2009 IEEE International Conference on Data Mining Workshops

Semantic-rich Markov Models for Web Prefetching

Nizar R. Mabroukeh and C. I. Ezeife
School of Computer Science
University of Windsor
Windsor, Canada
mabrouk@uwindsor.ca

***Abstract*—Domain knowledge for web applications is currently being made available as domain ontology with the advent of the semantic web, in which semantics govern relationships among objects of interest (e.g., commercial items to be purchased in an e-Commerce web site).**

Our earlier work proposed to integrate semantic information into all phases of the web usage mining process, for an intelligent semantics-aware web usage mining framework. There are ways to integrate semantic information into Markov models used in the third phase for next page request prediction. Semantic information is combined with the transition probability matrix of a Markov model. This way, it provides a low order Markov model with intelligent accurate predictions and less complexity than higher order models, also solving the problem of *contradicting prediction*. This paper proposes to use semantic information to prune states in Selective Markov models SMM, semantic information can lead to context-aware higher order Markov models with about 16% less space complexity.

***Keywords*-Markov Models; Domain Ontology; Semantic Distance; Next Page Request Prediction; Web Prefetching.**

I. Introduction and Motivation

Predicting user's next page request on the World Wide Web is a problem that affects web server's cache performance and latency. Different methods exist that can look at the user's sequence of page views, and predict what next page the user is likely to view so it can be prefetched. One way is to use association rules as a result of sequential pattern mining [6]. Another way is to model the user's accessed web pages as a Markov process with states representing the accessed web pages and edges representing transition probabilities between states computed from the given user sequence in the web log. In this case, a trained Markov model can be used to predict the single next state, given a set of *k* previous states.

Recently, more businesses on the internet are starting to include domain ontologies in their online applications (e.g. Amazon.com[1], eBay[2]). Domain ontology provides a useful source of semantic information that can be used in next page prediction systems. The availability of this information and the tradeoff problem between state space complexity and accuracy in Markov models [8], trigger a need to integrate semantic information in the mining process.

The integration of semantic infromation directly in the transition probability matrix of lower order Markov models, was prosented as a solution to this tradeoff problem [5], resulting in semantic-rich lower order Markov models. This integration also solves the problem of *contradicting prediction*.

In this paper[3], we propose to use semantic information as a criteria for pruning states in higher order (where $k > 2$) Selective Markov models [4], and compare the accuracy and model size of this idea with semantic-rich markov models and with traditional Markov models used in the literature.

First we discuss Markov models in section II. Section III surveys related work, while Section IV previews domain knowledge and how it is prepared for use in Markov models. Section V discusses the integration of semantic information into low-order Markov models and the proposal to use it for pruning states in Selective Markov models (SMM). Experimental analysis and performance comparison are provided in Section VI, and finally conclusions and future work are presented in Section VII.

II. Problem Background

Given a sequence of web page views generated by a user browsing the world wide web. This sequence can be modeled as a set of pages, or a web session $\mathcal{W} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_l\}$, where $\mathcal{P}_i$ is a random variable representing the *i*th page view in $\mathcal{W}$. The actual web page in a user's web session will be represented by p_i. The problem of next page request prediction is to predict the web page that will be accessed next, i.e. $\mathcal{P}_{l+1}$.

To model the transition between different web pages in a Markov process, the probability that a user will access a certain web page next is based on the current state that he is visiting, resulting in a 1^{st}-order Markov model, or the previous k states in the sequence, resulting in a k^{th}-order Markov model. The probability of moving to state S_2 in a Markov model given the current state S_1 is the conditional probability $P(S_2|S_1)$. For example, the probability of the sequence $<be>$ happening, if $\mathcal{P}_i =< b >$ and $\mathcal{P}_{i+1} =< e >$,

[1]http://www.wsmo.org/TR/d3/d3.4/v0.2/#ontology
[2]www.ebay.com

[3]This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an operating grant (OGP-0194134) and a University of Windsor grant.

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.18

is the conditional probability of accessing $\mathcal{P}_{i+1}$ after $\mathcal{P}_i$, estimated as follows:

$$P(\mathcal{P}_{i+1}|\mathcal{P}_i) = \frac{frequency(< p_i p_{i+1} >)}{frequency(< p_i >)} \quad (1)$$

Let S_j^k be a state containing *k* page views from $\mathcal{W}$, and l be the number of pages the user visited so far, $S_j^k = < p_{l-(k-1)}, p_{l-(k-2)}, \ldots, p_l >$. The probability of accessing a page p_i after the set of *k* pages S_j^k is estimated in a kth-order Markov model from a history web log (training data) as follows:

$$P\left(p_i \mid S_j^k\right) = \frac{frequency(< S_j^k p_i >)}{frequency(< S_j^k >)} \quad (2)$$

Using $\mathcal{W}$, the page p_{l+1} that the user will most probably access next is given by

$$p_{l+1} = \arg\max_{p \in \mathbb{P}} \{P(\mathcal{P}_{l+1} = p|\mathcal{P}_l, \mathcal{P}_{l-1}, \ldots, \mathcal{P}_{l-(k-1)}\} \quad (3)$$

where $\mathbb{P}$ is the set of all pages in the web site. The **argmax** operator returns the page with the highest probability. The *contradicting prediction* problem occurs when **argmax** returns more than one result with equal probabilities.

With S_j^k representing the states of the Markov model, a markov process is modeled as a directed acyclic graph in which every vertex represents a state corresponding to a page view in the sequence, and edges labeled with probabilities representing transitions between the connected states according to (1), or (2) in the case of *k*th-order Markov model. All transition probabilities are stored in a transition probability matrix $\mathbf{P}_{n \times n}$, where *n* is the number of states in the model.

III. Related Work

In applying probabilistic models for web prefetching, the focus in the last decade has been on Markov models [2][3][4]. Bestravos in [2] used a method that first estimates the conditional probabilities of transitioning directly from each web page to every other web page within a time T_w based on server log file analysis. This is a 1^{st}-order Markov model for predicting surfer paths. Bestravos did not, however, explore the effects of using longer surfer paths (higher-order Markov models) in the predictive model. Using an *n-gram* representation of user access paths, Pirolli and Pitkow [7] study the prediction power and effects of using higher order Markov models, based on a training set collected one day from xerox.com website, and tested against data collected the next day. Given a penultimate path match between paths in the training and test data, the model examined all the conditional probabilities $p(x_n|x_{n-1}, \ldots, x_{n-k})$ available for all pages x_n, and predicted that the page having the highest conditional probability of occurring next, would in fact be requested next. This research, in addition to Borges and Levene in [3], lead to the use of higher order Markov models for link prediction. The order of a Markov model corresponds to the number of prior events used in predicting a future event. So, a k^{th}-order Markov model predicts the probability of next event by looking at the past *k* events.

Table I
WEB ACCESS SEQUENCE DATABASE.

Transaction ID	Sequence
T_1	$p_2 p_3 p_2 p_1 p_5$
T_2	$p_2 p_1 p_3 p_2 p_1 p_5$
T_3	$p_1 p_2 p_5$
T_4	$p_1 p_2 p_5 p_2 p_4$
T_5	$p_1 p_2 p_1 p_4$

Using Markov models for prediction suffers from a number of drawbacks. As the order of the Markov model increases, so does the number of states and the model complexity. On the other hand, reducing the number of states leads to inaccurate transition probability matrix and lower coverage, thus less predictive power, and less accuracy. To counter the reduction in coverage, and as a solution to this tradeoff problem, various Markov models of differing order can be trained and used to make predictions. The resulting model is referred to as the All-*K*th-Order Markov model [8], such that if the kth-order Markov model cannot make the prediction then the (k-1)th-order Markov model is tried, and so on. The problem with using the All-*K*th-Order Markov model is the large number of states contributing to the complexity of the model and the latency of prediction, making it inappropriate for online prediction. On the other hand, selective Markov models (SMM) [4] that only store some of the states within the model have also been proposed as a solution to the mentioned tradeoff problem. They start off with an All-*K*th-Order Markov model, then a post pruning approach is used to prune out states that are not expected to be accurate predictors. The result is a model that has the prediction power of All-*K*th-Order models with less space complexity and more prediction accuracy. Deshpande and Karypis in [4] provide three different criteria which might be used separately to prune states in the model before prediction, that is, frequency, confidence, and error. But they did not study the effect and the relation of domain kowledge and semantics on selective Markov models, neither did they try to combine the three pruning criteria into one pruning measure.

IV. Domain Knowledge and Semantic Information

It is assumed here that different user browsing sessions are provided in a clean web log, similar to Table I, and that domain knowledge is made available in the form of domain ontology provided by the ontology engineer during the design of the web site. A *core ontology* with *axioms* is defined by Stumme et al. [9] as a structure $\mathcal{O} := (\mathcal{C}, \leq_{\mathcal{C}}, \mathcal{R}, \sigma, \leq_{\mathcal{R}}, \mathcal{A})$ consisting of:

Table II
DOMAIN KNOWLEDGE CONTAINED IN EACH ACCESSED PAGE.

Accessed Page	Actual corresponding web page	Ontology mapping
p_1	/cameras.html	*Cameras*
p_2	/cameras/canon.html	*Still Camera*
p_3	/chem/fsoln.html	*Filmdeveloping solution*
p_4	/film/videofilm.html	*Video Film*
p_5	/elect/dbatteries.html	*Dry Battery*

$$M = \begin{bmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ p_1 & 0 & 1 & 5 & 1 & 3 \\ p_2 & 1 & 0 & 2 & 2 & 3 \\ p_3 & 5 & 2 & 0 & 4 & 8 \\ p_4 & 1 & 2 & 4 & 0 & 8 \\ p_5 & 3 & 3 & 8 & 8 & 0 \end{bmatrix}$$

Figure 1. Semantic distance matrix.

- two disjoint sets $\mathcal{C}$ and $\mathcal{R}$ whose elements are called *concept identifiers* and *relation identifiers*, respectively,
- a partial order $\leq_{\mathcal{C}}$ on $\mathcal{C}$, called concept hierarchy or taxonomy,
- a function $\sigma : \mathcal{R} \rightarrow \mathcal{C}^+$ called *signature* (where $\mathcal{C}^+$ is the set of all finite tuples of elements in $\mathcal{C}$),
- a partial order $\leq_{\mathcal{R}}$ on $\mathcal{R}$, called *relation hierarchy*, and
- a set $\mathcal{A}$ of logical axioms in some logical language $\mathcal{L}$.

For example, objects representing products in an e-Commerce application (call it *eMart*) are instances of concepts (also called *classes*) represented formally in the underlying domain ontology using a standard ontology framework, and an ontology representation language like OWL[4]. For example, a "Canon PowerShot A2000 IS" is a brand of a digital still camera sold on *eMart*, that is an instance of the *Digital* class which is a subclass of *Still Camera* class. The subclass relationship is represented by the concept hierarchy. The ontology behind *eMart* is realized as the semantic web adopted in the e-Commerce application, such that each web page is annotated with semantic information, during the development of the website, thus showing what ontology class it is an instance of. For example, page p_2 from Table I, contains the "Canon PowerShot A2000 IS" product, which makes it an instance of the subclass of *Digital Still Camera* in the ontology. Table II shows a mapping between the web pages from the sequence database of Table I being mined and the ontology $\mathcal{O}$ (not shown here due to space limitation), as a result of the preprocessing described.

During mapping of web pages to their corresponding classes, *semantic distance* can be computed and stored in a look up matrix, called the *semantic distance matrix M* [5], as in Figure 1.

We define the *Semantic Distance* M_{p_i,p_j} as a measure of the distance in the ontology $\mathcal{O}$ between the two classes of which p_i and p_j are instances. In other words, it is the measure in units of semantic relatedness between any two web pages p_i and p_j, assuming that a single web page represents only one concept from the ontology. Semantic distance is achieved by computing the topological distance, in separating edges (is-a relations), between the two classes in the ontology, by counting the number of is-a edges required to get from the class which represents p_i to the class which represents p_j before the mining process. The more related two pages are, the lower is their semantic distance.

A *Semantic Distance Matrix M* is an $n \times n$ matrix of all the semantic distances among all the *n* web pages in the sequence database.

$$M = \begin{bmatrix} \mathrm{M}_{p_1,p_1} & \cdots & \mathrm{M}_{p_1,p_n} \\ \vdots & \ddots & \vdots \\ \mathrm{M}_{p_n,p_1} & \cdots & \mathrm{M}_{p_n,p_n} \end{bmatrix}$$

, where M_{p_i,p_j} is the number of edges separating p_i from p_j.

Another related term used here is the *Maximum Semantic Distance* η, which is a value that represents the maximum allowed semantic distance between any two web pages. Maximum semantic distance is inversely proportional to the maximum level of relatedness a user would allow between two concepts. It can be user-specified (i.e., a user with enough knowledge of the used ontology can specify this value) or it can be automatically calculated from the minimum support value specified for the mining algorithm, by applying it as a restriction on the number of is-a edges in the ontology graph. For example, if the minimum support used in the mining algorithm is 5% and the number of edges in the ontology is 60 edges, then η = 3, meaning that the maximum semantic distance allowed between any two classes in the ontology is only 3 edges away, $\eta = min\_sup \times |\mathcal{R}|$.

V. THE PROPOSED SEMANTIC-RICH MARKOV MODELS

While semantic information can be used in Markov models to provide semantically accurate and informed predictions, there are mainly two goals. First, to come out with low order Markov models that have a comparative predictive power to higher order models, while at the same time using less complex state space. Secondly, to solve the contradicting prediction problem mentioned in Section II. Two ways are discussed here for using semantic information in Markov models. The first way in Section V-A directly integrates semantic distances into the probability transition matrix of low order Markov models. The second method introduced in Section V-B, and proposed in this paper, uses the semantic distance as a measure to prune the states in a Selective Markov model.

A. Semantics Integration and Contradicting Prediction

The semantic distance matrix *M* is directly combined with the transition matrix $\mathbf{P}$ of a Markov model of the given sequence database, into a weight matrix *W*. This weight

[4]http://www.w3.org/TR/owl-features/

$$P = \begin{bmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ p_1 & 0 & 0.43 & 0.14 & 0.14 & 0.28 \\ p_2 & 0.5 & 0 & 0.125 & 0.125 & 0.25 \\ p_3 & 0 & 1 & 0 & 0 & 0 \\ p_4 & 0 & 0 & 0 & 0 & 0 \\ p_5 & 0 & 0.25 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2. Transition probability matrix for Markov model from Table I.

$$W = \begin{bmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ p_1 & 0 & 1.33 & 0.64 & 1.04 & 0.98 \\ p_2 & 1.37 & 0 & 0.87 & 0.87 & 0.87 \\ p_3 & 0.74 & 1.89 & 0 & 0.79 & 0.58 \\ p_4 & 0.93 & 0.87 & 0.73 & 0 & 0.47 \\ p_5 & 0.86 & 1.11 & 0.64 & 0.64 & 0 \end{bmatrix}$$

Figure 3. Weight matrix *W* resulting from combining *M* (Figure 1) with Markov transition matrix $\mathbf{P}$ according to eq. (4).

matrix is consulted by the predictor software, instead of $\mathbf{P}$, to determine future page view transitions for caching or prefetching.

The *Weight Matrix W* can be defined as an $n \times n$ matrix resulting from combining the semantic distance matrix *M* with the Markov transition probability matrix $\mathbf{P}$, as follows,

$$W_{p_i,p_j} = \mathbf{P}_{S_i,S_j} + \begin{cases} 1 - \dfrac{M_{p_i,p_j}}{\sum_{k=1}^{n} M_{p_i,p_k}} & , \; M_{p_i,p_j} > 0 \\ 0 & , \; M_{p_i,p_j} = 0 \end{cases} \quad (4)$$

In details, and in order to combine $\mathbf{P}$ and *M*, *M* has to be normalized such that each entry is between 0 and 1, so it can fit in any Markov-based prediction tool in place of $\mathbf{P}$. This is achieved by dividing each row entry by the row sum $\sum_{k=1}^{n} M_{p_i,p_k}$. The next step would be to add both matrices together, in order to enrich $\mathbf{P}$ with semantic distance measures. But, the values in *M* represent a distance, such that the higher the value, the more is the distance. This value is inversely proportional to the required output weight (i.e., a greater distance should result in a smaller weight). To solve this problem, each non-zero entry in the normalized *M* is subtracted from 1.

For prediction, assume that in the test set the user went through this sequence of page views $<p_2 p_5 p_1 p_3>$. Looking at $\mathbf{P}$ in Figure 2, there is a 100% probability that the user will next view page p_2. A problem that could arise here is *contradicting prediction*, for example, assume in Figure 2 that $P(p_2|p_1) = P(p_5|p_1) = 0$, and notice that $P(p_3|p_1) = P(p_4|p_1)$, which means that there is an equal probability a user will view page p_3 or p_4 after viewing page p_1. Thus, the prediction capability of the system will not be accurate in terms of which is more relevant to predict after p_1, and the prediction will be ambiguous. Integration of the semantic distance matrix can solve this problem. The transition probability matrix can be combined with the given semantic distance matrix of Figure 1, resulting in *W*, as in Figure 3, according to equation (4).

For deeper discussion and experimental analysis of this integration of semantics with transition probabilities, we refer the reader to our previous work in [5].

B. Using Semantic Distance for State Pruning

This paper proposes to use the maximum semantic distance measure η for state pruning in Selective Markov models (SMM). In this case, an All-*K*th-Order Markov model is built first as in [8], next states that do not contribute to the model, i.e. which have zero frequency, are pruned. Then, any state S_j^k, having $M_{p_{l-(k-1)},p_{l-(k-2)}} > \eta$, where l is the number of pages the user visited so far and *j* is a simple enumeration of the states in the model, such state will be pruned from the model. Next we show a detailed example and then discuss performance in Section VI. We limit our models to 3^{rd}-Order Markov models, similar to Deshpande and Karypis [4].

Example: Figure 4 shows the All-*K*th-Order Markov model for the web log in Figure 4(a). This model consists of 1^{st}, 2^{nd} and 3^{rd} -order models. To create the selective Markov model, states with a right arrow $\rightarrow$ are pruned as they do not contribute to the model (i.e. they have no next state), and states with a double right arrow $\Rightarrow$ are pruned since the semantic distance between the pages is higher than the maximum allowed semantic distance (assumimg $\eta = 2$). For example, state $S_{18}^2 = <p_5, p_2>$ is pruned since $M_{p_5,p_2} = 3$ is greater than η based on the semantic distance matrix in Figure 1. In this example, 14 states are pruned from just the 2^{nd}-order part of the selective model, 5 of which are pruned based on semantic distance, in total resulting in 70% reduction in just the 2^{nd}-order state space of the model over the All-*K*th-order model.

VI. Experimental Analysis

Experiments were carried out on three kinds of data sets. Two data sets, *DS-1* and *DS-2* are generated using the IBM resource data generator [1]. *DS-1* is a small data set resembling a web log of 5000 user sessions, while *DS-2* is a large data set resembling a web log of 80,000 user sessions. A third data set *DS-3* is a staged data set manually constructed to resemble *eMart*'s web log with 200,000 user sessions. Characteristics of the three data sets are described in Figure 5. The assumptions made in Sections II and IV, also apply on the dataset and the way experiments were conducted.

Data set	# of Transactions	# of Unique pages	Ave. Trans. Length
DS-1	5000	113	2.5
DS-2	80000	200	5
DS-3	200000	155	8

Figure 5. Data sets used for experimental analysis.

Using these data sets, 1^{st}-Order, 2^{nd}-Order, and All*K*th-Order Markov models, along with frequency-pruned SMM

SessionID	Access Sequence
$\mathcal{W}_1$	$p_2p_3p_2p_1p_5$
$\mathcal{W}_2$	$p_2p_1p_3p_2p_1p_5$
$\mathcal{W}_3$	$p_1p_2p_5$
$\mathcal{W}_4$	$p_1p_2p_5p_2p_4$
$\mathcal{W}_5$	$p_1p_2p_1p_4$
$\mathcal{W}_6$	$p_3p_1p_4p_2p_1p_5$
$\mathcal{W}_7$	$p_4p_1p_2p_5p_2p_3p_2p_1$
$\mathcal{W}_8$	$p_1p_4p_2p_3p_1p_2p_5$

(a)

1^{st}-Order State	p_1	p_2	p_3	p_4	p_5
$S_1^1 =< p_1 >$	0	0.38	0.08	0.23	0.23
$S_2^1 =< p_2 >$	0.43	0	0.21	0.07	0.29
$S_3^1 =< p_3 >$	0.40	0.60	0	0	0
$S_4^1 =< p_4 >$	0.20	0.40	0	0	0
$S_5^1 =< p_5 >$	0	0.29	0	0	0

(b)

2^{nd}-Order State	p_1	p_2	p_3	p_4	p_5
$S_1^2 =< p_1, p_2 >$	0.20	0	0	0	0.80
$\Rightarrow S_2^2 =< p_1, p_3 >$	0	1.00	0	0	0
$S_3^2 =< p_1, p_4 >$	0	0.67	0	0	0
$\rightarrow S_4^2 =< p_1, p_5 >$	0	0	0	0	0
$S_5^2 =< p_2, p_1 >$	0	0	0.17	0.17	0.50
$S_6^2 =< p_2, p_3 >$	0.33	0.67	0	0	0
$\rightarrow S_7^2 =< p_2, p_4 >$	0	0	0	0	0
$\Rightarrow S_8^2 =< p_2, p_5 >$	0	0.50	0	0	0
$\Rightarrow S_9^2 =< p_3, p_1 >$	0	0.50	0	0.50	0
$S_{10}^2 =< p_3, p_2 >$	1.00	0	0	0	0

2^{nd}-Order State	p_1	p_2	p_3	p_4	p_5
$\rightarrow S_{11}^2 =< p_3, p_4 >$	0	0	0	0	0
$\rightarrow S_{12}^2 =< p_3, p_5 >$	0	0	0	0	0
$S_{13}^2 =< p_4, p_1 >$	0	1.00	0	0	0
$\Rightarrow S_{14}^2 =< p_4, p_2 >$	0.50	0	0.50	0	0
$\rightarrow S_{15}^2 =< p_4, p_3 >$	0	0	0	0	0
$\rightarrow S_{16}^2 =< p_4, p_5 >$	0	0	0	0	0
$\rightarrow S_{17}^2 =< p_5, p_1 >$	0	0	0	0	0
$\Rightarrow S_{18}^2 =< p_5, p_2 >$	0	0	0.50	0.50	0
$\rightarrow S_{19}^2 =< p_5, p_3 >$	0	0	0	0	0
$\rightarrow S_{20}^2 =< p_5, p_4 >$	0	0	0	0	0

(c)

Figure 4. (a) Sample web log with user transactions. (b) Resulting probability transition matrix for 1^{st}-Order Markov model. (c) Resulting probability transition matrix for 2^{nd}-Order Markov model.

are built for testing and comparison. Semantic information in the form of a semantic distance matrix (SDM) for each data set, is generated randomly. While for *DS-3*, the semantic distance matrix is manualy constructed based on the given *eMart*'s ontology. Using this matrix, semantic-rich 1^{st}-order Markov models and semantic-pruned SMM are also constructed for testing. A frequency threshold [4] of 0 is used in the Selective markov models (SMM), while varying values for η are used in the semantic-pruned SMM.

Testing is done in the following way. First, every data set is divided into a training set, which is the first 75% sessions in the data set, and a test set, the remaining 25%. Then, in the training part, the described Markov models are construced from the data sets and the model size for each one is noted. The model size is defined here as the number of states in each model. The testing part is made similar to the method described in Deshpande and Karypis [4], that is, every model is given a trimmed session from the test set for prediction, in which the last page of the session is hidden. The prediction made by the model is then compared with the hidden page of the session to compute the accuracy of the model. Accuracy represents the predictive power of the model and is measured as the percentage of successful predictions made.

The performance in these experiments is measured by model size and accuracy, the two factors in the tradeoff problem described previously. The goal is to find the best model that can provide accurate predictions while maintaining a comparatively small model size. Sometimes a model might not be able to provide a prediction due to two reasons. First, a contradicting prediction problem might occur. Secondly, the hidden page might not have been present in the training set. If that happens, it will output the web page with the highest frequency as a default prediction, in the case of selective or stand-alone Markov models, or it will depend on semantic distance measures to make an informed prediction that is considered semantically correct, as is the case in semantic-rich models.

Semantic-rich 1^{st}-order Markov models are found to totally eliminate the contradicting prediction problem in all of the data sets used. For example, testing *DS-3* using the 1^{st}-order Markov model resulted in a number of 258 contradicting predictions, while running the same test using the semantic-rich 1^{st}-order Markov model, resulted in 0 contradicting predictions.

Figure 6, shows results of the experiments. One can notice that semantic-rich 1^{st}-order Markov models have the same model size as regular 1^{st}-order models, this is because no pruning is used in semantic-rich models. While these semantic-rich models solve the problem of contradicting prediction, they also provide very close accuracy to that of regular 1^{st}-order models. This accuracy differs depending on the nature of the sessions in the data sets. For example, in *DS-3*, the accuracy of semantic-rich model (that is 29.80%) is about equal to that of non semantic-rich (which is 30.02%), because, once the web log was examined, it was found that users traversed pages that are highly semanticaly related and would resemble the structural relations between the pages in the web site.

Highlighted in Boldface in Figure 6, are the highest prediction accuracies for each data set, which show that frequency-pruned SMM is mostly the best choice for high

Model	*DS-1*		*DS-2*		*DS-3*	
	Accuracy in %	Size	Accuracy in %	Size	Accuracy in %	Size
1st-order	17.50	870	17.50	6320	30.02	992
Sem. 1st-order	12.50	870	19.02	6320	29.80	992
2nd-order	18.00	25230	18.70	181858	30.12	30752
All$K$th-order	**26.34**	757770	19.80	1526176	29.52	985056
FPSMM	25.81	21547	**25.03**	807617	**31.83**	12741

Figure 6. Comparing accuracy and model size of different Markov models for the different data sets. Sem. 1st-order stands for semantic-rich 1^{st}-Order Markov models, and FPSMM stands for frequency-pruned selective markov models with 0 frequency threshold.

Max Semantic Distance η	*DS-1*		*DS-2*		*DS-3*	
	Accuracy in %	Size	Accuracy in %	Size	Accuracy in %	Size
$\eta = 5$	5	2697	0	31163	11.71	5032
$\eta = 20$	10.71	7283	12.9	89039	20.00	9118
$\eta = 50$	20.01	10066	19.35	254399	27.43	10230
$\eta = 70$	**22.77**	**15844**	**23.17**	**726855**	**31.00**	**11420**
$\eta = 90$	24.95	20364	24.81	755641	31.34	12420
$\eta = 110$	25.81	21547	25.00	774275	31.80	12741

Figure 7. The accuracy and model size of semantic- and frequency-pruned selective markov models using different data sets.

accuracy and small model size, as the results show an average decrease of 57% in model size, with only an average of 2.7 deviation in accuracy. The second best is the 2^{nd}-order markov model, which confirms the findings in [4]. But, could there be a better compromise in which the accuracy is higher and the model size is smaller? In an attempt to answer this, the proposed semantic-pruned SMMs are built with differing values for η, as in Figure 7.

It can be noted from Figure 7 that, as the value of η decreases, so does the model size. Which is expected, since less η means more pruning will take place, and accuray also decreases due to the same reason. At a value of $\eta = 70$, the semantic-pruned SMM does provide an accuracy close to that of its respective frequency-pruned SMM (FPSMM) in Figure 6, with an average difference of only 1.91 in accuracy, and at the same time maintain a smaller state space, with an anverage of 16% decrease in model size

VII. Conclusions and Future Work

The integration of semantic information, drawn from an underlying domain ontology, into probabilitic low-order Markov models is discussed. Semantic information is infused into the Markov transition probability matrix to convert it to a matrix of weights for better-informed prediction, and to overcome the problem of contradicting prediction. This paper takes this idea a step further by proposing to use maximum semantic distance as a measure for pruning higher-order Markov models.

The performance of semantic-rich 1^{st} and 2^{nd}-order Markov models is studied and compared with that of higher-order Selective Markov models and semantic-pruned Selective Markov models. It was found that semantic-pruned SMM have a 16% smaller size than frequency-pruned SMM and provide nearly an equal accuracy. Experiments also show that semantic-rich low-order Markov models can overcome the problem of contradicting prediction

Future work includes the development of a method that can gather better semantic information to be used than simple semantic distance, and investigating the benefits of pushing the ontology towards more semantic information that can aid inferencing, along with association rules in the recommendation/prediction phase of web usage mining.

References

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

[2] A. Bestravos. Using speculation to reduce server load and service time on the www. In *Proceedings of the 4th ACM Intl. Conf. on Information and Knowledge Management*, 1995.

[3] J. Borges and M. Levene. Data mining of user navigation patterns. In *Web Usage Analysis and User Profiling, LNAI 1836*. Springer, 2000.

[4] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *Transactions on Internet Technology*, 4(2):163–184, 2004.

[5] N. R. Mabroukeh and C. I. Ezeife. Using domain ontology for semantic web usage mining and next page prediction. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, November, 2-6 2009. To appear.

[6] N. R. Mabroukeh and C. I. Ezeife. A taxonomy of sequential and web pattern mining algorithms. *ACM Computing Surveys*, 2010. To appear.

[7] P. Pirolli and J. E. Pitkow. Distributions of surfers' paths through the world wide web: Empirical characterization. *World Wide Web*, 1:1–17, 1999.

[8] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict www surfing. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems 2*, pages 13–21, October 1999.

[9] G. Stumme, A. Hotho, and B. Berendt. Semantic web mining: State of the art and future directions. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 4(2):124–143, 2006.