In Praise of Laziness: A Lazy Strategy for Web Information Extraction

Rifat Ozcan[1], Ismail Sengor Altingovde[2], and Özgür Ulusoy[1]

[1] Computer Engineering Department, Bilkent University, Ankara, Turkey
{rozcan,oulusoy}@cs.bilkent.edu.tr
[2] L3S Research Center, Hannover, Germany
altingovde@L3S.de

Abstract. A large number of Web information extraction algorithms are based on machine learning techniques. For such extraction algorithms, we propose employing a lazy learning strategy to build a specialized model for each test instance to improve the extraction accuracy and avoid the disadvantages of constructing a single general model.

1 Introduction

Information extraction (IE) is a well-defined task that essentially aims to extract structured data either from semi-structured data, like HTML and XML pages, or unstructured data; most basically, free text. Web IE algorithms are usually based on machine learning (ML) and pattern recognition techniques to exploit the regularities in the data. Previous works essentially focus on improving the accuracy of the extracted information, whereas the applicability of the extraction approaches is less of a concern. Ironically, algorithms with higher accuracy usually need to exploit a larger set of evidences from the training datasets, which implies higher computational requirements for model creation, and, in turn, potentially less applicability in practice. Furthermore, for practical applications, such demanding requirements for model creation process can delay taking more recent data into account, and confuse and/or frustrate the end users who might have provided some feedback.

In this study, we essentially focus on the Web IE algorithms that are based on ML techniques and advocate that the lazy learning strategy usually employed for classifiers can also be applied for Web IE, to remedy the aforementioned disadvantages of model inference. More specifically, instead of inferring a general model using all available training data, we propose a lazy strategy that learns a model using only the most relevant training instances to a given input document. Such a strategy is especially useful for the scenarios where: (i) the IE task involves extracting a wide range of fields that do not necessarily appear in each training document, (ii) the training document set is highly heterogeneous (i.e., Web pages on the same general topic but collected from various different resources), and (iii) the size of the training collection consistently increases in time (e.g., due to some sort of explicit/implicit annotations from the users). For the first two

R. Baeza-Yates et al. (Eds.): ECIR 2012, LNCS 7224, pp. 565–568, 2012.

scenarios, constructing a single model from the entire training set, so-called *eager* strategy, may not be feasible. This is because significant computational resources would be investigated to construct a very general model that covers the cases that might never or very rarely be encountered during actual extraction. For the third scenario, each time new training data is available, the model needs to be re-generated. During this period, which might take a long time for complex algorithms and large datasets, the new input will be unavailable to guide the extraction process. In such cases, applying a lazy strategy as a replacement or at least as a complementary approach (say, until the model is re-trained) to the eager strategy can be an attractive option.

Our work investigates the potential of using lazy learning strategy for Web IE algorithms based on machine learning techniques. We demonstrate the virtue of lazy strategy using a supervised rule-based IE approach, namely Sequence Rules with Validation (SRV) algorithm [3]. Our experiments reveal that the lazy strategy, even with a small percentage of training data, can yield comparable or sometimes better accuracy than the eager strategy.

In the next section, we first briefly review the related work and then describe the SRV algorithm as it is adapted in this work. In Section 3, we present and evaluate our lazy strategy using the SRV algorithm. We conclude in Section 4.

2 Related Work and Background

Web Information Extraction. A wide range of approaches are proposed for Web IE problem in the literature. Chang et al. provide an extensive analysis of IE systems based on three dimensions, namely, task domain, automation degree, and applied techniques [2]. They also classify IE systems based on the method of construction, a process which is carried on either manually, by crafting explicit rules into the wrappers, or automatically, using one of the supervised, semi-supervised or unsupervised techniques.

Lazy learning is a classical strategy best exemplified with the nearest neighbor or instance-based learning applied in many different problem domains (e.g., see [1]). Our work presented here is inspired from a recent study where the authors utilize lazy learning for associative document classification [4]. Their approach is based on the intuition that a lazy strategy can exploit more focused and relevant evidences so that both the accuracy and efficiency of the classification can be improved. In the associative classification scenario, the lazy strategy prevents generation of a global and large set of rules that would be rather useless during actual classification; and thus improves accuracy, and even efficiency (using a simple caching mechanism) of the classifier. We envision that the Web IE algorithms based on machine learning can also benefit from the lazy strategy in similar ways, and provide here our first results in this direction. In a very recent study [5], a lazy evaluation of different analyzer modules within an IE pipeline is proposed, with sole purpose of improving efficiency. However, to the best of our knowledge, no previous works address lazy learning for information extraction as we do in this study.

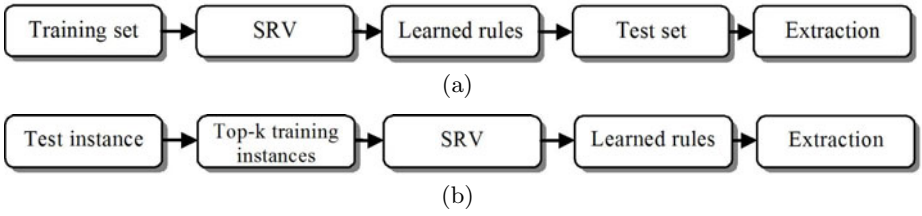

Fig. 1. Web IE with SRV algorithm using (a) eager, and (b) lazy learning strategies

Adaptation of SRV Algorithm. SRV is a machine learning algorithm that learns first order logic extraction pattern rules from training examples [3]. Training set consists of annotated HTML pages. Each learned extraction rule consists of predicates. Rule learning phase starts with an empty rule and it greedily adds predicates to the rule that achieves the most information gain by covering as many positive examples as possible and at the same time as few negative examples as possible.

The original SRV algorithm [3] is a single-slot version that only extracts single fields; i.e., it does not capture relationships between these fields. For instance, in a scenario of extracting instructor information from course homepages, it can extract instructors names and email addresses independently, but can not associate each name with the corresponding e-mail. We slightly modified SRV algorithm to allow multi-slot extraction rules by defining two additional predicate types.

3 A Lazy Strategy for Web Information Extraction

In this paper we demonstrate the virtue of the lazy learning strategy using the SRV algorithm for the task of extracting a number of fields from course homepages. As shown in Fig. 1, a crucial step for the lazy strategy is choosing k nearest neighbors from the training set for a given test instance. Since the instances in our problem domain are Web documents, we compute the cosine similarity between two documents vectors with $tf \times idf$ weighting. Note that, structural clues such as the DOM tree of a document or certain HTML tags surrounding text fragments can also be exploited for determining training instances that can be most useful to extract fields from the test instance. We leave exploring such additional features for nearest neighbor computation as a future work.

In the experiments, we use a dataset consisting of 900 computer science course homepages from WebKB project [6]. Lazy and eager learning strategies are evaluated based on their performance on extracting course code, course name and instructor name fields from these web pages. All pages in the dataset are annotated for these three fields by human judges. We apply a 5-fold cross validation for our experiments so that in each fold a different 180-pages subset of the dataset is reserved as the test set and the rest is used as the training set. Note that, eager approach uses all 720 pages for training while the lazy strategy only selects k nearest neighbors of each test page from the training set.

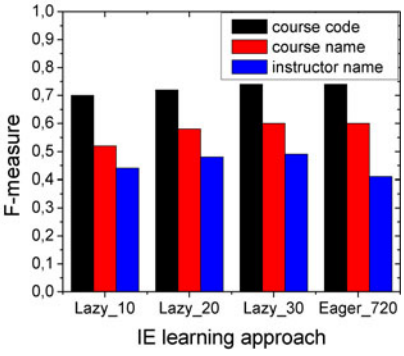

Fig. 2. Effectiveness of eager and lazy strategies for the extracted fields

Fig. 2 presents the effectiveness (in terms of F-measure) of eager and lazy strategies for extracting each one of the three fields, namely, course code, course name, and instructor name. Lazy strategies are experimented by selecting top 10, 20, and 30 training instances per test instance (the eager approach uses all 720 pages for learning). Our findings show that SRV with lazy learning, even when trained with only top-10 instances, can be as effective as the eager approach. The effectiveness of the lazy approach seems to increase with the number of instances selected for training, especially for the instructor name field. In this latter case of extracting instructor name, lazy strategy using 30 training samples can even outperform the eager approach, while it yields almost the same F-measure figures as the eager strategy for extracting the other two fields.

4 Conclusion

Our work provides the first results showing that employing a lazy learning strategy for machine-learning based Web information extraction is a promising approach and it can yield comparable accuracy to the eager strategy.

Acknowledgments. This work is partially supported by EU FP7 Project CUBRIK (contract no. 287704).

References

1. Aha, D.W. (ed.): Lazy learning. Kluwer Academic Publishers, Norwell (1997)
2. Chang, C.-H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. IEEE Trans. Knowl. Data Eng. 18(10), 1411–1428 (2006)
3. Freitag, D.: Information extraction from html: Application of a general machine learning approach. In: Proceedings of AAAI/IAAI, pp. 517–523 (1998)
4. Veloso, A., Meira Jr., W., Zaki, M.J.: Lazy associative classification. In: Proceedings of IEEE International Conference on Data Mining, pp. 645–654 (2006)
5. Wachsmuth, H., Stein, B., Engels, G.: Constructing efficient information extraction pipelines. In: Proceedings of CIKM 2011, pp. 2237–2240 (2011)
6. WebKB: CMU, world wide knowledge base (WebKB) project (2011) `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data`