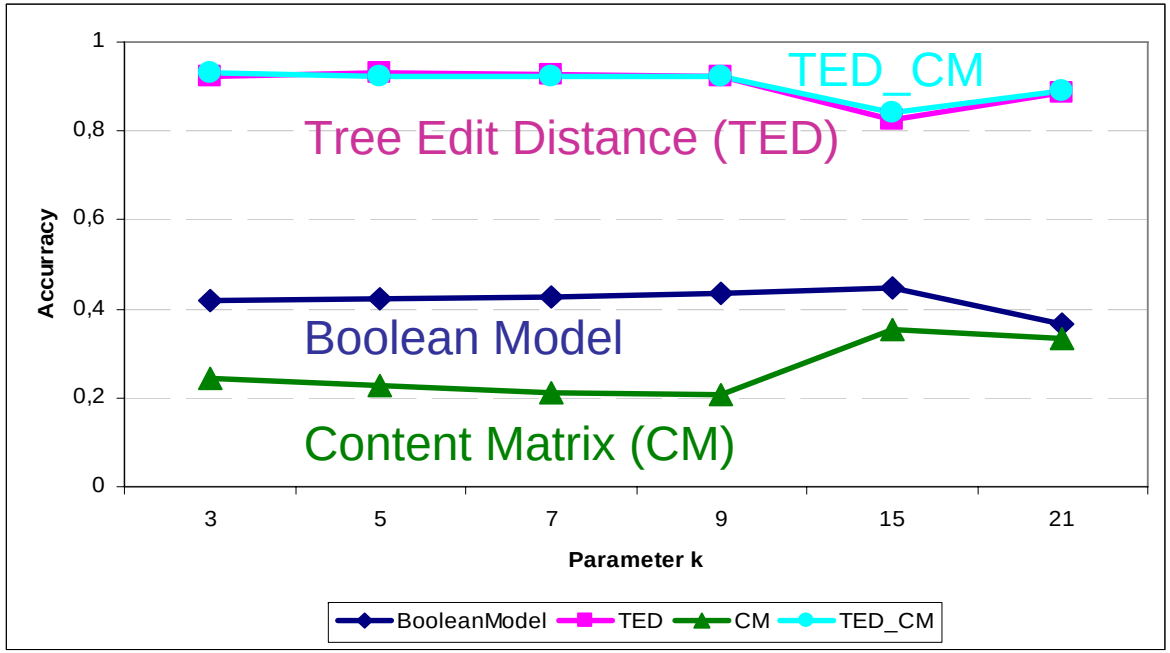
- How does the *size of the training set* affect the accuracy?



- Findings
  - Valid only for this dataset (**`m-db-s-0`**)
  - Training size did not greatly impact the accuracy (except at 10%)
  - ➔ High homogenous document classes

Evaluation – CAS Experiment

- How does *CAS setting* affect the accuracy?



- Findings
  - Compared four methods
  - TED performed equally well on both collections
  - Combined approach showed best results

Evaluation – Comparison

- Results of other INEX participants

Approach	Accuracy	Micro Recall	Macro Recall	Micro Precision	Macro Precision
CSOM-SD	0.873	-[a]	-	-	-
IDT	-	0.968	0.960	-	-
TED_CM	0.934	0.934	0.934	0.937	0.911

[a] '-': means value not available

  - CSOM-SD: Self-Organizing-Maps [Hagenbuchner et al.]
  - IDT: Inductive Decision Trees [Candillier et al.]
  - TED_CM: Tree Edit Distance + Content Matrix

- Findings
  - TED_CM outperforms CSOM-SD
  - IDT better than TED_CM, but unfortunately no precision values

Conclusion

- XML classification impose new constraints (CAS)
- Similarity measures are key-concepts
- Two extensions of edit distance
  - Based on cost functions
  - Tree Edit Distance (structure matching)
  - Content Matrix (content matching)

  ➔ Combined approach
- Experiments showed promising results

Future work

- XML classification requires more investigation w.r.t.
  - Other XML corpora (content-rich)
  - Other classification algorithms (e.g., SVM, NN)
  - Other similarity models
    - Move operations
    - Operations on subtrees
    - etc.