Data Mining on Symbolic Knowledge Extracted from the Web

Rayid Ghani*, Rosie Jones*, Dunja Mladenić†*, Kamal Nigam*, Seán Slattery*

*School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
<firstname>.<lastname>@cs.cmu.edu

†Department for Intelligent Systems
J. Stefan Institute
Ljubljana, Slovenia
<firstname>.<lastname>@ijs.si

ABSTRACT

Information extractors and classifiers operating on unrestricted, unstructured texts are an errorful source of large amounts of potentially useful information, especially when combined with a crawler which automatically augments the knowledge base from the world-wide web. At the same time, there is much structured information on the World Wide Web. Wrapping the web-sites which provide this kind of information provide us with a second source of information; possibly less up-to-date, but reliable as facts. We give a case study of combining information from these two kinds of sources in the context of learning facts about companies. We provide results of association rules, propositional and relational learning, which demonstrate that data-mining can help us improve our extractors, and that using information from two kinds of sources improves the reliability of data-mined rules.

1. INTRODUCTION

The World Wide Web has become a significant source of information. Most of this computer-retrievable information is intended for consumption by humans and is not readily-available as a data source in computer-understandable form. One current research challenge for this domain is to have computers not only gather and represent knowledge existing on the Web, but also to use that knowledge for planning, acting, and creating new knowledge. In other words, is it possible to learn new things from the Web?

If this challenge is thought of as a stepwise process of first gathering knowledge and then mining it, then several researchers have addressed the first piece of this challenge. The wrapper induction community [12, 11] has developed learning algorithms for extracting propositional knowledge from highly-structured automatically-generated web pages. Their goal is to reconstruct the explicit data sources used to create the web pages. For example, [3] efficiently learn extractors for information about movie theaters and restaurants from Web-based entertainment guides, and combine this information with a map system to create an integrated application. The information extraction community, which grew up around the MUC conferences [14], is oriented more towards extracting propositional knowledge from free-form, unstructured data sources. The goal for these techniques is to reconstruct in symbolic form knowledge known by the author and represented explicitly in the text of the web page in question. The field has progressed from hand-constructed extraction rules [20] and [19] to learning extraction rules from a set of data. For example, [9] learns rule-based information extractors to identify the name of a person given their home page. A third approach deals with extracting *relational* knowledge existing on the Web through a combination of web pages and their hyperlink structure. The goal is to look beyond the formatted text on web pages, to learn to identify relations suggested by hyperlinks between pages. In one example, [18] use relational learning to identify advisor-advisee relations between faculty and graduate students using the text and hyperlinks on their web pages.

In our previous work [5, 6], the Web–>KB project has focused on integrating these three types of information gathering for the purpose of constructing relational and propositional symbolic knowledge using the Web as our data source. We used a large set of feature extractors including simple hand-written wrappers, learned information extractors, text classification and relational learning. The research demonstrated that it is possible to discover, with relatively high accuracy, a collection of facts within a specific domain of interest by selective spidering of the Web.

This represents only the first stage of our initially-stated challenge; we must still demonstrate that information extracted from the Web is both accurate and detailed enough to be useful. We aim to first construct such a knowledge base and then perform data mining on it to identify patterns of knowledge that were not explicitly represented as facts on the Web.

In this paper, we detail our current work in creating and using a knowledge base about corporations around the world. Built by spidering both primary and secondary information sources on the Web, we have collected a knowledge base of mostly-true relational and propositional facts on a total of 4312 companies. We have applied several data mining techniques to this knowledge base. Our preliminary results indicate that there is indeed promise in automatically learning new things from the Web. For example, we discover interesting regularities in our data such as "Advertising agencies tend to be located in New York." Such a rule is automatically constructed by extracting and identifying locations and industry sectors of all our companies, and then noticing that companies in the advertising industry disproportionately have locations in New York. The knowledge that we have extracted to date is primarily common sense or known knowledge about companies that are not explicitly represented as facts in our knowledge base. We consider this an appropriate first step in demonstrating the feasibility of this approach. In future efforts, we aim to discover novel relations in our data that are true and meaningful.

Note that the approach to text mining we advocate in this paper stands in significant contrast to what is traditionally termed *text data mining* [10]. Other approaches use the text itself as the strata for performing a mining analysis. For example, [10] has created a system for gene function discovery using medical texts. [2] describe a process for text mining to discover grammatical, morphological and structural rules that hold true for the text in question. In contrast, we do not use the base-level text as input to our mining algorithms, but first build a traditional data mining dataset through the use of a variety of simple and elaborate text feature extractors. Then, we apply fairly traditional data mining algorithms to discover knowledge about the subject of the text.

This paper first describes the features, data sources, and learning algorithms used to construct the corporate knowledge base. Then we present a brief overview of the data mining techniques we employ. We show our initial data mining results. Finally, we discuss our plans for continuing this research with specific suggestions and discussion of future work.

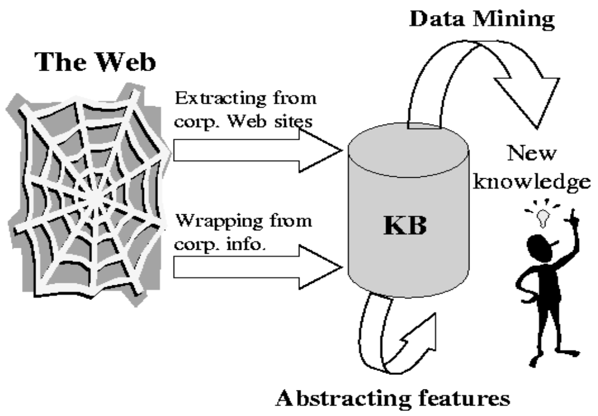

Figure 1: Process of acquiring potentially interesting information about companies from the Web.

2. DATA SOURCES AND FEATURES

Our goal was to assemble a knowledge base containing information about a large number of companies, then use this knowledge base for data mining experiments to explore some general properties of companies and their relationships with each other. This process is illustrated in Figure 1. To begin data collection, we consulted the *Hoovers Online* Web resource (`<www.hoovers.com>`) which contains detailed information about a large number of companies, and selected those companies with home-page URLs listed. These company names and URLs were given to a custom crawler we built for extracting information from company Web sites. This crawler visited 4312 different company Web sites and searched the first 50 Web pages on each site (in breadth first order) for the features listed in Section 2.1. In all our crawler examined just over 108,000 Web pages. To augment this information, we built a wrapper to extract information about each of these 4312 companies from Hoovers. The details of these wrapped features are given in Section 2.2. The complete list of features is given in Table 1.

2.1 Extracted features

The extracted features all come directly from crawling the company Web sites. A variety of techniques, from the simple to the elaborate were used to create them.

At the simple end, the **links-to** and **mentions** features were found using simple text searches on all the web pages, with a pre-defined list of company URLs and names taken from Hoovers. The **officers** were found using a very simple regular expression on any page that contained the word "officer" or "director". The **performs-activity** feature is similar, looking for keywords associated with each type of activity on the Web pages of the company. If the top level domain of the company's home page URL is a country domain, then that country is used as the value for the feature **url-country**.

Text classification methods were used to extract **sector** and **coarse-sector** features. Using an independent set of companies with known **sector** and **coarse-sector** labels, we built a Naive Bayesian model for the sector labels (200 different values) and the coarse-sector labels (12 different values), based on a sample of Web pages from each company's Web site. Naive Bayes as commonly used on text [13] is a standard text classification algorithm which is easy to train and performs quite well. For each company crawled in our new knowledge base, the labels predicted by these models on the pooled Web pages from the company's site were used as the **sector** and **coarse-sector** values.

The **locations** feature was extracted using the most advanced Information Extraction techniques. A Naive Bayes model of regions of text surrounding locations was used as described in [8] in conjunction with phrase-based extraction rules learned from a handful of seed examples using meta-bootstrapping and the AutoSlog system [17].

2.2 Wrapper features from secondary sources

In contrast to the features extracted from the company Web sites, the extractors used to obtain company information from Hoovers could rely on a mostly regular format in which to find the relevant information. Information extractors from such automatically generated text are usually called wrappers.

Various simple wrappers were written to extract the features from the Hoovers' pages for each company. From the Hoovers' *Capsule* page we extracted **hoovers-sector**, **hoovers-industry**, **hoovers-type**, **address** and some of the values for **officers**, **competitor** and **subsidiary**. When available, we extracted values for **products**, **auditors**, **competitors**; and **revenue**, **net-income**, **net-profit** and **employees** for all the years listed.

2.3 Abstracted features

We augmented these extracted features with some new features built from them. Eight of the features we added described relationships between companies based on cross-referencing features, such as **share-officers** and **same-state**, and were attempts to give our data mining algorithms some background knowledge about some relationships we believed might be useful when searching for regularities.

We also added four other features to discretize our continuous features (**revenue**, **net-income**, **net-profit** and **employees**). Mostly these were added to allow one of our data mining algorithms, which could not accept continuous features, to use discretized versions of them.

3. DATA MINING ALGORITHMS

In our experiments to find patterns in our knowledge base, we used several learning algorithms. The following section describes each algorithm and gives some motivation as to why these algorithms would be expected to perform well for our tasks.

Given a dataset of information on companies collected from the Web, our first question is typical in Data Mining contexts: Can we learn

Feature	Values	Description
		EXTRACTED FEATURES
performs-activity	8	The types of activity this company engages in.
links-to		Companies whose web sites are pointed to by this company.
mentions		Companies whose name occurs on this company's Web site.
officers		Officers of this company.
sector	200	Naive Bayes predicted economic sector of company.
coarse-sector	12	Naive Bayes predicted coarse-grained economic sector.
locations		Derived from a naive Bayes classifier on small regions of text surrounding country names [8], and autoslog-based rules [17].
url-country	39	Inferred from the URL domain name where applicable.
		WRAPPED FEATURES
hoovers-sector	28	Sector listed on the company's Hoovers page.
hoovers-industry	298	Industry listed on the company's Hoovers page.
hoovers-type	18	Public, private, school etc.
address		Address as listed on hoovers.
city, state		Extracted from **address**.
competitor		Companies that compete with this company.
subsidiary		Companies listed as subsidiaries of this company.
products	4648	Product categories extracted from the products page.
officers		Officers listed on the Hoovers page.
auditors	266	Company auditors.
revenue		Revenue data for up to the last 10 years.
net-income		Net Income data for up to the last 10 years.
net-profit		Net Profit data for up to the last 10 years.
employees		Number of employees each year for up to the last 10 years.
		ABSTRACTED FEATURES
same-state		Companies in the same state as this company.
same-city		Companies in the same city as this company.
share-officers		Companies that have officers in common with this company.
mentions-same		Companies that mention some company also mentioned by this company.
links-to-same		Companies that link to some company also linked to by this company.
reciprocally-mentions		Companies mentioned by this company, who mention this company.
reciprocally-links		Companies linked to by this company, who link to this company.
reciprocally-competes		Companies listed as a competitor of this company, who list this company as a competitor.
revenue-binned	10	Revenues for each of up to 10 years binned into 10 equal sized bins.
net-profit-binned	10	Net profits similarly binned.
net-income-binned	10	Net income similarly binned.
employees	10	Employees similarly binned.

Table 1: Complete list of features used.

something about the companies represented in our data, are there any interesting, new things to be found there? That motivated the use of an unsupervised algorithm for discovering associations in large datasets described in Section 3.1.

Inspection of the data and the features used by the unsupervised learning resulted in an approach for narrowing the problem by defining potentially interesting target concepts. For instance, one potentially useful and learnable target function is distinguishing between companies from different economic sectors. In order to find regularities for a particular target concept, we used supervised algorithms for learning propositional and first order rules. We were also interested in learning rules that characterize relationships between companies. Some of these relationships would be very naturally captured by first-order rules, generalizing across relationshsips between pairs of companies in our dataset. We hoped to discover rules of the form competitor(A,B) :- sector (A,S), sector (B,S), not links_to (A,B), mentions (A,B).

Our abstracted features described in Section 2.3 were also an attempt to encode some of these kinds of information in ways that would make it easier for propositional relational learners to capitalize on them.

3.1 Discovering associations

In order to find associations in the data, we first discretized all the continuous features and then mapped each feature to as many Boolean features as it has distinct values. In this way, we ended with about 26 000 features and examples represented with sparse vectors.

Using these Boolean features to represent our data, we generated association rules by applying the Apriori algorithm [1] using the publicly available implementation [4], a version of which is incorporated in the commercially available data mining package "Clementine" [21]. In a typical datamining setting, it is assumed that there is a finite set of literals (usually referred to as items) and each example is some subset of all the literals. The Apriori algorithm performs efficent exhaustive search by using dynamic programming and pruning the search space based on the parameters given by the user for minimum support and confidence of rules. This algorithm has been widely used for mining association rules over "basket data", where literals are all the items in a supermarket and examples are transactions (specific items bought by customers).

An association rule is an implication of the form $X \to Y$, where X and Y are subsets of literals and $X \cap Y = \phi$. We say that the rule holds with *confidence c* if *c*% of examples that contain X also contain Y. The rule is said to have *support s* in the data if *s*% of examples contain $X \cup Y$. In other words, we can say that for the rule $X \to Y$, support estimates $P(X,Y)$ and confidence estimates $P(Y|X)$.

3.2 Learning propositional rules

Decision trees have often been used in Data Mining tasks such as finding cross-selling opportunities, performing promotions analysis, analyzing credit risk or bankruptcy, and detecting fraud. We use the C5.0 algorithm (an extension of C4.5 proposed by Quinlan(1993)) which generates a decision tree for the given dataset by recursive partitioning of the data. The particular implementation of C5.0 we use is part of the commercially available data mining package "Clementine". In our experiments, we use the Information Gain ratio as the splitting/selection criterion and perform pruning at different levels. Since our goal is to discover patterns in our data and not to classify or predict unseen instances, we derive a rule set from a decision tree by writing a rule for each path in the decision tree from the root to a leaf.

3.3 Learning relational rules

We are searching for regularities in a relational knowledge base, and thus able to benefit from using a relational learner. Quinlan's FOIL system [15, 16] is a greedy covering algorithm for learning function-free Horn clauses. FOIL induces each Horn clause by beginning with an empty tail and using a hill-climbing search to add literals to the tail until the clause covers only (mostly) positive instances. The evaluation function used for the hill-climbing search is an information-theoretic measure.

By using the relational description of companies in our knowledge base directly, FOIL can use patterns in the relationships between companies in its search for interesting regularities. This is in contrast with association rules and decision trees which are confined to using the propositionalised versions of our knowledge base. As with decision trees, we used FOIL in a classification-oriented approach to data mining, giving it target concepts to learn.

4. EXPERIMENTAL RESULTS

Experiments were performed using the features given in Section 2 (in some cases using a subset of the features) and the three algorithms described in Section 3. Since we are looking for interesting regularities in the data, we evaluated the generated models based on the coverage of the training examples and by checking the content of the models.

Our first set of experiments aimed at discovering associations in the data using association rules as described in Section 3.1. The second set of experiments involved deciding on the target relation in order to find rules describing the target concept. We selected the following target relations: **hoovers-sector, hoovers-type, auditors, competitor, share-officers, country, and state**. We generated propositional rules using Decision trees (see Section 3.2) and first order rules using the first order rule learning system (see Section 3.3). The findings are described in the rest of this Section.

4.1 Apriori Experiments

Using association rules as described in Section 3.1. we generated rules using all but the continuous features. Using the default parameter setting (minimal support of a rule set to 10% and minimal confidence of a rule set to 80%) we obtained 2658 association rules. Inspection of the most frequent rules pointed out the need for data cleaning. [7] point out that data cleaning is a significant step in any data-mining application. In our case, where some of our features are known to be noisy, closer inspection of a very high-accuracy rule revealed the source to be systematic error in one of our extractors. ("Human Resources" was mistakenly extracted as an officer of companies in many instances). In this way, the data-mining approach lends itself to a two-phase approach, in which we can also improve our extractors. After removing rules that contained some of the wrongly extracted features, we ended up with 254 association rules. Below are some examples of the rules we found among them. For each rule we give its confidence expressed as percentage of examples containing all the rule features, followed by the percentage of examples for which the association holds (support, confidence).

Among the highest confidence rules are those relfecting associations between our Extracted features. Examples are the following rules that can intuitively be understood as *companies with documentation on their sites, that either are located in USA or provide technical assistance, are involved in sales*.

```
performs-activity=sell :- locations=united-states,
  links-to=adobe-systems-incorporated (10.8%, 93.0%)
performs-activity=sell :- performs-activity=technical-assistance,
  links-to=adobe-systems-incorporated (11.9%, 91.1%)
```

We checked our database for instances linking to **adobe-systems-incorporated**, and confirmed that this is mostly due to web pages linking to PDF files (documentation), and also linking to Adobe to provide visitors with the possibility of reading their documentation (by downloading a PDF acrobat reader).

A second interesting regularity is that in our data, **most companies located in Japan either sell or perform research** (see the two rules below), while companies located in USA either sell or supply.

```
performs-activity=sell :- locations=japan (14.5%, 90.8%)
performs-activity=research :- locations=japan (13.2%, 82.2%)
```

We confirmed that about one third of our companies are located in USA (37%) and among them about 70% perform research (verified by running with a lower confidence threshold than our default).

```
performs-activity=research :- locations=united-states (26.9%, 72.5%)
```

Running the association rules algorithm with lower support and confidence (support 5%, confidence 50%), revealed that **companies mentioning software on their Web pages are mostly located in the USA**. We also found that companies performing sales, supply and research that have documentation (link to adobe-systems-incorporated) on their Web pages are probably (61.2%) located in USA (the second rule below). The third rule below can be intuitively understood as **most companies in the technology sector are located in the USA**.

```
locations=united-states :- performs-activity=supply,
  performs-activity=expertise, mentions=software (5.3%, 64.0%)
locations=united-states :- performs-activity=sell,
  performs-activity=supply, performs-activity=research,
  links-to=adobe-systems-incorporated (7.0%, 61.2%)
locations=united-states :- performs-activity=supply,
  coarse-sector=technology-sector (5.8%, 50.1%)
```

Association rules involving financial features (revenue, income and profit) showed that **most of the companies in our dataset are stable in their finances**. For instance, the following rule shows that a company with high revenue in 1993–1996 is highly probable (99.5%) to have a high revenue again in 1997.

```
revenue-1997=high :- revenue-1996=high, revenue-1995=high,
  revenue-1994=high, revenue-1993=high (5.0%, 99.5%)
```

In order to get some more associations with lower confidence for some features we considered especially interesting, we reduced the feature set to the following four features: **url-country, hoovers-sector, competitor** and **auditors**. After transforming them to Boolean features we had 3532 features. Running the association rules algorithm on this reduced set of features with low support and confidence (support 1%, confidence 10%) resulted in 38 rules with this support and confidence or higher.

Suprisingly, there is an association between **auditors** and **hoovers-sector**. The following rules give three conclusions supported by about 1-2 % of our data. First, **companies in computer-software-&-services have Pricewaterhouse Coopers (20.9%) or Ernst & Young (14.3%) as their auditor**. Second, **companies in diversified-services have Price-Waterhouse Coopers (15.7%) or Arthur Andersen (13.9%) as their auditor**. Third, **companies in drugs have Ernst & Young (26.8%) as their auditor**.

```
auditors=pricewaterhousecoopers-llp :-
  hoovers-sector=computer-software-&-services (1.7%, 20.9%)
auditors=ernst-&-young-llp :-
  hoovers-sector=computer-software-&-services (1.2%, 14.3%)
auditors=pricewaterhousecoopers-llp :-
  hoovers-sector=diversified-services (1.2%, 15.7%)
auditors=arthur-andersen-llp :-
  hoovers-sector=diversified-services (1.1%, 13.9%)
auditors=ernst-&-young-llp :-
  hoovers-sector=drugs (1.0%, 26.8%)
```

We can also see associations between **hoovers-sector** and **competitors** as follows. **About half of the companies that compete with microsoft-corporation are in computer-software-&-services** (the first rule) **and about a quarter of companies that are in computer-software-&-services compete with microsoft-corporation**.

```
hoovers-sector=computer-software-&-services :-
  competitor=microsoft-corporation (2.1%, 54.9%)
competitor=microsoft-corporation :-
  hoovers-sector=computer-software-&-services (2.1%, 25.7%)
```

The following rules show a competitor which is a good predictor for different **hoover-sectors** supported by about 1% of our companies. They can be understood as follows: **most companies competing with Conagra inc., KMart Corporation and BP Amoco p.l.c. are in food-beverage-&-tobacco, retail and energy, respectively**.

```
hoovers-sector=food-beverage-&-tobacco :-
  competitor=conagra-inc (1.0%, 89.8%)
hoovers-sector=retail :-
  competitor=kmart-corporation (1.0%, 75.0%)
hoovers-sector=energy :-
  competitor=bp-amoco-p.l.c. (1.1%, 73.0%)
```

4.2 Decision Trees

Association rules allow us to find arbitrary associations between many features, at the cost of representational complexity. If we are willing to decide on a target function per run, a decision tree learner can explore more complex rules. In the decision trees shown in this section, the first number in brackets refers to the number of examples covered by the rule. The second shows the fraction of them which have the target label shown.

We learned a decision tree to predict the economic sector as described by Hoovers. One of our predictors was a naive Bayes classifier for economic sector, using both a coarse and finer-grained classification which were not identical to Hoovers'. Note that this approach could allow us to improve on the accuracy of a classifier based on the web pages by using other features. It also allows us to learn a mapping between similar features derived from different sources, which can then permit the two features to be used identically.

```
city Atlanta
  revenue1996 =< 0.1   -> Diversified Services (28, 0.179)
  revenue1996 > 0.1   -> Computer Software & Services (20, 0.2)
city Houston
  coarse-sector [basic-materials, capital-goods, transportation]
                                  -> Manufacturing (10, 0.3)
  coarse-sector [financial, healthcare, technology]
                     -> Computer Software & Services (21, 0.238)
  coarse-sector [conglomerates, consumer-cyclical,
    consumer-non-cyclical, energy, services, utilities]
                                        -> Energy (49, 0.49)
city Dallas
  net_income1999 =< 19   -> Health Products & Services (25, 0.2)
  net_income1999 > 19   -> Leisure (25, 0.2)
city Minneapolis
  employees1996 =< 2.4   -> Diversified Services (23, 0.174)
  employees1996 > 2.4   -> Manufacturing (20, 0.3)
```

Figure 2: Partial decision tree for Hoovers sector using combination of learned features extracted from web-pages and symbolic features wrapped from the Hoovers web-site.

```
coarse-sector [utilities]   -> Utilities (69, 0.623)
...
coarse-sector [energy]
  hoovers_type NIL   -> Energy (39, 0.897)
...
  hoovers_type Public
    employees1993 =< 4.6   -> Energy (42, 0.357)
    employees1993 > 4.6   -> Telecommunications (20,0.4)
coarse-sector Services
  sector Communications-services
      net_income1999 =< 1.8   -> Media (38, 0.342)
      net_income1999 > 1.8   -> Telecommunications (33,0.333)
...
coarse-sector Technology
  sector Waste-management-services
      net_income1998 =< 4   -> Computer Software
                                  & Services (38, 0.421)
      net_income1998 > 4   -> Diversified Services (22,0.227)
coarse-sector Financial
  revenue1992 =< 4.5
    sector Investment-services
      employees1993 =< 0   -> Financial Services (35,0.429)
      employees1993 > 0   -> Banking (23, 0.739)
  revenue1992 > 4.5   -> Financial Services (62, 0.548)
coarse-sector Transportation
  sector Misc-transportation
    revenue1996 =< -1.2   -> Telecommunications (21,0.286)
    revenue1996 > -1.2
      net_income1996 =< 22   -> Diversified Services (31, 0.323)
      net_income1996 > 22   -> Manufacturing (22,0.227)
  sector Railroad
    employees1999 =< 36.4
      net_income2000 =< 3.1   -> Media (33, 0.152)
      net_income2000 > 3.1   -> Drugs (21, 0.238)
    employees1999 > 36.4   -> Transportation (49,0.184)
```

Figure 3: Partial decision tree for Hoovers sector using combination of learned features extracted from web-pages and symbolic features wrapped from the Hoovers web-site. For this tree we excluded the city feature to focus on learning rules to improve our web-page based sector classifiers.

The resulting decision tree is shown in Figure 2. Interestingly, depending on the city the company is located in, different features are then used to predict the sector. For Atlanta, computer companies have a higher revenue than diversified services companies (same for Chicago; not shown). For Houston, depending on the coarse-sector (based on noisy Naive Bayes classification of the company web-pages), we predict either Manufacturing, Computer Software & Services, or Energy. For Dallas, most Health companies are non-profit and thus have a lower income than leisure companies.

Next we excluded the city feature to focus on learning rules to improve our web-page based sector classifiers. The resulting decision tree is shown in Figure 3. Note that Telecommunications has more employees than Energy and can help weed out incorrect classifications in the coarse-sector prediction for energy. Where the Naive Bayes classifier predicts communications-services, income can be used to distinguish between Media (lower-income) and Telecommunications (high). Where the Naive Bayes classifier predicts investment-services, employees can be used to distinguish between Financial Services(lower) and Banking (high). This decision tree also finds irregularities in the Naive Bayes predictions for the transportation

```
US Company =< 0    -> Public (932, 0.838)
US Company > 0
  net_profit2000 =< 0
    revenue1998 =< 0.2
      hoovers_sector Aerospace/Defense    -> Subsidiary (6, 0.5)
      hoovers_sector
        Computer Software & Services    -> Private (51, 0.725)
      hoovers_sector Drugs    -> Private (14, 0.571)
      hoovers_sector Financial Services    -> Private (39, 0.564)
      hoovers_sector
        Food Beverage & Tobacco    -> Private (62, 0.629)
      hoovers_sector
        Health Products & Services    -> Not-for-Profit (40, 0.475)
      hoovers_sector Leisure    -> Private (84, 0.679)
...
      hoovers_sector
        Telecommunications    -> NIL (28, 0.536)
      hoovers_sector Diversified Services
  sector Immigration-law    -> Foundation (24, 0.333)
  sector International-law    -> Partnership (18, 0.833)
  sector Maritime-law    -> Partnership (11, 0.909)
```

Figure 4: Partial decision tree for Hoovers type using combination of learned features extracted from web-pages and symbolic features wrapped from the Hoovers web-site.

sector (last rule in the tree).

Our next decision tree target function was hoovers_type, which attempts to learn rules to predict Hoovers' classification of companies into Private, Public, Not for Profit, etc. We defined the feature US Company for this decision tree, which is defined to be a company whose address given by Hoovers is a state in the US. The resulting tree is shown in Figure 4. In our data-set, the bulk of non-US companies are publicly traded. For US companies, those in the health services sector are non-profit, while others with low profit and revenue are private. In addition, unless the sector is Diversified Services, then if predicted sector predicted by the naive Bayes classifier is law(immigration,maritime) then either it is either a foundation, or a partnership.

The final decision tree learning task we undertook was to predict a composite feature hq-state_country, for which posible values are all US States, and Country names, as defined in the Hoovers address information. One of our predictors was url-country, which we derive from the company's URL, if it is indicative of a country. This is derived from the internet standard RFC 1591 based on ISO 3166 two-letter country codes. Our decision tree validates this extraction method, showing that it always correctly predicts companies headquartered in Australia, Japan and the United Kingdom. When the URL did not provide us with the url-country the tree uses other features to predict hq-state_country. The first feature selected in hoovers-type; when this is NIL the tree uses the hoovers_industry feature.

Note the preponderance of medical companies and industries in California, and the fact that high-profit technology companies are based in Massachusetts (perhaps well established companies) while lower profit companies (say, start-ups) are more likely to be headquartered in California. Our data-mining also reveals the locations of banking centers around the US, as well as picking up on expected correlations such as gambling in Nevada, oil in Texas, high-tech industries in California, and banking, fashion and advertising in New York. The tree is shown in Figure 5.

4.3 FOIL Experiments

Propositional rules using FOIL were used to investigate learning rule-sets for two broad classes of target function. The first, and simpler computationally, class were unary relations. Specifically, we chose to learn rule-sets for each value of **hoovers-sector** and **auditors** of each company.

```
url-country AU    -> Australia (13, 1.0)
...
url-country JP    -> Japan (140, 1.0)
url-country UK    -> United Kingdom (67, 1.0)
url-country NIL
hoovers_type Cooperative    -> CA (27, 0.148)
  hoovers_type Division of    -> CA (15, 0.333)
  hoovers_type Government-owned    -> CA (23, 0.174)
  hoovers_type Joint Venture of    -> NY (15, 0.2)
  hoovers_type Mutual Company    -> Canada (10, 0.2)
  hoovers_type Not-for-Profit    -> TX (42, 0.143)
  hoovers_type Partnership    -> NY (45, 0.356)
  hoovers_type Private    -> CA (682, 0.196)
  hoovers_type Public    -> Canada (395, 0.301)
  hoovers_type School    -> TX (13, 0.231)
  hoovers_type Subsidiary    -> CA (218, 0.165)
...
  hoovers_type NIL
    hoovers_industry Advertising    -> NY (7, 0.429)
    hoovers_industry Aerospace/Defense - Products    -> FL (15, 0.2)
    hoovers_industry Agricultural Operations & Products    -> CA (8, 0.5)
    hoovers_industry Apparel - Clothing    -> NY (15, 0.467)
    hoovers_industry Banking - Mid-Atlantic    -> MD (10, 0.4)
    hoovers_industry Banking - Midwest    -> IL (37, 0.216)
    hoovers_industry Banking - Northeast    -> PA (30, 0.367)
    hoovers_industry Banking - Southeast    -> GA (29, 0.276)
    hoovers_industry Banking - Southwest    -> TX (7, 0.714)
    hoovers_industry Banking - West    -> CA (27, 0.704)
    hoovers_industry Biotechnology - Medicine    -> CA (62, 0.371)
    hoovers_industry Biotechnology - Research    -> CA (8, 0.625)
    hoovers_industry Corporate Professional & Financial Software    -> CA (43, 0.209)
    hoovers_industry Engineering Scientific & CAD/CAM Software    -> CA (10, 0.6)
    hoovers_industry Gambling Resorts & Casinos    -> NV (12, 0.667)
    hoovers_industry Investment Banking & Brokerage    -> NY (20, 0.4)
    ...
    hoovers_industry Medical Appliances & Equipment    -> CA (43, 0.349)
    hoovers_industry Medical Instruments & Supplies    -> CA (28, 0.286)
    hoovers_industry Networking & Communication Devices    -> CA (25, 0.52)
    hoovers_industry Oil & Gas Exploration & Production    -> TX (34, 0.441)
    hoovers_industry Oil & Gas Services    -> TX (18, 0.722)
    hoovers_industry Semiconductor - Integrated Circuits    -> CA (11, 0.636)
    hoovers_industry Semiconductor - Specialized    -> CA (11, 0.636)
    hoovers_industry Semiconductor Equipment & Materials    -> CA (27, 0.444)
    hoovers_industry Wireless Satellite & Microwave Communications Equipment    -> CA (17, 0.412)
    hoovers_industry Information Technology Consulting Services
      net_profit1996 =< 0.5    -> CA (23, 0.261)
      net_profit1996 > 0.5    -> MA (27, 0.185)
```

Figure 5: Partial decision tree for Hoovers state and country using combination of URL based predictor, and symbolic features wrapped from the Hoovers web-site. Uses type of company as a feature, to produce a correspondence between US-states and industry sectors.

For **hoovers-sector** we found a rule that can intuitively be read as **companies headquartered somewhere other than Fremont competing with "Computer Associates International" are in the computer software & services sector**.

```
computer-software-&-services(A) :- hq-city(A,B),
  B<>fremont, competitor(A,C),
  hq-city(C, islandia), not(employees_binned(A,?,?)).
```

In our knowledge base, this rule is correct for 51 companies and does not match any other companies. A little further investigation reveals that "Computer Associates International" is the only company in our knowledge base headquartered in Islandia.

A **hoovers-sector** rule with lower coverage, but drawing on some of our extracted features, covers 8 companies correctly and none incorrectly in our knowledge base. The rule can intuitively be understood as **companies headquartered in New York, that are not in natural-gas-industry nor technology-sector, are in the media industry.**

```
media(A) :- hq-city(A,new-york), sector(A,B),
  B<>natural-gas-industry, coarse-sector(A,C),
  C<>technology-sector, competitor(?,A),
  performs-activity(A,?),not(products(A,?)), not(locations(A,?)).
```

Two other **hoovers-sector** rule use all three kinds of features, namely **sector** and **locations** (extracted from web-pages), **auditors** (from wrapped Hoovers web-site), and **reciprocally-competes** (an Abstracted feature). They all use the learned **sector** Naive Bayes model and refine it with knowledge about type of company or company auditors.

Note that the unbound variable in locations(A,?) and reciprocally-competes(A,?) can be read as **had a location we extracted from the web-site** and **has a company listed on its Hoovers pages that also lists it as a competitor**. (Note that not all Hoovers competitor relationships are reciprocal in this way). The first rule matches 26 companies correctly and one incorrectly, while the second matches eight companies correctly and none incorrectly.

```
metals-&-mining(A) :-
  sector(A,gold-and-silver-industry), locations(A,?),
  type(A,public).
retail(A) :- sector(A,retail-apparel-industry),
  reciprocally-competes(A,?),
  auditors(A,deloitte-&-touche-llp).
```

Next we learned rule-sets to predict the **auditors** of a company. The highest coverage rule we found matched only four companies in our dataset, but all of them correctly. It can be intuitively understood as **companies headquartered in Madrid having listed historical financial information use Arthur Andersen as their auditor.**

```
arthur-andersen(A) :- hq-city(A,madrid), net_profit(A,?,?).
```

Finally, we attempted to learn some binary relations. This presented some practical difficulties (due to algorithm complexity) and also turned up some problems in our knowledge base. Our first binary target relation was **competitor**. To cut down the computation, we used only **hq-city**, **url-country**, **links-to** and **hoovers-sector** as background features. Reassuringly, that simple run discovered the following regularity matching 11407 companies correctly and none incorrectly. In English, this rule states that **two companies in the same sector are competitors**.

```
competitor(A,B) :- A<>B,
  hoovers-sector(A,C), hoovers-sector(B,C).
```

5. DISCUSSION

We have demonstrated that we can discover interesting regularities about companies by extracting, and then mining information on the Web. However, difficulties arose in this process that are deserving of note and discussion. One difficulty we encountered was in the errorful nature of our facts. Most traditional processes of data mining include an extensive phase of data cleaning. In our scenario, data cleaning was more problematic than usual because we have additional sources of noise from the imperfection of our feature extractors. For example the **company-mentions-company** relation was over-populated by matching on such generic shortened company names such as “The Limited”. Our data cleaning/mining went through several iterations, where our mining algorithms would discover regularities that were clearly a result of insufficient data cleaning. When automatically constructing knowledge bases with imperfect extractors, the data cleaning effort will necessarily be of this iterative pattern.

Additionally we note the need for feature selection, especially for relational learning. Both the memory usage and run time of the FOIL algorithm proved to be problematic for the size of our extracted data set. Additionally, several of the features were prominent in terms of number of literals, but low on content. These suggest the need for feature selection techniques. One possibility is to perform a two-pass feature selection and learning process. First, select relatively simple, unary target relations to learn. This allows rule learning algorithms to perform efficiently, as many fewer constructed negative examples are required. The results of this first-pass learning will suggest a subset of features that are useful for data mining. Thus, we use the results of the first-pass learning not for the rules themselves, but to suggest the features to use. Then using only a subset of the features, run the expensive, binary relation discovery. This process proved effective for our use of FOIL.

One result we were pleased to observe was the interaction between the symbolic features and the statistically-derived (naive Bayes) features. Based on the text of a company’s web pages, the **sector** feature predicts an industry sector. However, as shown in Section 4.2, learning the **Hoovers-sector** involved more than just mapping from **sector** to **Hoovers-sector**. The decision tree was able to identify regions of the classification space for which naive Bayes was a poor predictor, and correct for it with the use of symbolic features. This paradigm of combining statistical and symbolic features may prove useful as there is often a collection of both symbolic and text data without a clear method of combination. Additionally, this points the way for deriving values for features such as **Hoovers-sector** for companies which are too small to be listed on corporate information web sites.

6. FURTHER WORK

The results described in this paper suggest a number of research directions, impacting each of information extraction, machine learning, and data-mining from text. The use of disparate knowledge sources lends itself to improvement of the less accurate features through the use of the more accurate ones. For information extraction, we could use the information from wrapped web-sites as a source of training data to improve our extractors. This could be beneficial both at the sentence level, giving us a way of labeling the corpus we build from crawling a company web-site, and at the web-site classification level, giving us a way of adapting a text classifier trained on a slightly different training set. In addition we can augment our extractors to operate on both the text and symbolic features, providing meta-extractors that look not only at web pages, but also use background knowledge.

An additional direction is greater automation of the data-cleaning of extracted features. We discovered anomalies in a relatively ad hoc manner during the work described here. By running data-mining as a form of sanity-check at the time of extractor construction, we can detect and hence avoid errors which are rare in general, but which occur frequently in a large enough collection. This augments any testing we do using a pre-labeled test set, since it permits systematic error testing in enormous, previously unseen sets of data.

Some of the actual information extraction we performed was at the level of keyword spotting. Extending this to use machine learning on data either hand-labeled, or labeled in a semi-supervised manner using Hoovers data can provide richer and more reliable features. For data-mining, we found that the unsupervised exploratory approach of Apriori was attractive, but weak in representation. Combining unsupervised search with a decision tree or relational learner could give us greater power in data-mining. Such an approach would need to be both incremental and iterative, incorporating feature selection as a sub-task, in order to render it computationally tractable. Due to time constraints we did not run our crawler on all companies represented on the Hoovers web-site. We could also run these algorithms on companies not found on Hoovers, by running the crawler more generally. Coupled with data cleaning, we may be able to perform better data-mining, with much the same experimental set-up. In addition, since we have learned rules which predict certain features in the absence of others, it would be instructive to try using those rules to relabel certain parts of our data, and re-run the mining algorithms. In this way we can view our knowledge-base at any period in time as a collection of knowledge in flux, as we gain better and better understanding of patterns that underly the data.

7. REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press*, pages 307–328, 1996.

[2] H. Ahonen, O. Heinonen, M. Klemettinen, and A. I. Verkamo. Applying data mining techniques in text analysis. In *Report C-1997-23, Dept. of Computer Science, University of Helsinki*, 1997.

[3] G. Barish, C. A. Knoblock, Y.-S. Chen, S. Minton, A. Philpot, , and C. Shahabi. Theaterloc: A case study in information integration. In *IJCAI Workshop on Intelligent Information Integration*, Stockholm, Sweden, 1999.

[4] C. Borgelt. apriori. `http://fuzzy.cs.Uni-Magdeburg.de/~borgelt/`.

[5] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the Fifteenth National Conference on Artificial Intellligence (AAAI-98)*, pages 509–516, 1998.

[6] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 2000.

[7] J. S. Deogun, V. V. Raghavan, A. Sarkar, and H. Sever. Rough sets and data mining: Analysis of imprecise data. In T. Y. Lin and N. Cercone, editors, *Data mining: Trends in research and development*, pages 9–46. Kluwer Academic, 1996.

[8] D. DiPasquo. Using html formatting to aid in natural language processing on the world wide web. Master's thesis, Carnegie Mellon University, 1998. http://www.cs.cmu.edu/ webkb/danthesis.ps.gz.

[9] D. Freitag. *Machine Learning for Information Extraction in Informal Domains*. PhD thesis, Carnegie Mellon University, 1999.

[10] M. Hearst. Untangling text data mining. In *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.

[11] C. Knoblock, S. Minton, J. L. Ambite, N. Ashish, P. Modi, I. Muslea, A. G. Philpot, and S. Tejada. Modeling web sources for information integration. In *AAAI-98*, 1998.

[12] N. Kushmerick. *Wrapper Induction for Information Extraction*. PhD thesis, University of Washington, 1997.

[13] T. M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Inc., 1997.

[14] *MUC-4 Proceedings*, San Mateo, CA, 1992. Morgan Kaufmann.

[15] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–2666, 1990.

[16] J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. In *Proceedings of the European Conference on Machine Learning*, pages 3–20, Vienna, Austria, 1993.

[17] E. Riloff and R. Jones. Learning Dictionaries for Information Extraction Using Multi-level Boot-strapping. In *Proceedings of the Sixteenth National Conference on Artificial Intellligence (AAAI-99)*, pages 1044–1049. The AAAI Press/MIT Press, 1999.

[18] S. Slattery and M. Craven. Combining statistical and relational methods for learning in hypertext domains. In *Proceedings of the 8th International Conference on Inductive Logic Programming (ILP-98)*, 1998.

[19] S. Soderland, D. Fisher, and W. Lehnert. Automatically learned vs. hand-crafted text analysis rules. Technical Report TC-44, University of Massachusetts, Amherst, CIIR, 1997.

[20] S. Soderland and W. Lehnert. Wrap-up: A trainable discourse module for information extraction. *Journal of Artificial Intelligence Research (JAIR)*, 2:131–158, 1994.

[21] SPSS. Clementine. `http://www.spss.com/clementine/`.