Fixing Weakly Annotated Web Data Using Relational Models

Fatih Gelgi, Srinivas Vadrevu, and Hasan Davulcu

Department of Computer Science and Engineering,
Arizona State University,
Tempe, AZ, 85287, USA
{fagelgi,svadrevu,hdavulcu}@asu.edu

Abstract. In this paper, we present a fast and scalable Bayesian model for improving *weakly annotated data* – which is typically generated by a (semi) automated information extraction (IE) system from Web documents. Weakly annotated data suffers from two major problems: they (i) might contain incorrect ontological role assignments, and (ii) might have many missing attributes. Our experimental evaluations with the TAP and RoadRunner data sets, and a collection of 20,000 home pages from university, shopping and sports Web sites, indicate that the model described here can improve the accuracy of role assignments from 40% to 85% for template driven sites, from 68% to 87% for non-template driven sites. The Bayesian model is also shown to be useful for improving the performance of IE systems by informing them with additional domain information.

Keywords: Weakly annotated data, information extraction, classification, Bayesian models.

1 Introduction

Recent years have witnessed a huge data explosion on the Web. All kinds of commercial, government and scientific organizations have been publishing their data to enable better information sharing. However, heterogeneity at the presentation, schema and instance levels makes it extremely difficult to find and relate information from different sources. Recently, there has been some ground breaking work on (meta) data extraction from template driven Web sites using semi-automated techniques [1] and ontology-driven automated data extraction techniques [2] from text. And also, some work exists on fully automated techniques for extracting (meta) data from data rich segments of Web pages [3,4,5].

In this paper, our focus will be on improving *weakly annotated data* (WAD) which is typically generated by a (semi) automated information extraction (IE) system from the Web documents. In WAD, *annotations* correspond to ontological role assignments such as *Concept*, *Attribute*, *Value* or *Noise*. WAD has two

L. Baresi, P. Fraternali, and G.-J. Houben (Eds.): ICWE 2007, LNCS 4607, pp. 385–399, 2007.

major problems; (i) might contain incorrect role assignments, and (ii) have many missing attribute labels between its various entities.

We will use the Web pages in Figure 1 to illustrate WAD that might be extracted using an IE algorithm such as [3,5]. Each of these pages presents a single instance of the 'Digital Camera' concept. In Figure 1(a), attributes such as 'storage media', and values such as 'sd memory card' have uniform and distinct presentation. However, for an automated system it would be extremely difficult to differentiate the 'storage media type' label as an attribute and 'sd secure digital' as its value due to their uniform presentation in Figure 1(b). On the other hand, in Figure 1(c) the attribute 'storage media' does not even exist, but only its value 'sd memory card' has been reported.

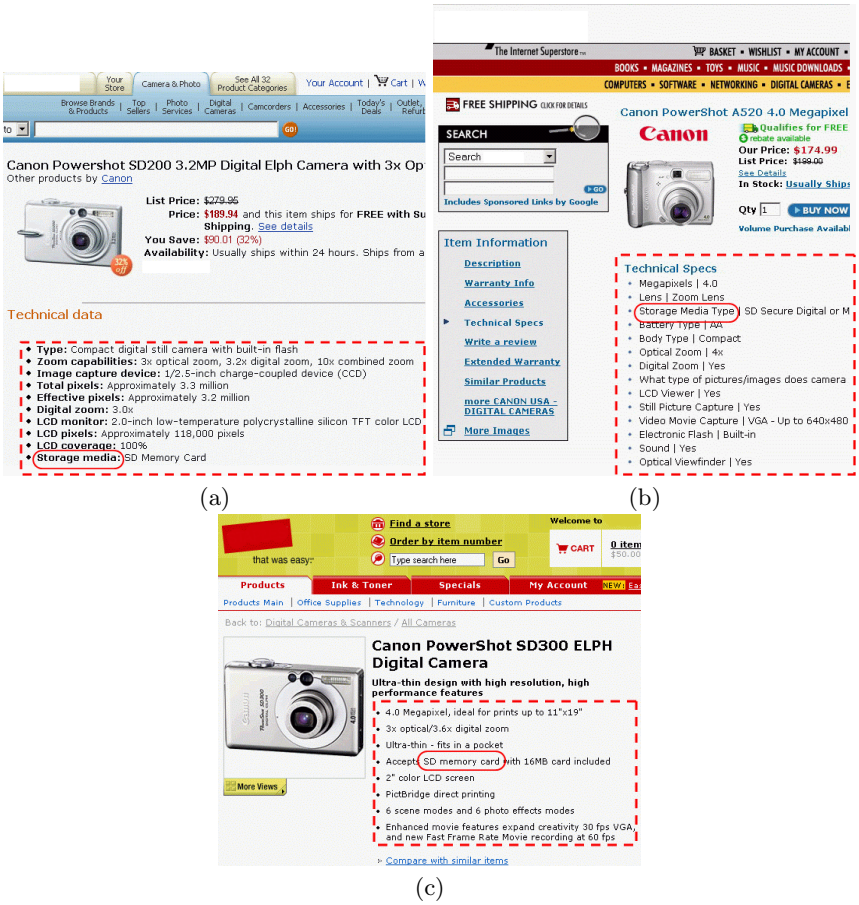


Fig. 1. The instance pages for Canon digital camera from three different Web sites. Digital camera specifications are marked by dashed boxes in each page. In page (a), attributes are explicitly given as bold whereas in (b) they are not obvious. On the other hand, the attributes are altogether missing in (c).

Florescu [6] indicates that schemas in unstructured data can be very rich and they might be difficult to model using DTDs or ER models. Schemas can be derived from the data instead of driving the data generation, or schemas can be a posteriori overlaid existing data. Motivated by the idea, we first extract a *relational graph* from WAD to capture the association strength between labels in the given document collection. This method is based on the identification of the correlation between labels as co-occurrence constraints emphasized in [6]. Then, we build a domain specific probabilistic model that utilizes the extracted relational graph to improve WAD by automatically **correcting the role assignments of the labels** and **discovering their missing attributes**. We also show that the boost up in the accuracy of the WAD provides additional information for the IE systems to improve their performance. Note that since we are working on a huge data set such as the Web, we exploit linear-time algorithms to ensure the speed and the scalability of our methods while not losing much from the quality.

We formulate the role assignment problem as a classification problem. A Bayesian model is used due to the robustness of Bayesian models on classification tasks [7] with large number of features. Since discovery of the Bayesian network dependencies on data is a hard problem [8], we stick to the independence assumption which also ensures the scalability of the proposed model for the Web data.

The distinguishing feature of our model from the standard Bayesian models is the preservation of the structure of the *relational graph* (see Section 2 for details) by incorporating the edge probabilities. Furthermore, the number of features (i.e. tagged labels) are not fixed, as required by conventional classification techniques.

As stated in [9], naive Bayes classifiers perform well on strongly annotated data, i.e., correctly tagged training data, but they are very sensitive to redundant and irrelevant features. Therefore, the excessive amount of irrelevant features and incorrect role assignments inherent in WAD would render a naive Bayes classifier to be ineffective. A subsection is also included to discuss the weaknesses of naive Bayes.

Probabilistic Relational Models (PRMs) [10] are powerful methods to learn the underlying structure of relational data. However, PRMs and other classification approaches on relational data such as [11] assume strongly annotated data, and their scalability is a problem which makes it inappropriate for the Web data.

2 System Overview

In our framework, we assume each label is tagged with one of the four ontological roles listed below;

- *Concept* (C): A concept defines a category or a class of similar items. E.g., 'books' and 'digital camera' are some of the concepts in the shopping domain.
- *Attribute* (A): An attribute is a property of an instance or a concept. E.g., 'storage media' is an attribute of the 'canon powershot sd200' instance and the 'digital camera' concept.

– *Value* (V): A value is a label that provides the value information for an attribute of a certain concept or an instance of a concept. E.g., 'storage media' attribute of the 'canon powershot sd200' instance has the value 'sd memory card'.
– *Noise* (N): Any label that does not have any of the above ontological roles are assigned to be noise. For example, some of the labels in headers, footers or navigation aids, such as 'back to' shown in Figure 1(c) could be annotated as noise.

We assume that we can gather "sufficient statistics" for WAD through a collection of domain specific Web sites. From the automatically extracted data we generate a *relational graph* of the domain where nodes correspond to the labels with assigned roles, and the edges correspond to association strengths between nodes. These annotated labels and relations between them are assumed to be the output of automated IE systems such as RDF files. Such a graph would capture the global occurrence statistics of the labels and their associations within a domain.

In the next phase, for each label in a given Web page, we run a Bayesian classifier that utilizes all the labels in that Web page as its *context* to identify the best role for that label. The advantage of our probabilistic model over a naive Bayes classifier will be discussed in the next section. We refer to our probabilistic model as the *Bayesian classifier* in the rest of the paper.

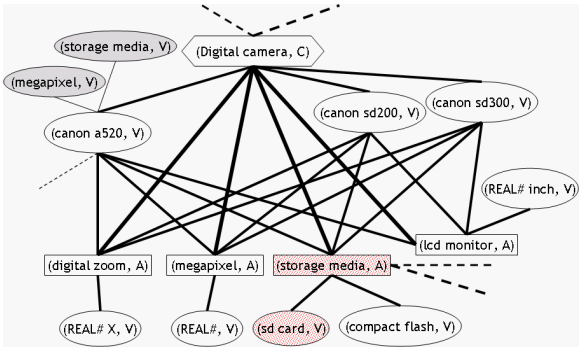


Fig. 2. A fragment of the relational graph for the Shopping domain is shown. Each node is composed of a $\langle label, role \rangle$ pair. The thickness of an edge is proportional to the *association strength* between its nodes.

We will briefly explain how the system operates on the example given in Figure 1. A fragment of the corresponding relational graph is depicted in Figure 2. Consider the label 'storage media' marked in Figure 1. Based on our assumption of "sufficient statistics", a collection of Web pages such as those in Figure 1(a), would yield a strong association between 'canon sd200' as an object and 'storage media' as an attribute. Whereas, the incorrect annotation, extracted from Figure 1(b) would yield a weak association between 'canon a520' as an object and the 'storage media' as a value. Hence, the Bayesian classifier presented here would be able to re-assign the attribute role to the 'storage media' label by using the statistics in the relational graph within its context. Similarly, the 'storage media' attribute of the 'sd card' value which is missing in Figure 1(c), could be inferred by utilizing its context and the model.

3 Probabilistic Model

In this section, we will first present the notations used for the formal description of the model. We will formally define the problem of re-annotating the labels of a Web page and present the probabilistic algorithm. Next, we will define the missing attribute inference problem and propose a solution. We will also explain some of the implementation issues and complexity analysis.

The notation used for formalization is given as follows:

- The set of all labels in the domain is denoted as $\mathcal{L}$.
- The **ontological roles** $\mathcal{R}$ is the set of *Concept*, *Attribute*, *Values* or *Noise*. Formally, $\mathcal{R} = \{C, A, V, N\}$.
- A **term** is a pair $\langle l, r \rangle$ composed of a label l and a role $r \in \mathcal{R}$. In other words, terms are tagged labels in the Web pages. Each label in a Web page is assumed to be tagged with only one of the given ontological roles above.
- In this setting, we consider all the labels in each Web page are tagged with roles, hence we define a **Web page** to be a vector of its terms. Formally, assuming m labels in the Web page $\mathcal{W}$; $\mathcal{W} = \{\langle l_1, r_1 \rangle, \langle l_2, r_2 \rangle, \ldots, \langle l_m, r_m \rangle\}$.
- The **relational graph** $\mathcal{G}$ is a weighted undirected graph where the nodes are the terms in the domain, and the weights on the edges represent the *association strength* between the terms.
- In our framework, the **context** of a label $l \in \mathcal{L}$ in a Web page $\mathcal{W}$ is the Web page $\mathcal{W}$ itself.

The nodes in $\mathcal{G}$ denote the labels with their ontological roles and the edges denote the association strengths between the annotated labels. Node weights are initialized as the counts of the corresponding terms and the edge weights are the counts of the corresponding edges in the document collection. Formally, w_{ij} which is the weight between the terms i and j is initialized as the number of times the edge (i, j) appeared in the entire domain. Similarly, w_i represents the weight of the node i and initialized as the occurrence of the corresponding term in the domain, i.e., term count. Note that the edges are undirected since association strength between labels is a bi-directional measure.

3.1 Label Role Inference

The role of a label depends on its *context*. This context of a label is intuitively defined to be its own Web page. The problem of role assignment for each label can now be formally defined as follows;

Definition 1. *Given a Web page $\mathcal{W}$, the probability of a term $\langle l, r \rangle$ where $l \in \mathcal{L}$ and $r \in \mathcal{R}$ is $P(\langle l, r \rangle | \mathcal{W})$.*

This corresponds to the probability of the classification of l as r to be correct. Then, the role with the maximum probability will be the role assignment for the particular label l that is,

$$\arg\max_{r} P(\langle l, r\rangle | \mathcal{W}). \tag{1}$$

For simplicity we use the *naive assumption* which states that,

Assumption 1. *All the terms in $\mathcal{G}$ are independent from each other but the given term $\langle l, r\rangle$.*

Furthermore, we only utilize the first order relationships of a term in its context, i.e, neighbors of the term in $\mathcal{G}$. One can easily extend the model for higher order relationships however the trade-off is the higher complexity which is undesired for Web data.

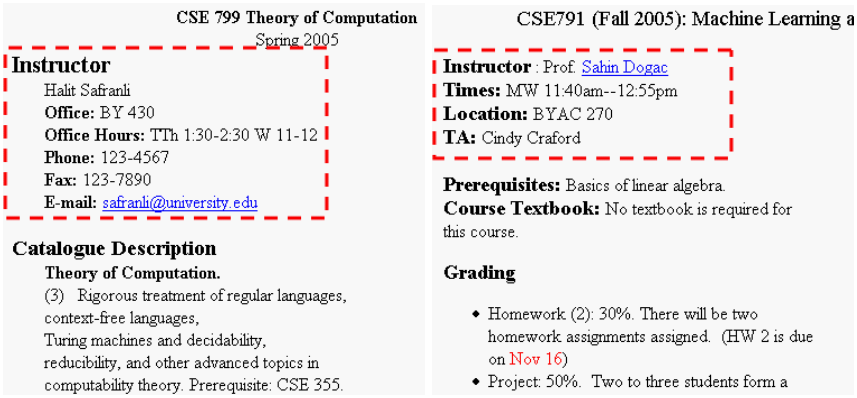


Fig. 3. Ambiguity of the role of the label 'Instructor' in the Courses domain. (a) is its rare occurrence as a concept, and (b) shows its common one as an attribute.

During the role assignment probability calculation of a term, since we would like to utilize only the label's context we also assume,

Assumption 2. *The prior probabilities of all the roles of a label l are uniform.*

Note that, the priors of the roles of the labels other than l in the Web page are their support values as determined by their frequencies. To motivate the idea, consider the label 'Instructor' in Figure 3. 'Instructor' rarely occurs as a concept in the *Courses* domain. Its attributes such as 'Phone', 'Fax', 'E-mail' are also presented in Figure 3(a). However, 'Instructor' usually appears as an attribute of a course in other documents as shown in Figure 3(b). Thus, the prior probability, i.e., $P(\langle Instructor, A\rangle) >> P(\langle Instructor, C\rangle)$, might strongly bias the role assignment towards its more common role. This would yield an incorrect tagging for the label as an attribute in Figure 3(a).

Now, with the above assumptions, we can state the following theorem.

Theorem 1. *Let* $\mathcal{W} = \{t_1, t_2, \ldots, t_m\}$. *Then,*

$$\arg\max_r P(\langle l, r\rangle|\mathcal{W}) = \arg\max_r \prod_{i=1}^{m} P(\langle l, r\rangle|t_i). \tag{2}$$

Proof. Let $t = \langle l, r\rangle$. By Bayes's rule,

$$P(t|\mathcal{W}) = P(t|t_1, t_2, \ldots, t_m) = \frac{P(t_1, t_2, \ldots, t_m|t)P(t)}{P(t_1, t_2, \ldots, t_m)}.$$

Using the independence assumption,

$$= \frac{\prod_{i=1}^{m} P(t_i|t)P(t)}{\prod_{i=1}^{m} P(t_i)}$$

Again using Bayes's rule,

$$= \frac{\prod_{i=1}^{m} P(t|t_i)P(t_i)}{P(t)^m} \cdot \frac{P(t)}{\prod_{i=1}^{m} P(t_i)} = \frac{\prod_{i=1}^{m} P(t|t_i)}{P(t)^{m-1}}$$

By Assumption 2, $P(t)^{m-1}$ will be constant. That is,

$$\arg\max_r P(t|\mathcal{W}) = \arg\max_r \prod_{i=1}^{m} P(t|t_i).$$

□

As shown in Figure 2, a conditional probability such as $P(t|t_i)$ depends on the association strength between the terms t and t_i in the relational graph $\mathcal{G}$. That is, $P(t|t_i) = \frac{P(t,t_i)}{P(t_i)} = \frac{w_{tt_i}}{w_{t_i}}$ by Bayes's rule where w_{tt_i} is the weight of the edge (t, t_i) and w_{t_i} is the weight of the node t_i. Our probability model is based on the methodology of association rules [12]. Hence, the initialization for the above conditional probabilities is defined analogous to $P(t|t_i) \equiv Confidence(t_i \rightarrow t)$ [13]. This formulation is consistent with Assumption 2 since it is independent from the prior, $P(t)$.

3.2 Missing Attribute Inference

In WAD, most of the attribute labels are missing, especially in the non-technical domains and non-template driven Web sites. Discovering missing relations is one of the crucial tasks during automated meta-data extraction. Our probabilistic model can also be tailored to infer some of the missing attributes.

Suppose two related entities have a missing attribute in a Web page. The first entity may be either a concept or an instance of a concept whereas the second one may be a value or a set of values.

Definition 2. *Given two related entities* e_1 *and* e_2 *in a Web page, the probability of a label* $l \in \mathcal{L}$ *to be the attribute between them is* $P(\langle l, A\rangle|\mathcal{S})$ *where* $\mathcal{S} = e_1 \cup e_2$.

Thus, the missing attribute can be inferred by the following formula,

$$\arg\max_{l \in \mathcal{L}} P(\langle l, A \rangle | \mathcal{S}). \tag{3}$$

And, with the same assumptions described above,

Theorem 2. *Let e_1 and e_2 be two entities and $\mathcal{S} = e_1 \cup e_2$. Then,*

$$\arg\max_{l \in \mathcal{L}} P(\langle l, A \rangle | \mathcal{S}) = \arg\max_{l \in \mathcal{L}} \prod_{t \in S} P(\langle l, A \rangle | t). \tag{4}$$

Proof. Follows from the same methodology in the proof of Theorem 1.

□

3.3 Complexity Analysis

Assuming there are n terms and m associations between them, the initialization phase requires only $O(n + m)$ time by utilizing an adjacency list for the relational graph. It is also assumed that it takes $O(1)$ time to map a term to its corresponding node utilizing a hash table. For the rest of the analysis we will assume that we are working on a very large n and $m = O(n)$, i.e., the average number of relations for each label is constant, which is expected from the Web data. Hence, this phase has $O(n)$ time and memory complexity.

For a label with a particular role in a Web page, all conditional probabilities are calculated depending on the other labels in its Web page. That amounts to $O(|\mathcal{W}|)$. Considering all the labels and all roles the total probability calculations for each Web page will be $O(|\mathcal{R}||\mathcal{W}|^2)$. We only need the memory to store the role probabilities for each label in the Web page, that is $O(|\mathcal{W}|)$. Supposing we have p Web pages in a large data set, $O(|\mathcal{R}||\mathcal{W}|^2)$ is also considered to be constant since the size of a Web page is limited and independent from how large p is. In other words, classification phase is $O(p)$ time and memory. Finally, the overall system has $O(n + p)$ time and memory complexity.

While inferring missing attributes, for each $\langle entity, attribute, entity \rangle$ triple, it is not necessary to explore all the attribute labels. Instead, we only check the labels which are related to both entities which takes only constant time (due to the assumption of constant number of relations for each label indicated above). That makes the complexity of attribute inference $O(|\mathcal{W}|)$ for each page $\mathcal{W}$. Similarly, the constant bound on $O(|\mathcal{W}|)$ will yield execution time for entire data set to be $O(p)$. This phase requires only constant memory.

For the entire system, log-probabilities generate a slight overhead that does not change the complexity. In conclusion, the presented probabilistic framework is linear in terms of both the number of terms and the number of web pages thus yielding a fast and scalable model.

3.4 Discussion on Naive Bayes

For the WAD, naive Bayes has entirely different characteristics from our Bayesian classifier. The formulation for the naive Bayes classifier is;

$$\arg\max_r P(\langle l, r\rangle|\mathcal{W}) = \arg\max_r \left[\prod_{i=1}^{m} P(t_i|\langle l, r\rangle)\right] P(\langle l, r\rangle). \tag{5}$$

Naive Bayes uses the reverse conditional probability;

$$P(t_i|\langle l, r\rangle) = \frac{P(t_i, \langle l, r\rangle)}{P(\langle l, r\rangle)}. \tag{6}$$

This violates Assumption 2 since $P(t_i|\langle l, r\rangle)$ is conditioned on $\langle l, r\rangle$. Hence, it would not be able to reason with the context of the label alone – instead relies on the prior probabilities $P(\langle l, r\rangle)$ which yields substantially lower performance as illustrated in our experimental results.

4 Experiments

The descriptions of the three data sets used in the experiments are as follows:

1. **TAP Dataset:** Stanford *TAP Knowledge Base 2* [1] data set. The selected categories alone comprise 9,068 individual Web pages as shown in details in Table 1.
2. **CIPS Dataset:** We prepared a data set which is composed of *faculty*, *course* home pages, *shopping* and *sports* Web pages consisting of 225 Web sites and more than 20,000 individual pages. The computer science department Web sites are *meta-data-driven*, i.e., they present similar meta-data information across different departments. Shopping and sports are some popular attribute rich domains.

Table 1. TAP data set used in experiments

Web sites	# of Web pages	Average # of labels per page
AirportCodes	3829	34
CIA	28	1417
FasMilitary	362	89
GreatBuildings	799	37
IMDB	1750	47
MissileThreat	137	40
RCDB	1637	49
TowerRecords	401	63
WHO	125	21
Overall	9068	200

3. **RoadRunner Dataset:** [3]'s data set comprised of 200 pages in 10 categories.

To overcome the lack of statistics for continuous values, we preprocessed the common data types of values such as percentage, dates, numbers etc. using simple regular expressions. In each experiment, we use the entire data set as training set to exploit global information of the domain, i.e., relational graph of the domain. Context based role inference, i.e., the Bayesian model is based on that relational graph.

The reported accuracies are measured according to the following formula:

$$Accuracy = \frac{\#\ of\ correct\ annotations}{\#\ of\ total\ annotations}$$

4.1 Experiments with the TAP Data Set

To test our probabilistic method, we converted RDF files in the TAP knowledge base into triples, then we applied distortions to obtain the inputs. For synthetic data, we considered real world situations and tried to prepare the input data as similar as possible to the data on the Web. There are two types of distortion: *deletion* and *role change*. Setting the distortion percentages for both deletion and role change first, we used the percentages as distortion probabilities for each tagged label in the Web page in our random distortion program.

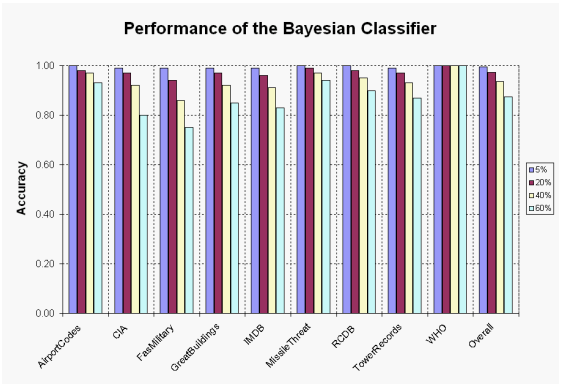
Over TAP data set, we prepared test cases for three kinds of distortions. In the first one, we only applied deletion with different percentages. In the second, similarly we only applied role changing. And the last one is the mixture of the previous two; we applied the same amount of deletions and role changes.

The evaluations of TAP data set are done in automated way assuming the original TAP data set has the correct annotations. We present our results in two categories: (1) individual Web sites and (2) mixture Web sites. For both categories, the Bayesian classifier performed with 100% accuracy for distorted data with only deletions. The reason is, deletion does not generate ambiguity since the initial data is unambiguous. Thus, we found unnecessary to include them in the tables and figures. As a baseline method, we used a naive Bayes classifier.

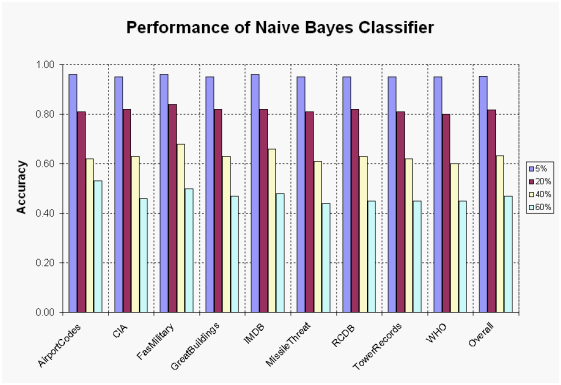
Experiments with the individual Web sites provided us encouraging results to start experiments with mixture of Web sites as shown in Figure 4. The figure displays the final accuracies of Web sites which are initially distorted with {5, 20, 40, 60} percent role changes. Overall results show that there is a huge gap between the Bayesian and the naive Bayes classifiers. Even for 60% role changes the Bayesian classifier performed with more than 85% accuracy. The performance is usually better with the Web sites containing large number of Web pages due to the high consistency and regularity among the Web pages. Another factor is the size of the tagged label set in the Web pages. The larger the set, the more difficult to keep the context concentrated on the related roles in ambiguous

data. That played the most important role for the low performance with *CIA* and *FasMilitary* Web sites and, high performance with *WHO* and *AirportCodes*. On the other hand, naive Bayes slightly improved the initial accuracies as shown in Figure 4(b).

In the experiments with mixture of Web sites, we tested the Bayesian classifier with {5, 20, 40, 60, 80} percent role changes and {(5,5), (15,15), (25,25), (35,35), (45,45)} percent (deletion, role change) distortions. Figure 5 shows the performance of the Bayesian classifier in terms of concept, attribute and value accuracy, and the final Web site accuracies respectively. The overall performance for the mixture Web sites is slightly lower than the individual Web sites due to the fact that mixture Web sites initially have some ambiguities. Similar to the results in the individual Web sites, naive Bayes has not been successful to increase the accuracies of annotations in the mixture ones. The comparison charts in Figure 5(a) and (b), clearly presents the significant difference between the naive Bayes and our Bayesian classifier.



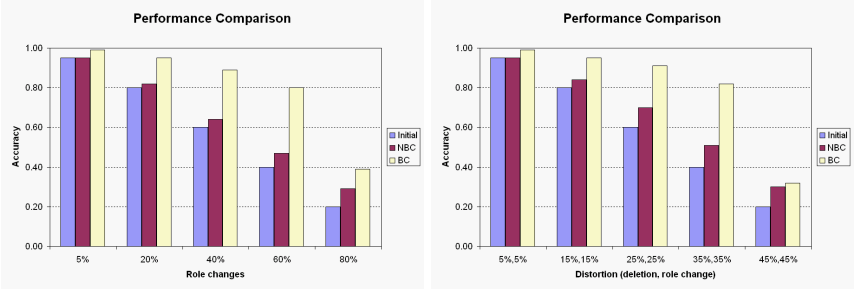
(a)



(b)

Fig. 4. Performance of the Bayesian and the naive Bayes classifiers for label accuracies in individual Web sites are shown

In conclusion, the overall results show that our Bayesian model can recover the TAP data up to 80% even with 60% and (35%,35%) distortion. The results are not surprising since the data set is simple template driven and also the relations are not complicated. In addition, the experimental results strongly support our claims about the unreliability of prior probabilities and weakness of naive Bayes given in Section 3. Verifying the robustness of the system with template driven Web sites, next we will give the experimental results with a non-template driven data set.



(a) (b)

Fig. 5. The comparison of the Bayesian (BC) and the naive Bayes (NBC) classifiers

4.2 Experiments with the CIPS Dataset

Recalling the motivating examples, one can observe that labels are highly ambiguous at both the syntactic and the semantic levels. Many labels have different roles in different Web pages depending on the context.

In this experiment, we used the semantic partitioner [5] to obtain initial annotations of the labels from Web pages. The semantic partitioner system transforms a given Web page into an XML-like structure by separating and extracting its meta-data and associated data. For the evaluations, we created a smaller data set containing randomly selected 160 Web pages from each domain. We divided samples into 4 groups with 40 pages from each category, and each group is evaluated by a non-project member computer science graduate student. The overall accuracy of each category provided in Figure 6 is based on the total accuracy of these sample documents.

The accuracies of initial bootstrapping by the semantic partitioner and of the corrected data by the Bayesian classifier is presented in Figure 6. Since the faculty and course categories are the sub-categories of computer science departments, they presented very similar characteristics as shown in Figure 6(a) and (b). The overall accuracies have been increased roughly from 71% to 89% – a 18% boost. In the shopping domain, although the increment of the value accuracy is similar to the previous two, the overall accuracy jumped up from 69% to 93% – a

Table 2. The results of missing attribute inference over CIPS data set

Category	**CIPS data set**			**Sampled data set**			
	# of sites	# of pages	# of infer.	# of pages	# of infer.	# of crr.	%
Faculty	60	7617	1232	160	38	26	0.68
Course	60	4228	1009	160	49	35	0.71
Shopping	47	3361	4523	160	198	153	0.77
Sports	58	5805	2877	160	82	51	0.62
Total	225	21011	9621	640	367	265	0.72

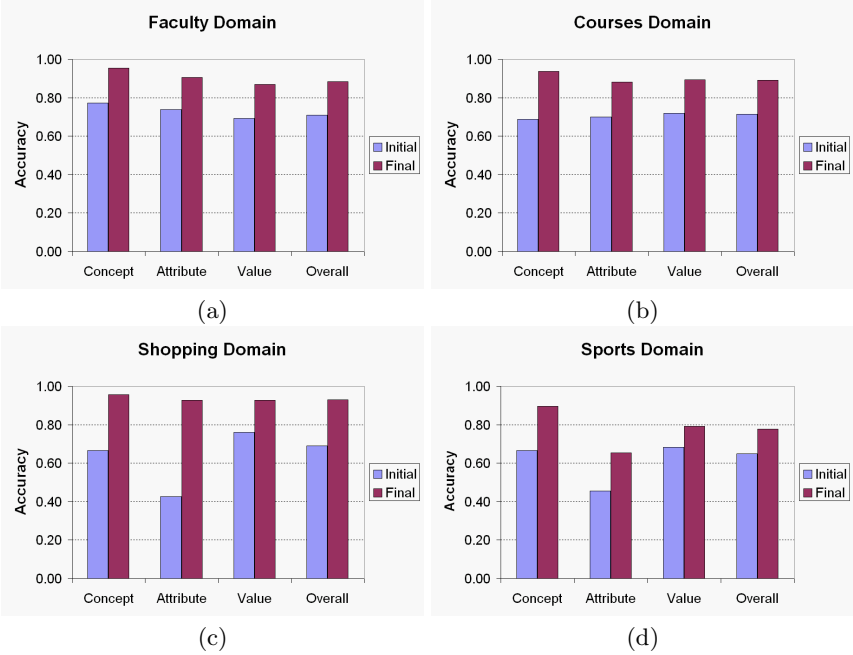


Fig. 6. Performance of the Bayesian classifier in CIPS data set

24% boost. Mistagged concepts and attributes have been corrected with very high accuracy. Initial taggings for meta-data (concepts + attributes) has the lowest accuracy among those four categories since the labels are presented more uniformly than in the other categories as illustrated in Figure 1(b). Fortunately, the labels used for the meta-data in the domain are more common yielding strong statistic thus high recovery accuracy. Conversely, the overall accuracy of the sports category is the lowest among four categories since the jargon of the domain varies much more than the other domains.

The result of the missing attribute inferences in CIPS dataset is presented in Table 2. For the overall data set 9641 attributes have been inferred, 72% of which are correct according to the statistics of the sample data set. The accuracy of the missing attribute inference in the shopping category is slightly better than in the other categories due to the usage of more common meta-data labels as mentioned above. As an example for missing attribute inference, the attributes 'megapixel', 'zoom', 'storage media' and 'lcd' are correctly inferred in the web page in Figure 1(c).

4.3 A Case Study with IE Systems

This experiment was conducted to illustrate the utility of the probabilistic model for improving the performance of an IE system. With the permission of the

authors, we modified the semantic partitioner code. The original semantic partitioner operates by first identifying approximate tandem repeats of presentation information corresponding to the labels within a Web page. Then, it groups and annotates the labels into XML-like hierarchical structures.

The annotations by the semantic partitioner were used as input for the Bayesian classifier. Next, the semantic partitioner was modified to utilize the inferred role probability distributions, in addition to the presentation information so that it can identify more tandem repeats and extract data even in the presence of irregularities. For example in Figure 1(b), the original semantic partitioner fails to distinguish the attributes and values in the 'Technical Specs' area since they are presented similarly. However, the modified semantic partitioner recognizes the roles of these labels as attributes and values, and correctly identifies the repeating sequence of attribute-value pairs.

Table 3. Comparison of the performance the RoadRunner algorithm with semantic partitioner system, before and after utilizing the probabilistic domain model

Classes				**Comparative Results**		
site	description	#pages	metadata	RoadRunner	Sempart (before)	Sempart (after)
amazon.com	cars by brand	21	yes	21	-	21
amazon.com	music bestsellers by style	20	no	-	-	-
buy.com	product information	10	yes	10	10	10
buy.com	product subcategories	20	yes	20	20	20
rpmfind.net	packages by name	30	no	10	10	10
rpmfind.net	packages by distribution	20	no	20	20	20
rpmfind.net	single package	18	no	18	18	18
rpmfind.net	package directory	20	no	-	20	20
uefa.com	clubs by country	20	no	20	-	20
uefa.com	players in the national team	20	no	20	-	-

Table 3 shows the performance of the original and modified semantic partitioner using the public RoadRunner data set. Of the ten categories, the modified semantic partitioner was able to extract information from three additional categories. The modified system was also able to extract information in the two categories (package directory and music bestsellers by style) where RoadRunner system fails since these pages do not follow a regular grammar. The 'uefa' data is organized in terms of complex tables, RoadRunner was able to infer the template by using two sample pages whereas the semantic partitioner (both initial and modified) was unable to extract from such tables using a single page. Overall, the performance of the modified semantic partitioner is better than the original one and it is comparable to the RoadRunner system.

5 Future Work and Conclusion

In this paper, we proposed a fast and scalable probabilistic model to improve the Web data annotations that are generated through (semi) automated IE systems. Our method can be distinguished by its capability of reasoning with contextual

information. Although the initial data contains many incorrect annotations and missing attributes, the Bayesian model presented here was shown to substantially improve the Web data annotations for both template driven and non-template driven Web site collections. We conjecture that the model can be incorporated into IE systems to improve their performance.

The future work includes the formulation of a generic expectation - maximization (EM) framework between an IE system and the Bayesian classifier described here which iteratively improves the annotations.

References

1. Guha, R., McCool, R.: TAP: A semantic web toolkit. Semantic Web Journal (2003)
2. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., McCurley, K.S., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: A case for automated large-scale semantic annotation. Journal of Web Semantics 1(1), 115–132 (2003)
3. Crescenzi, V., Mecca, G.: Automatic information extraction from large web sites. Journal of ACM 51(5), 731–779 (2004)
4. Lerman, K., Getoor, L., Minton, S., Knoblock, C.: Using the structure of web sites for automatic segmentation of tables. In: ACM SIGMOD, Paris, France, pp. 119–130. ACM Press, New York (2004)
5. Vadrevu, S., Gelgi, F., Davulcu, H.: Semantic partitioning of web pages. In: WISE, New York, NY, USA, pp. 107–118 (2005)
6. Florescu, D.: Managing semi-structured data. Queue 3(8), 18–24 (2005)
7. Murphy, K.: A brief introduction to graphical models and bayesian networks. Available online at: http://www.cs.ubc.ca/ murphyk/Bayes/bnintro.html (1998)
8. Chickering, D.M.: Learning bayesian networks is NP-complete. Learning from Data: Artificial Intelligence and Statistics V (1996)
9. Gama, J.: Iterative bayes. Theoretical Computer Science 292(2), 417–430 (2003)
10. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: IJCAI, pp. 1300–1309 (1999)
11. Neville, J., Jensen, D.: Iterative classification in relational data. In: AAAI Workshop on Learning Statistical Models from Relational Data, pp. 13–20 (2000)
12. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: ACM SIGMOD, Washington, D.C, pp. 207–216 (1993)
13. Alpaydin, E.: 3. In: Introduction to Machine Learning, pp. 39–59. MIT Press, Cambridge (2004)