XML Classification

Swathy Giri

Masters Thesis Defense

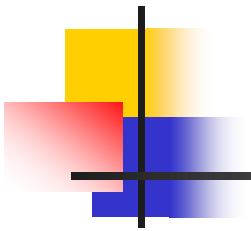
Nov 15th 2004

Committee:

Dr.Susan Gauch

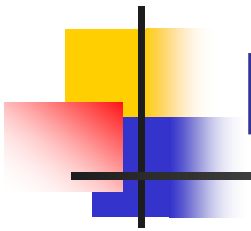
Dr.John Gauch

Dr.Douglas Niehaus



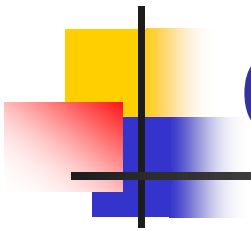
Outline

- Motivation
- Goals
- Related Work
- Data Sets
- System Design
- Evaluation Metric
- Evaluation Experiments
- Predicting Valuable Fields
- Conclusions
- Future Work



Motivation

- HTML is used to represent documents on WWW
  - Data + formatting information
- Organization and Querying
  - Automatic text-classification techniques- Ignore most formatting information
- Migration towards XML for data representation
  - Data + Metadata (Characteristics of data)
  - Metadata referred to as Tags, Field- data (content) within a Tag
- Need for Efficient Organization and Querying techniques



Goals

- Develop a classifier for XML documents
  - Individual fields
  - Weighted combination of fields
- Confirm our hypothesis that fields matter, some fields matter more than others

Goals .....(Cont.)

- Develop an algorithm to predict valuable fields *a priori*
- Validate the algorithm on previously unseen collection

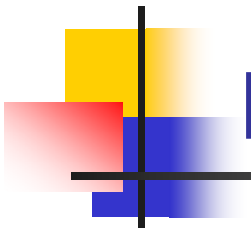
Structure of an XML doc.

```
<?xml version = "1.0"?>
<THESIS>
<AUTHOR> Swathy Giri </AUTHOR>
<TITLE> XML Classification </TITLE>
<DETAILS> Classification of XML
        documents based on content
</DETAILS>
<DATE> November 15 2004 </DATE>
</THESIS>
```

Related Work

- “Classification and Intelligent Search on XML “ – Norbert Fuhr, Gerhard Weikum
  - Considers structure of XML documents
  - Evaluation under progress

- “A belief networks-based generative model for Structured Documents. An application to the XML Categorization **“**- Ludovic Denoyer, Patrick Gallinari
  - Considers both structure and content



Related Work ..(Cont.)

- “XRules: An Effective Structural Classifier for XML Data” -Mohammed J. Zaki, Charu C. Aggarwal
  - Uses *Structural rules*

Data Sets

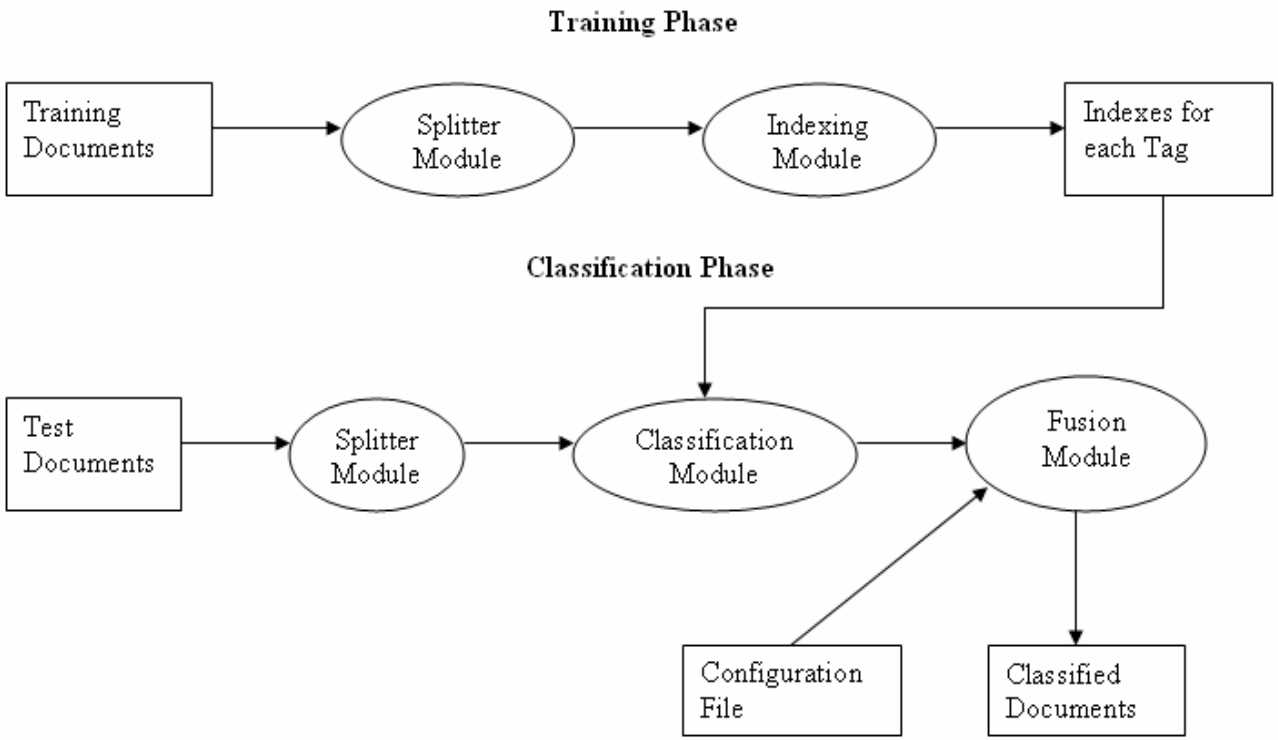
- 2 Data sets, manually created
- 160 documents per data set from 4 categories
- 40 documents/category
- Training – 30 documents per category
- Testing- 10 documents per category

- Data Set 1
  - Collected from www.bbc.com and www.rediff.com
  - Categories- News, Business, Health and Science
  - Used for initial experiments

Data Sets ...(Cont.)

- Data Set 2
  - Information about companies from WWW
  - Categories- Hardware, Technology, Advertising and Cosmetics
  - Used for validation of our algorithms

Overall System Design

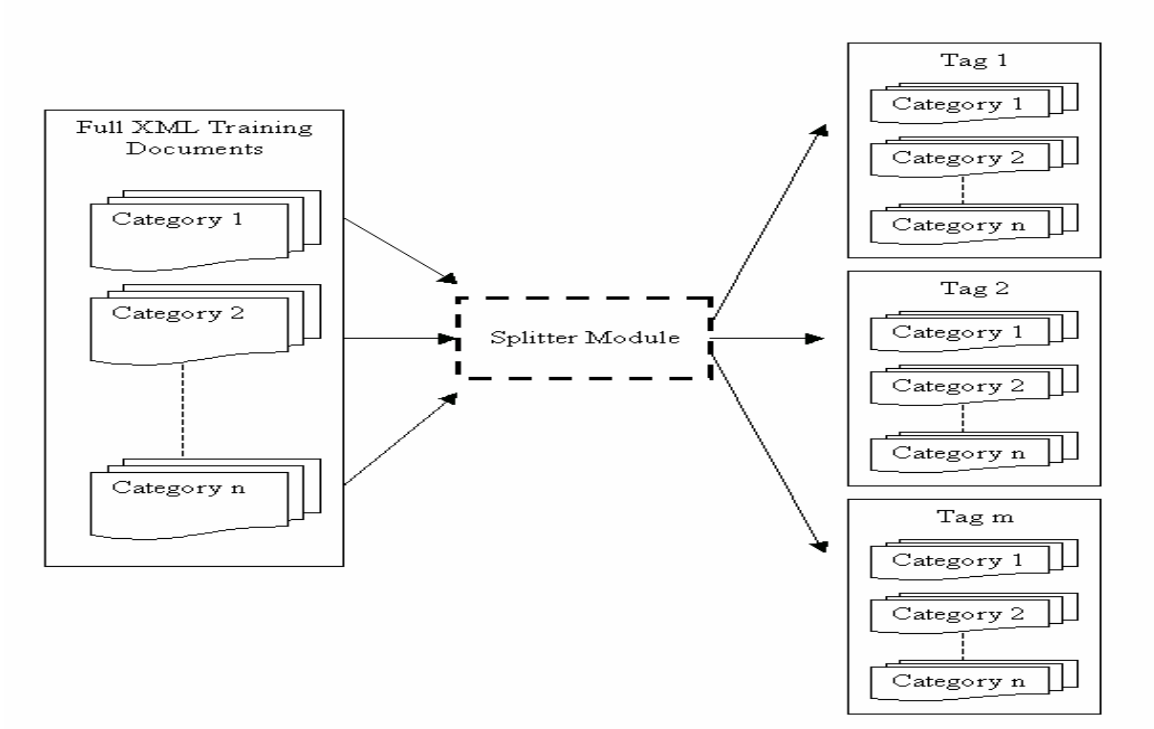


Training Phase -Goals

- Build representative vocabulary for each category
- Train on each field separately
- Train on document as a whole (baseline)

Training Phase- Splitter Module

- Collect sample text from training documents for each category, for each field



Training Phase..(Cont.)- Indexer Module

- Index each category's representative text
- Stores word/weight pairs per category for fast categorization

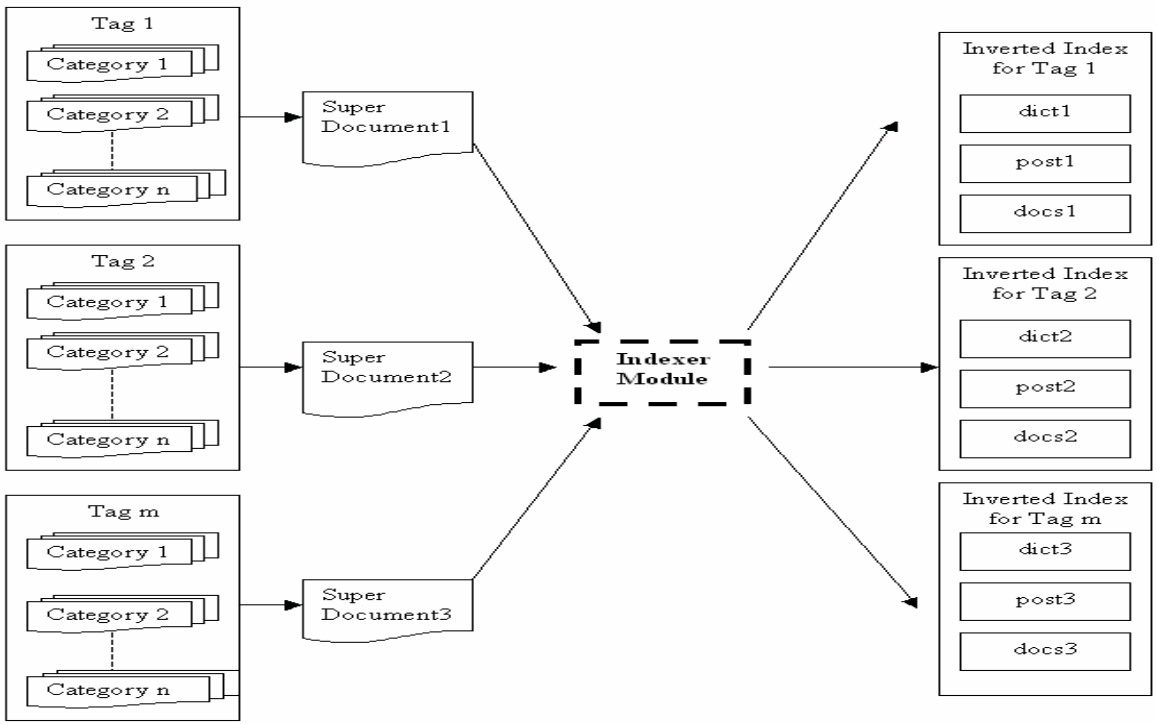


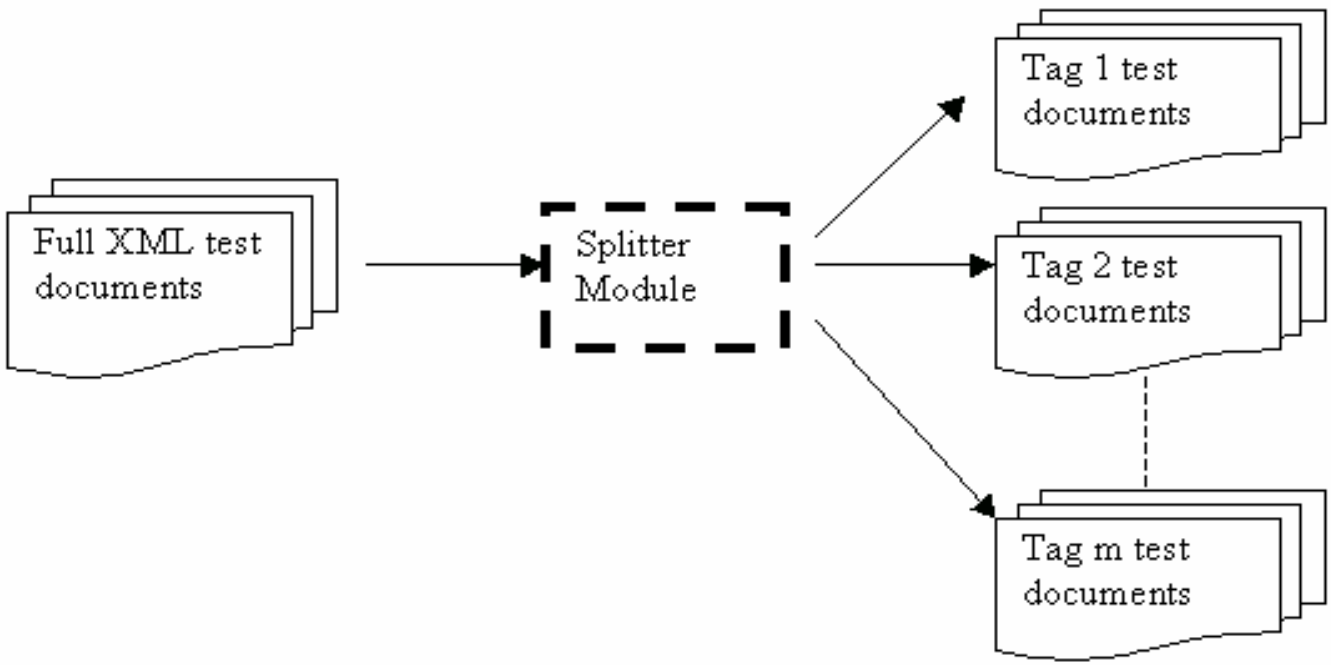
Figure 5: Indexing Process

Classification Phase- Goals

- Match new document to best category based on matches between document vocabulary and category vocabulary (created during indexing in the training phase)

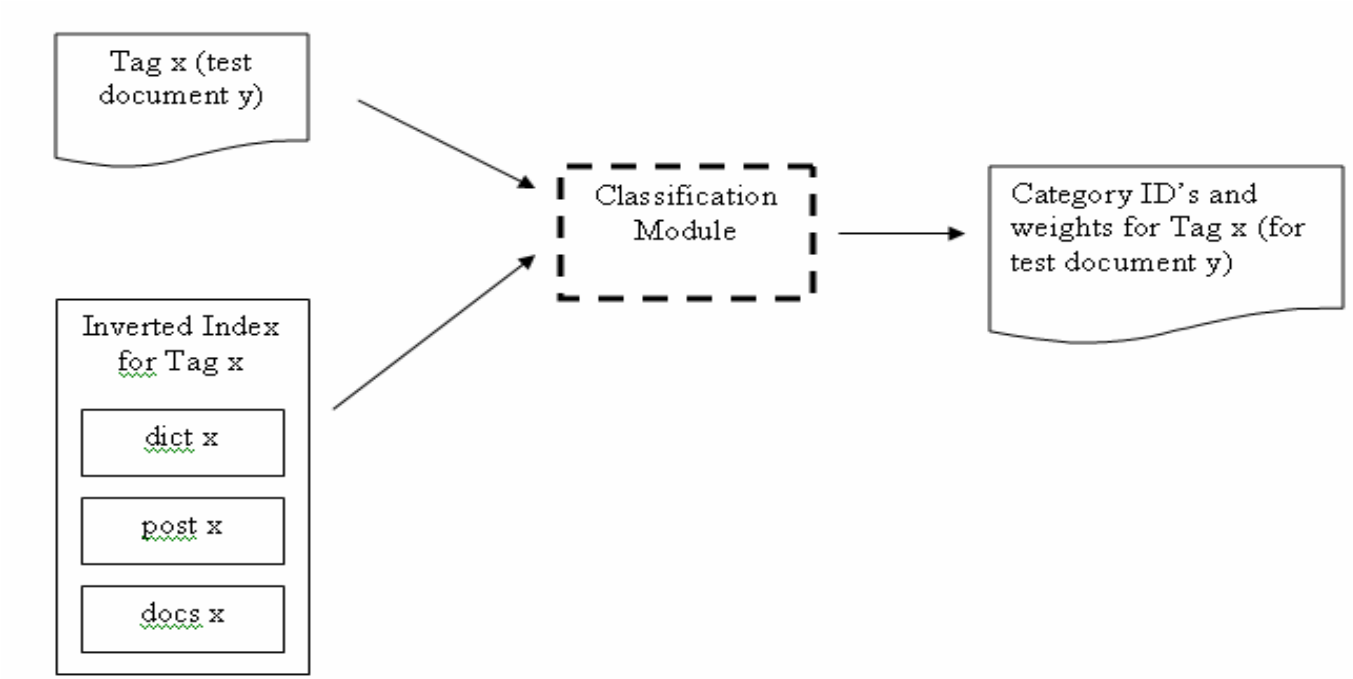
Classification Phase- Splitter Module

- Break each test document into fields for separate categorization



Classification Phase..(Cont.)- Classification Module

- Classifies each test document field separately
- Output: Category ID and weight for top N matches



Classification Result- Sample

- Results of classification for a test document: *cos_31.xml.cat*

Category ID	Weight
2	1.000000
4	0.851478
3	0.633892
1	0.281207

Classification Phase ...(Cont.) - Fusion Module

- Goal- Combine the results of the per field categorizers to a single results list

- Weighted sum of categorizer results for each field per test document

$$\text{weight\_category}_i = \sum_{j=1}^{m} \text{field\_weight}_j * \text{category\_weight}_i$$

Evaluation Metric

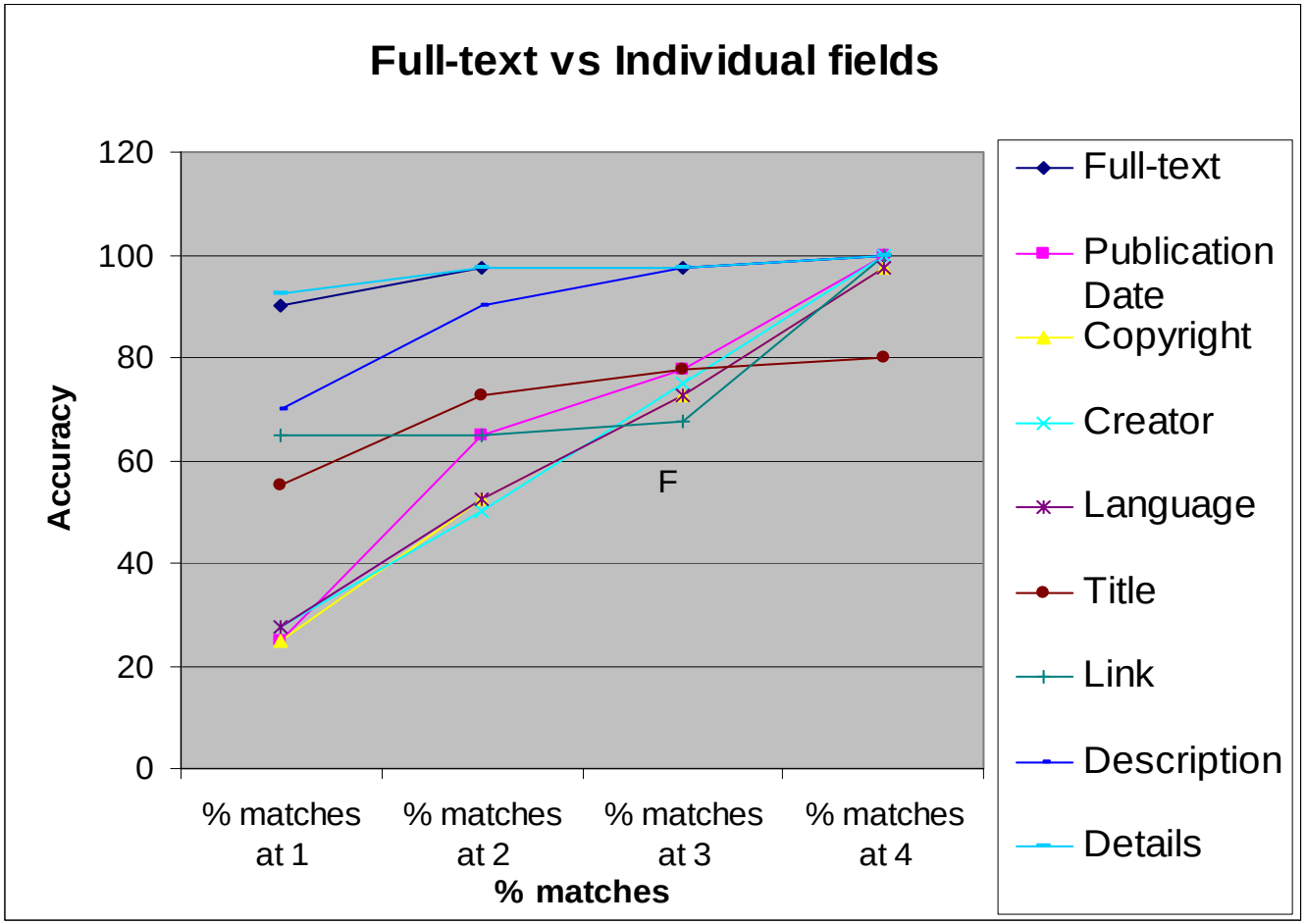
- Classification result for each test document compared with ‘truth’
- Position in which ‘truth’ value appears in the result list is located
- Percentage of test documents for whom truth value occurs as top match and in 2^{nd}, 3^{rd} and 4^{th} is calculated.

Evaluation Experiments(DS1)

- Experiment 1- Classification with single fields
  - Hypothesis
    - Using certain fields will yield better results than using full-text for classification.
  - Procedure
    - Classifiers trained on content of each of the fields
    - Test documents classified on each of the classifiers
  - Baseline
    - Test documents classified using a full-text classifier

Evaluation Experiments…(Cont.)

- Results



Evaluation Experiments...(Cont.)

- Discussion
  - Fields that perform well- Details
    - Characteristics
      - Large number of tokens
      - Variability in content
  - Fields performing badly- Publication Date, Language, and Copyright
    - Characteristics
      - High percentage of numbers
      - Very few tokens
      - Repetition of a small set of words

Evaluation Experiments...(Cont.)

- Experiment 2- Combination of fields for classification
  - Hypothesis
    - Using a combination of fields will perform better than full-text and individual fields for classification
    - weighting fields differently can also improve classification accuracy
  - Procedure
    - All possible weights (summing up to 1.0) generated
    - Every test document classified by each non-zero weighted classifier
    - Results combined using fusion module

Evaluation Experiments...(Cont.)

- Baseline- Full-text classifier
- Results

# non-zero feilds	BaseLine	1 field (Details)	2 fields	3 fields	4 fields	5 fields
% accuracy at top match	90%	92.5%	92.5%	95 %	95 %	87.5 %

Evaluation Experiments...(Cont.)

- Discussion
  - Combination producing best result with 3 fields

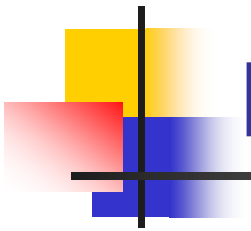
    Details - 0.6

    Link - 0.2

    Title - 0.2
  - Adding 1 more field did not improve accuracy
  - Adding more than 1 field decreased accuracy

Evaluation Experiments...(Cont.)

- Conclusions
  - Characteristics of well-performing fields
    - Large number of tokens
    - High variability in content
    - Less percentage of numbers compared to text



Predicting Valuable Fields

- Goals
  - Design an algorithm that would help to decide fields that would prove useful for classification
  - We have used a back-fit approach by trying to make the algorithm predict the fields helpful for Data Set 1
  - Key-Characteristics-# tokens, Variability and % of numbers

Predicting Valuable Fields - Algorithm

- *Step1:* Calculate **# of tokens**, **Variability**, and **% of numbers** for each field across all the documents in the collection using formulae:

  - **# of tokens in Field T_i** = Total number of words in T_i
  - **Variability for Field T_i** = $\frac{\text{Number of unique words in Field } T_i}{\text{\# of tokens in } T_i}$
  - **% of numbers in Field T_i** = $\frac{\text{Total number of numbers in } T_i}{\text{Total number of characters in } T_i}$

Predicting Valuable Fields -Analysis of DS1

Row	Characteristics	PubDate	Copyright	Creator	Link	Title	Language	Description	Details
1	# of tokens	242	26	241	311	1554	5496	6319	135929
2	Normalized Score # of tokens	0.0016	0.0002	0.0016	0.0021	0.0104	0.0366	0.0421	0.9055
3	Variability	0.128	0.077	0.008	0.013	0.714	0.075	0.378	0.172
4	% of numbers	37.23%	0.00%	0.00%	0.00%	0.64%	18.20%	0.79%	0.52%
5	# tokens Score	0.01	0.00	0.01	0.02	0.08	0.29	0.34	7.24
6	Variability Score	4	3	1	5	8	2	7	6
7	% of numbers Score	1	8	8	8	4	2	3	5
8	Total Score	5.04	11.00	9.04	13.05	12.25	4.88	11.01	32.73

Predicting Valuable Fields – Algorithm..(Cont.)

- *Step2:* Calculate Normalized ***# of tokens score*** by using formula

  - **# of tokens score T_i** $= \dfrac{\#\text{ of tokens for field } T_i \ * \ \#\text{ of Fields}}{\Sigma_{i=1}^{n} \#\text{ of tokens } T_i}$

  (Scores will be in the range 1 to # of fields)

- *Step3:* Calculate **Variability score** and **% of numbers score** by using formulae

Predicting Valuable Fields – Algorithm..(Cont.)

- **Variability score**- rank order the fields by variability and score the most variable field as " 8" and the least variable field as "1"

- **% of numbers score**- rank order the fields by their % of numbers and score the highest percentage as "1" and the lowest as "8".

Predicting Valuable Fields -Analysis of DS1

Row	Characteristics	PubDate	Copyright	Creator	Link	Title	Language	Description	Details
1	# of tokens	242	26	241	311	1554	5496	6319	135929
2	Normalized Score # of tokens	0.0016	0.0002	0.0016	0.0021	0.0104	0.0366	0.0421	0.9055
3	Variability	0.128	0.077	0.008	0.013	0.714	0.075	0.378	0.172
4	% of numbers	37.23%	0.00%	0.00%	0.00%	0.64%	18.20%	0.79%	0.52%
5	# tokens Score	0.01	0.00	0.01	0.02	0.08	0.29	0.34	7.24
6	Variability Score	4	3	1	5	8	2	7	6
7	% of numbers Score	1	8	8	8	4	2	3	5
8	Total Score	5.04	11.00	9.04	13.05	12.25	4.88	11.01	32.73
9	Relative Score	0.05	0.11	0.09	0.13	0.12	0.05	0.11	0.33

Predicting Valuable Fields – Algorithm..(Cont.)

- *Step4:* Calculate **Total score** and **Relative score** using formulae

  - **Total score** for $\mathbf{T_i}$ = 3 * # of tokens score + Variability score + % of numbers score

  - **Relative score** for $\mathbf{T_i}$ = $\dfrac{\text{Total Score for } T_i}{\sum_{i=1}^{n} \text{Total score for } T_i}$

Predicting Valuable Fields – Algorithm..(Cont.)

- *Step5:* Calculate the **threshold** (TH) value using formula
  - TH (for relative score) $= \dfrac{\Sigma_{i=1}^{n} \text{Total score for } T_i}{\text{\# of fields} * 100}$
  - Apply it to the **Relative score** of every field to determine whether or not the field will be included for classification.
- *Step6*: Finally, calculate the **Weights** for each field selected in the previous step, using the formula

$$\textbf{Weight } T_i = \frac{\text{Relative score } T_i}{\text{Sum of Relative scores of fields above TH}}$$

Predicting Valuable Fields -Analysis of DS1

Row	Characteristics	PubDate	Copyright	Creator	Link	Title	Language	Description	Details
1	# of tokens	242	26	241	311	1554	5496	6319	135929
2	Normalized Score # of tokens	0.0016	0.0002	0.0016	0.0021	0.0104	0.0366	0.0421	0.9055
3	Variability	0.128	0.077	0.008	0.013	0.714	0.075	0.378	0.172
4	% of numbers	37.23%	0.00%	0.00%	0.00%	0.64%	18.20%	0.79%	0.52%
5	# tokens Score	0.01	0.00	0.01	0.02	0.08	0.29	0.34	7.24
6	Variability Score	4	3	1	5	8	2	7	6
7	% of numbers Score	1	8	8	8	4	2	3	5
8	Total Score	5.04	11.00	9.04	13.05	12.25	4.88	11.01	32.73
9	Relative Score	0.05	0.11	0.09	0.13	0.12	0.05	0.11	0.33
10	Weights	0	0	0	0.2	0.2	0	0	0.6

Table 7: Analysis of characteristics of fields in DS1

Validating the algorithm on DS2

- DS2 has 10 fields and documents have been selected from 4 different categories, information about companies

  (Name, url, HQ Location, BRLocation, Product, Service, Date Visited, Creator, HQPhone, BR Phone)

- Fields Selected and weights using the algorithm :

Product	Service
0.5	0.5

- Best combination using Brute-force:

Product	Service
0.6	0.4
0.4	0.6

Validating the algorithm on DS2

- Accuracy obtained with combination generated by the algorithm

  82.5%

- Accuracy obtained with Baseline(full-text)

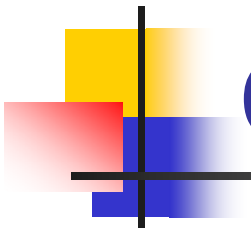
  65 %

- Accuracy obtained with combinations generated by brute-force algorithm

  85 %

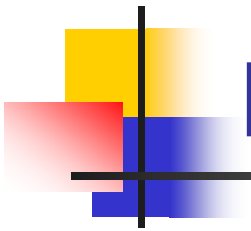
Validating the algorithm on DS2

- Thus, performance of our system is
  - 25 % better (17.5 % absolute improvement) than our baseline system
  - within 0.03 % (2.5 % absolute degradation) of the best combination found by the brute-force method



Conclusions

- Selected fields can be used to improve classification
- Characteristics of useful fields have been identified
- Algorithm to identify useful fields presented



Future Work

- Extension to multiple Schemas
  - Normalizing participating schemas
- Automating field selection
  - Key-pieces available
- Further Validation
  - Larger data sets
  - Larger schema
  - Real-world data sets