U-REST: An Unsupervised Record Extraction SysTem

Yuan Kui Shen
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge MA, 02139 USA
yks@csail.mit.edu

David R. Karger
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge MA, 02139 USA
karger@csail.mit.edu

ABSTRACT

In this paper, we describe a system that can extract *record* structures from web pages with no direct human supervision. Records are commonly occurring HTML-embedded data tuples that describe people, offered courses, products, company profiles, *etc.* We present a simplified framework for studying the problem of unsupervised record extraction – one which separates the algorithms from the feature engineering. Our system, U-REST formalizes an approach to the problem of *unsupervised record extraction* using a simple two-stage machine learning framework. The first stage involves clustering, where structurally similar regions are discovered, and the second stage involves classification, where discovered groupings (clusters of regions) are ranked by their likelihood of being records. In our work, we describe, and summarize the results of an extensive survey of features for both stages. We conclude by comparing U-REST to related systems. The results of our empirical evaluation show encouraging improvements in extraction accuracy.

Categories and Subject Descriptors

H.3.m [**Information Systems**]: Miscellaneous

General Terms

algorithms, experimentation

Keywords

record extraction, clustering

1. INTRODUCTION

Records are data tuples wrapped in HTML. On the web, records are presented as collections of consistently formatted HTML snippets. They manifest in *list pages* such as search results, course catalogues, or directory listings. While there are many types of records (nested records, tree records, *etc.*), our focus is on flat records: records characterized as having an ordered set of *fields* corresponding to the columns of an underlying data source (such as a database table). Given a list page P, the task of record extraction is to return the set of regions $C = r_1 \ldots r_n$ that best matches a human labelled reference set, L. When this task is accomplished without any human supervision, we call it unsupervised record extraction (URE). The supervised variant of this problem has been studied extensively. For instance, most recently, Hogue et al [2] demonstrated the use of a tree model to represent the extraction pattern. In general, supervised methods use learning techniques to induce an extraction pattern from a set of labelled examples. Because records repeat it is intuitive to think that simply searching for repetitions should aid in finding record instances. However not all repetitions are records. Some repetitions are composed of parts of records (*fields*), other constitute *collection-of-records*, and yet others come from formatting regularities that serve functions such as *navigation* or advertising. Hence, the URE task is difficult because there are different types of repetitions, and repetitions are often noisy. Several recent works have tried to tackle the URE problem. Omini [1] searched for record separator tags between contiguous record instances, posing the problem as one of segmentation. MDR/DEPTA [5] compared successive groups of subtrees and returned similar groupings as record regions. ViNT [6] utilized visual hints, *content lines*, as record identification features. However, little attention has been devoted to how features independently affect the system accuracy. Our work uses known machine learning techniques and survey some simple features to gain a clearer insight into the importance of features in the URE task.

2. SYSTEM OVERVIEW

The input to U-REST is a record-containing list page, the output is a set of potential record instances (record sets). List pages are first converted into a tag tree. A tag tree (or DOM) has a node for each open and close HTML tag pair. Each subtree of the page tag tree represents a distinct continuous (visible) region (or a block) on the web page. Any single subtree or set of adjacent subtrees (*sibling subtrees*) can represent a potential record instance. Our task is to return the sets of subtrees (or sibling subtrees) that best correspond to records.

During U-REST's pattern discovery phase, the page DOM is decomposed into constituent subtrees. Those subtrees that are clearly non-records: root of the page or non-content-bearing leaf nodes (e.g. `<BR>` tags) are removed. The resulting subtrees undergo clustering, using HAC (hierarchical agglomerative clustering). The goal of clustering is to find salient structural repetitions. HAC iteratively merges the closest two points (trees or clusters of trees). The HAC clustering metric determines the threshold at which the merging process terminates. We designed this metric as a tree-pair-

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

wise similarity function: $\phi(T_i, T_j) : \langle \phi_1(T_i, T_j), \ldots, \phi_n(T_i, T_j)\rangle \rightarrow \{0, 1\}$, ϕ is 1 if the pair is similar enough to be in the same cluster (0 otherwise). ϕ is a trained classifier defined by its vector of features over tree pairs. In our experiments, a linear kernel SVM was found to be the most effective. We also surveyed an extensive set of features [4] (some top performing ones are noted in table 1). Using feature selection, we found a three-optimal combination: 1) *Tree edit-distance* - using the standard polynomial time tree-edit distance algorithm compute the optimal alignment between two (ordered labelled) trees; the edit distance score is the count of the matching nodes in the optimal alignment. 2) *Tree context* - also an edit distance metric but applied on the tag path, the tag label sequence from the subtree to the root of the page. 3) *Tri-gram model* - models the tree as a vector of label triplets: the label of the root, the i^{th} child, and $(i+1)^{th}$ child. Our results show that preserving the structural order of trees is critical when comparing trees: tree-edit distance predominates all n-gram based metrics. After clustering, the discovered clusters contain non-record as well as record repetitions; non-record types include: fields, collections of records, decorative blocks, or navigational content. To differentiate record from these non-record clusters, we designed a record cluster classifier, modeled as a function over clusters: $\psi(C_i) : \langle \psi_1(C_i), \ldots, \psi_n(C_i)\rangle \rightarrow [0, 1]$. ψ assigns the most record-like cluster the highest score. Our feature survey and feature selection work [4] yielded three classes of features that performed well in this subtask. 1) *Contiguity* is the amount of content interleaved between record instances; intuitively, record instances that are close to each other should have very little content in between instances. This feature aids in differentiating fields from records. 2) *Content coverage* is the page-relative ratio of content that (the members of) a cluster occupy; the higher content coverage, the more importance that cluster adds to the page. This feature differentiates navigational content from records. 3) *Variation* is the measure of the mean formatting diversity under each subtree. Record clusters have high variation because each instance contains fields, and adjacent fields are usually formatted differently. Visually, individual record instances appear *heterogeneously* formatted, but collections of records appear *homogenously* formatted (For more details see [4]). An SVM trained on features from these three classes produces an optimal ψ; the system returns the highest ψ scoring cluster as the record cluster.

Feature	recall	precision	f-score
Tree-Edit-Distance	0.98458	0.92731	0.94985
Tree Context	0.90909	0.66714	0.73903
Parent-Child-Child (3-gram)	0.98582	0.39688	0.50056
2-gram	0.90926	0.34137	0.41022
1-gram	0.90939	0.26980	0.33532

Table 1: Summary of top ϕ_i features. Reported here are test recall/precision values for classifiers trained and tested on 10000+ tree pairs extracted from ten list pages.

3. EVALUATION AND RESULTS

We compared the record sets extracted by U-REST, Omini and MDR [3] with a reference human-labelled record set. The evaluation metric (reported in Table 2) is the number of pages correctly labelled at or above a *fixed* f-score[1]. Our evaluation metric differs from those reported by Omini or MDR because the metric is not an average of f-scores over record instances but the number of pages for which the systems achieved a fixed extraction quality. This evaluation method is more useful as it guarantees a base level of accuracy for the system. Manual analysis of Omini/MDR results shows that the mis-identification of record collections as records is one of key causes of system errors. The variation feature introduced in our work helped differentiate much of the record-collection cases.

System	#correct (/62)	%
U-REST	29	46.8
Omini	22	35.5
MDR	14/54	25.9
Best Possible	37	59.7
Baseline	9	14.5

Table 2: System performance comparison with cluster quality at f-score ≥ 0.75. Baseline ranks clusters based on their content coverage; Best-possible presupposes that an oracle has selected the best cluster after clustering (the upper limit imposed by clustering quality).

4. CONCLUSION

We demonstrated a simple two stage model URE system: U-REST. In the pattern detection subtask, we found that order-preserving features such as tree edit distance outperformed approximations such as n-gram models when comparing pairs of subtrees (for clustering). In the record cluster detection subtask, we found that a combination of variation, contiguity, and content coverage features produced the most accurate results. Our results show that features play a strong role in the final performance of an URE system; by simply optimizing features and feature selection, we can gain accuracy while maintaining system simplicity[2].

5. REFERENCES

[1] D. Buttler, L. Liu, and C. Pu. A fully automated extraction system for the world wide web. In *IEEE ICDCS-21*, April 2001.

[2] A. Hogue and D. Karger. Thresher: Automating the unwrapping of semantic content from the world wide web. In *WWW 2005 Conference*, 2005.

[3] B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. UIC Technical Report, 2003.

[4] Y. K. Shen. Automatic record extraction from the world wide web. Master's thesis, MIT, 2005.

[5] Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 76–85, New York, NY, USA, 2005. ACM Press.

[6] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully automatic wrapper generation for search engines, 2005.

[1] f-score $= \frac{2pr}{p+r}$, where $p = \frac{|C \cap L|}{|C|}$ and $r = \frac{|C \cap L|}{|L|}$

[2] Thanks to Alex Gruenstein, Ali H. Mohammad, Boris Katz for their helpful comments on this paper.