Web Information Extraction

Automatic Data Extraction from Lists and Tables in Web Sources
(Lerman, Knoblock, Minton)

Presentation by Marian Schedenig (9725416)
(Paper #7)
Motivation and Goals

- Create a wrapper to extract dynamic data from web pages
Motivation and Goals

- Create a wrapper to extract dynamic data from web pages
- Unsupervised algorithm
Motivation and Goals

- Create a wrapper to extract dynamic data from web pages
- Unsupervised algorithm
- Do not rely entirely on pure HTML structure
Algorithm Outline

- Find the page template
Algorithm Outline

- Find the page template
- Extract data
Algorithm Outline

- Find the page template
- Extract data
- Classify fields
Algorithm Outline

- Find the page template
- Extract data
- Classify fields
- Identify records
Finding the page template

Algorithm

- Take a set of example pages
Finding the page template

Algorithm

- Take a set of example pages
- Split them into tokens
Finding the page template

Algorithm

- Take a set of example pages
- Split them into tokens

Tokens: HTML, punctuation, numeric, capitalised alpha, lowercase alpha
Finding the page template

Algorithm

• Take a set of example pages
• Split them into tokens
  Tokens: HTML, punctuation, numeric, capitalised alpha, lowercase alpha
• Grow token sequence
Finding the page template
Algorithm

- Take a set of example pages
- Split them into tokens
  Tokens: HTML, punctuation, numeric, capitalised alpha, lowercase alpha
- Grow token sequence
- Append to template if it contains k tokens and appears exactly once on each page
Extracting data

Algorithm

- List = all tokens not in page sequence
Extracting data

Algorithm

- List = all tokens not in page sequence
- Extract list data from page
Extracting data

Algorithm

- List = all tokens not in page sequence
- Extract list data from page
- Split into extracts using separators
Extracting data
Algorithm

- List = all tokens not in page sequence
- Extract list data from page
- Split into extracts using separators
  sequential HTML tags
Extracting data

Algorithm

- List = all tokens not in page sequence
- Extract list data from page
- Split into extracts using separators
  sequential HTML tags
  punctuation characters: all but “.(−)'%” (empirically chosen)
Classifying fields
Basic considerations

- Content alone not sufficient
Classifying fields
Basic considerations

• Content alone not sufficient
e.g. similarity of restaurant and city names
Classifying fields
Basic considerations

- Content alone not sufficient
  e.g. similarity of restaurant and city names
- Separators alone not sufficient
Classifying fields

Basic considerations

- Content alone not sufficient
e.g. similarity of restaurant and city names

- Separators alone not sufficient

  ➤ Use both for classification
Classifying fields

Algorithm

- Enumerate unique identifiers
Classifying fields
Algorithm

- Enumerate unique identifiers
- Assign a set of features to each extract:
Classifying fields

Algorithm

- Enumerate unique identifiers
- Assign a set of features to each extract:
  - Preceding separator → integer value
Classifying fields
Algorithm

- Enumerate unique identifiers
- Assign a set of features to each extract:
  - Preceding separator ➤ integer value
  - Succeeding separator ➤ integer value
Classifying fields
Algorithm

- Enumerate unique identifiers
- Assign a set of features to each extract:
  - Preceding separator ➤ integer value
  - Succeeding separator ➤ integer value
  - Data type pattern ➤ Set of flags
Classifying fields
Determining patterns

- Group extracts into clusters by separators
Classifying fields
Determining patterns

- Group extracts into clusters by separators
- DataPro learns patterns for each cluster
Classifying fields
Determining patterns

- Group extracts into clusters by separators
- DataPro learns patterns for each cluster
e.g. [Number] ... “Street”
Classifying fields
Determining patterns

- Group extracts into clusters by separators
- DataPro learns patterns for each cluster
e.g. [Number] ... “Street”
- Determine flags for each extract:
Classifying fields
Determining patterns

- Group extracts into clusters by separators
- DataPro learns patterns for each cluster
e.g. [Number] ... “Street”
- Determine flags for each extract:
  \[ f_1, f_2, \ldots, f_n \]
Classifying fields
Determining patterns

• Group extracts into clusters by separators
• DataPro learns patterns for each cluster
e.g. [Number] ... “Street”
• Determine flags for each extract:

\[ f_1 f_2 \ldots f_n \]

\[ f_x = 1 \text{ if any of the patterns for cluster } x \text{ matches the extract } \]

\[ f_x = 0 \text{ otherwise } \]
Classifying fields
Algorithm

- Enumerate unique identifiers
- Assign a set of features to each extract:
  - Preceding separator ➤ integer value
  - Succeeding separator ➤ integer value
  - Data type pattern ➤ Set of flags
Classifying fields

Algorithm

- Enumerate unique identifiers
- Assign a set of features to each extract:
  - Preceding separator ➤ integer value
  - Succeeding separator ➤ integer value
  - Data type pattern ➤ Set of flags
- Use AutoClass to cluster extracts
Identifying records
Basic considerations

• Label every element in the list
Identifying records
Basic considerations

- Label every element in the list
  - break list into rows (=records)
Identifying records

Basic considerations

- Label every element in the list
  - break list into rows (=records)
- Problem: Missing columns
Identifying records
Basic considerations

- Label every element in the list
  ▶ break list into rows (=records)
- Problem: Missing columns
- Problem: AutoClass errors
Identifying records
Algorithm

- Think of the sequence of AutoClass labels as a string generated by a regular language
Identifying records
Algorithm

- Think of the sequence of AutoClass labels as a string generated by a regular language
- Learn this language to find the pattern describing a single record
Think of the sequence of AutoClass labels as a string generated by a regular language.

Learn this language to find the pattern describing a single record.

ALERGIA-derived algorithm.
Identifying records
Algorithm

- Think of the sequence of AutoClass labels as a string generated by a regular language.
- Learn this language to find the pattern describing a single record.
- ALERGIA-derived algorithm learns from positive examples, linear performance.
Summary

- Builds a wrapper
Summary

- Builds a wrapper
- Works for data structured using HTML tags and punctuation characters
Summary

- Builds a wrapper
- Works for data structured using HTML tags and punctuation characters
- Finds and skips page template