On Verifying Information Extractors

Daniel Ayala Hernández

1 Problem Statement

Currently, the Web provides many different information sources with valuable information that is available in human friendly formats only. This makes it difficult for software agent to sift through them to extract relevant information to feed automated business processes. Information extractors are software components that help in this task.

Unfortunately, information extractors may easily fail to extract the correct information when the structure of the web documents to which they are applied changes, be it because the web site changes or because a new document with a structure that has not been seen previously is found.

This motivates the need for verifiers, which are components that analyse the information that is returned by an information extractor and raise an alarm if it deviates significantly from the information that is known to be correct.

2 Related Work

Rapture [1] uses a set of numeric features to compute the similarity between an extractor's output and pre-verified outputs. Normal distributions are used as heuristics. Feature values are used to estimate the distribution parameters of each feature. These are used to derive the overall probability that an output is correct. If it is below a user-defined threshold, an alarm is raised.

DataProG [2] computes a means vector from numeric features of pre-verified outputs obtained from a set of queries to a Web source. During the verification phase,

D.A. Hernández(✉)
ETSI Informática, University of Sevilla, Avda. Reina Mercedes, s/n, 41012 Sevilla, Spain
e-mail: dayala1@us.es

Supported by the Spanish R&D&I program under grant TIN2013-40848-R.

F. de la Prieta et al. (eds.), *Trends in Pract. Appl. of Scalable Multi-Agent Syst., the PAAMS Collection*, Advances in Intelligent Systems and Computing 473,
DOI: 10.1007/978-3-319-40159-1_34

new outputs are obtained using the same set of queries and a new means vector is computed. If it is not statistically equivalent to the original one, an alarm is raised.

Maveric [3] improves on Rapture by normalising the distributions used as heuristics and taking negative information into account. (That information is obtained by using so-called perturbations on correct information). The weight of each feature on the final decision is computed using the Winnow algorithm.

We have identified the following problems regarding these proposals:

- Alarms are associated to entire datasets, instead of specific erroneous attributes.
- Rapture and DataProG train with positive examples only. Maveric uses negative examples, but they are synthetic, which means that they might well not be representative enough of actual negative examples. Furthermore, the perturbations have to be handcrafted.
- Rapture and DataProG do not give features different weights representing their relevance, and all are considered equally important. Maveric gives weights to features, but they are global.

3 Hypothesis

We hypothesise that we can improve on the existing verifiers by training binary classifiers that classify information as belonging or not to a certain class.

4 Proposal

Our proposal, Sydney, uses extracted attributes from pre-verified datasets to train binary classifiers. One classifier is created per information class, e.g., book, title or price. If an attribute belongs to the information class associated to a classifier it is used as a positive example, otherwise, it is used as a negative one.

When a dataset must be verified, its attributes are classified using every binary classifier. If the results are not consistent with the supposed class (its classifier gives a negative result or any other classifier gives a positive one), an alarm is raised. This alarm is associated to the inconsistent attribute. The confidence of each classifier is taken into account, so that more precise classifiers are given more importance.

Further verification is performed at dataset level using global features with positive examples only, since negative dataset examples are not available during training.

Unlike existing proposals, ours is trained with real examples of what are and what are not instances of each information class. Existing proposals only do the former or use artificial perturbations. This allows us to use a wider range of existing classification techniques.

5 Evaluation Plan

We intend to evaluate our verifier using results from existing information extractors. These will be verified and classified as well extracted or potentially erroneous. Erroneous extractions shall be used as negative examples.

References

1. Kushmerick, N.: Wrapper verification. World Wide Web **3**(2), 79–94 (2000)
2. Lerman, K., Minton, S., Knoblock, C.A.: Wrapper maintenance: A machine learning approach. J. Artif. Intell. Res. **18**, 149–181 (2003)
3. McCann, R., AlShebli, B., Le, Q., Nguyen, H., Vu, L., Doan, A.: Mapping maintenance for data integration systems. In: VLDB, pp. 1018–1029 (2005)