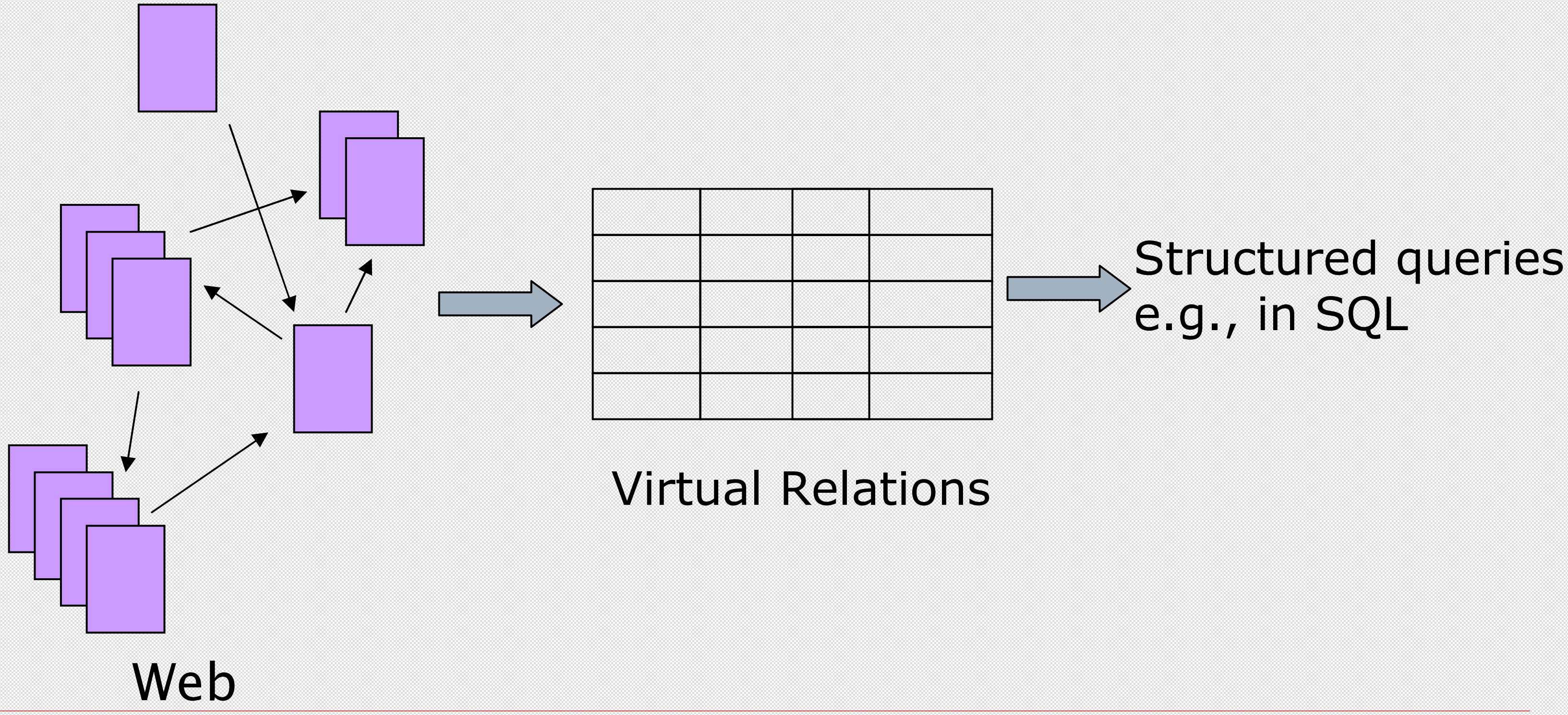
CS345
Data Mining

Virtual Databases

Example

- □ Find marketing manager openings in Internet companies so that my commute is shorter than 10 miles.



Web

Virtual Relations

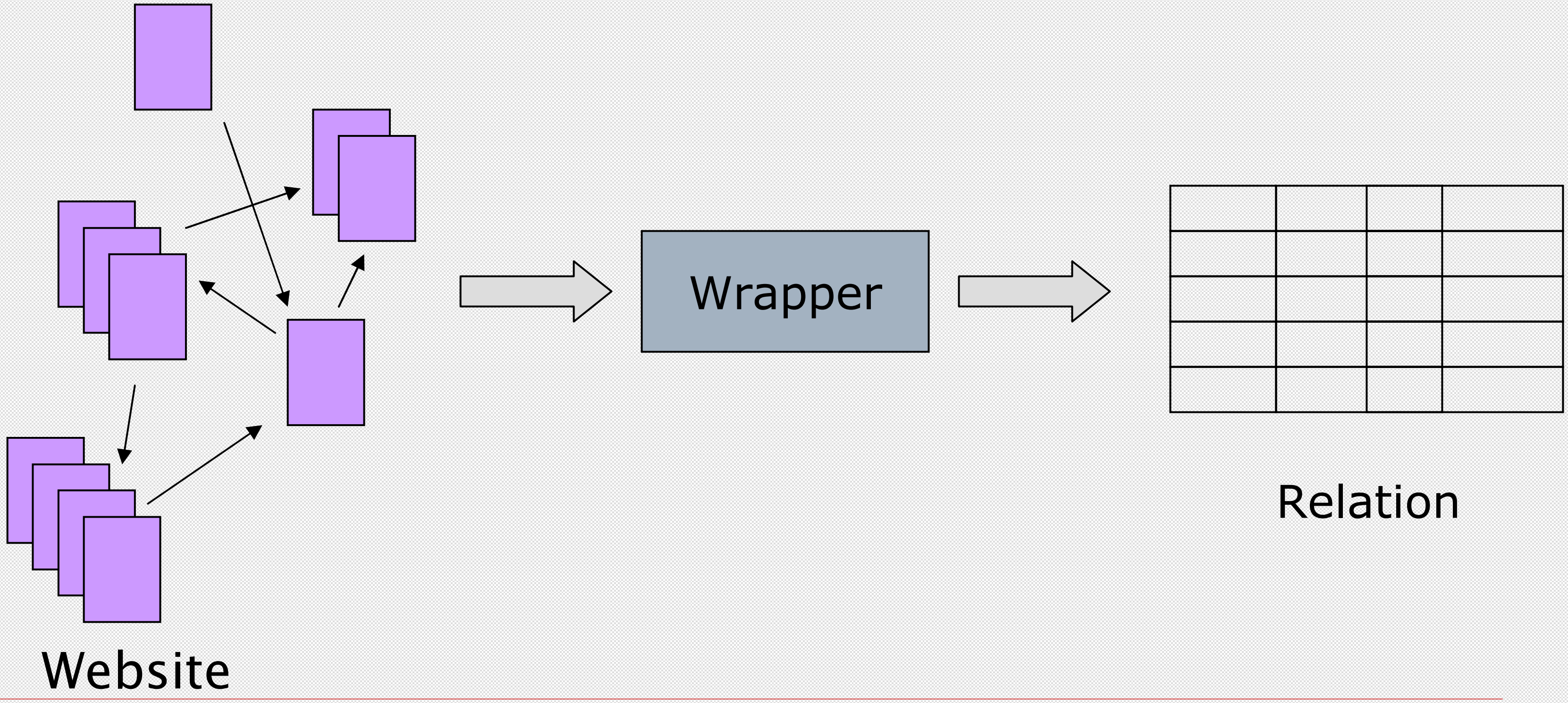
Structured queries e.g., in SQL

Applications

- Comparison shopping
  - shopping.com, fatlens, mobissimo,…
- Job search
  - indeed.com, simplyhired,…
- Classifieds Search
  - oodle
- Integrating web data with relational enterprise apps
  - purchasing, pricing,…

Wrappers

- Extract tuples from a single website
- Assume website is a static collection of pages i.e., no forms



Not same as Relation Extraction

- Why can’t we use DIPRE or Snowball?
  - Can’t assume that the same tuple can be found on many different websites
  - Need to extract **all** the tuples from each website
  - May need to normalize data values across websites
  - Data may be behind forms
    - Need to account for **query capabilities** of websites

Brute force approach

- Write a custom program tailored to the website
  - e.g., in perl, python,…
- Does not scale to thousands of websites
  - Each site needs a different wrapper
- Website changes break wrappers

Simpler problem

- Simplified version of wrapper problem
  - Given a set of pages from the same website, that share the same structure
    - E.g., product detail pages from Amazon.com
  - We have a target relation schema
    - E.g., (product,description,price)
  - Human labels a small subset of pages
    - Marks tuple components on pages
  - Can we deduce the structure?

Two web pages

```
<body><h1>Apple 20GB iPod</h1>
<img href=“xyz”>
Our Price: $204.99
<p> Cool product.
</body>
```

```
<body><h1>Apple 4GB iPod nano</h1>
<img href=“abc”>
Our Price: $250.99
<p> Even cooler product.
</body>
```

Labeled pages

```
<body><h1>Apple 20GB iPod</h1>
<img href="xyz">
Our Price: $204.99
<p> Cool product.
</body>
```

```
<body><h1>Apple 4GB iPod nano</h1>
<img href="abc">
Our Price: $250.99
<p> Even cooler product.
</body>
```

LR (Left-Right) Wrapper

```
<body><h1>Apple 20GB iPod</h1>
<img href=“xyz”>
Our Price: $204.99
<p> Cool product.
</body>
```

- Fix an order for attributes (product, price, description)
- Use patterns of the form $*L_i(\text{attribute}_i)R_i*$

L_1 = “<body><h1>”	R_1 = “</h1><img href=”
L_2 = “Our Price: ”	R_2 = “<p>”
L_3 = “<p>”	R_3 = “</body>”

Example: (Product, Price)

```
<body>
<b>Holiday Sale</b><em>save $$</em>
<p>
<b>Shoes:</b><em>$100</em> <br>
<b>Ship:</b><em>$1000</em>
</body>
```

```
<body>
<b>Everyday low prices</b><em>guaranteed</em>
<p>
<b>Sealing wax:</b><em>$1</em>
</body>
```

L_1=“<b>” R_1=“</b>”

L_2=“<em>” R_2=“</em>”

HLRT (Head-Left-Right-Tail) Wrappers

```
<body>
<b>Holiday Sale</b><em>save $$</em>
<p>
<b>Shoes:</b><em>$100</em> <br>
<b>Ship:</b><em>$1000</em>
</body>
```

```
<body>
<b>Everyday low prices</b><em>guaranteed</em>
<p>
<b>Sealing wax:</b><em>$1</em>
</body>
```

L_1="<b>" R_1="</b>"

L_2="<em>" R_2="</em>"

H="</em><p>" T = "<body>"

Example: (Product, Price)

```
<body>
<b>Holiday Sale</b><em>save $$</em>
<p>
<b>Shoes:</b><em>$100</em> <br>
<b>Ship:</b><em>$1000</em>
</body>
```

```
<body>
<b>Cabbages</b><em>$10</em>
<p>
<b>Sealing wax:</b><em>$1</em>
</body>
```

Cannot construct a HLRT wrapper

Book-author-year example

```
Books by <b>Isaac Asimov</b><ul>
<li>Foundation (1951)</li>
<li>Nightfall (1941)</li>
</ul><p>
Books  by <b>Arthur C Clarke</b><ul>
<li>Rendezvous with Rama (1976)</li>
</ul>
```

Limitations of HLRT

- Contiguous tuples
  - All tuple components must be on the same page
  - One tuple must end before next one begins
- Needs human labeling
  - Because labeling needs to be accurate
  - Can we use "noisy" automatic taggers that can make some mistakes?