Relation Extraction for Mining the Semantic Web

José Iria, Fabio Ciravegna

Department of Computer Science, University of Sheffield, UK
{j.iria, f.ciravegna}@dcs.shef.ac.uk

Abstract. The knowledge acquisition bottleneck problem, well-known to the Knowledge Management community, is turning the weaving of the Semantic Web (SW) into a hard and slow process. Nowadays' high costs associated with producing two versions of a document – one version for human consumption and another version for machine consumption – prevent the creation of enough metadata to make the SW realizable. There are several potential solutions to the problem. We advocate the use of automated methods for semantic markup, i.e., for mapping parts of unstructured text into a structured representation such as ontology. In this paper, we describe initial work on a general software framework for supervised extraction of entities and relations from text. The framework was designed so as to provide the degree of flexibility required by automatic semantic markup tasks for the Semantic Web.

1 Introduction

The knowledge acquisition bottleneck problem, well-known to the Knowledge Management community, is turning the weaving of the Semantic Web (SW) into a hard and slow process. Nowadays' high costs associated with producing two versions of a document – one version for human consumption and another version for machine consumption – prevent the creation of enough metadata to make the SW realizable. There are several potential solutions to the problem. We advocate the use of automated methods for semantic markup, i.e., for mapping parts of unstructured text into a structured representation such as ontology. Such methods are studied within the field of Information Extraction (IE) [1]. Information Extraction has been successfully applied in areas such as sale products indexing [2], job advertisement collection [3] and scientific article collection [4] from the Internet, among several others. For example, the information extraction task in the case of sale products indexing consists in identifying the description, price and seller of the product (among other features) within the textual information in product web pages.

Not only may IE contribute to the SW by enabling metadata creation, but also benefit from it. For instance, on the one hand Web mining systems benefit from using an IE component to retrieve metadata from textual information in Web pages, on the other hand applying extraction techniques to real-world web mining problems provides requirements not foreseen by traditional IE scenarios. An extractor can be learned for problems like "find any publications of a faculty member given its name and affiliation" or "find any paintings of a painter given its name". These are actually two of the tasks currently performed by the Armadillo web mining system [5]. Applying IE within the context of such a system allows to draw valuable requirements for real-world practical solutions to the information extraction problem. A promising approach is that of designing new algorithms that take advantage of existing ontologies and metadata in the Semantic Web so as to aid the extraction process.

Machine learning-based approaches to IE induce a set of extraction patterns from text that contains examples of named entities and the relations between those entities. Most of the existing approaches are supervised, i.e. the examples are previously marked-up and there is a training phase prior to extraction. Roth and Yih [6] propose a probabilistic approach for recognizing relations and entities in sentences taking into account mutual dependencies among them. Zelenko et al. [7] uses Support Vector Machines (SVM) and a kernel method to classify a tree-based representation of input sentences. Suzuki et al. [8] also uses a kernel method, but works on a graph-based representation. An unsupervised approach can be found in [9].

In this paper, we describe initial work on a general software framework for supervised extraction of entities and relations from text. The framework was designed so as to provide the degree of flexibility required by automatic semantic markup tasks for the Semantic Web. In this paper we restrict the description of the framework to two important features: its parametrizable plug-in architecture and the graph-based, adaptable data model used for representing the corpus.

2 The Trainable Relation Extraction Framework

The Trainable Relation Extraction framework (T-Rex) [10] has been developed as a testbed for experimenting with several extraction algorithms and several extraction scenarios, especially extraction from the web. The framework promotes the adoption of a divide and conquer approach, by delimiting subproblems that can be worked upon separately in order to improve the overall system.

While in many IE systems data representation and algorithm are tightly coupled, T-Rex features a canonical graph-based data model used by all algorithms implemented within the framework. A graph representation offers several advantages. Most notably it easily accommodates hierarchical representations and it ensures uniformity in the representation of several artifact types. For instance, T-Rex's data model allows expressing in a uniform way arbitrary links between subgraphs, such as co-reference links, grammar links, links related to HTML formatting and the annotations of relations provided by the user. Another advantage is promoting the rapid prototyping of new algorithms given that potentially all the features can be captured in the data model and reused. Figure 1 depicts an example of T-Rex's data model.

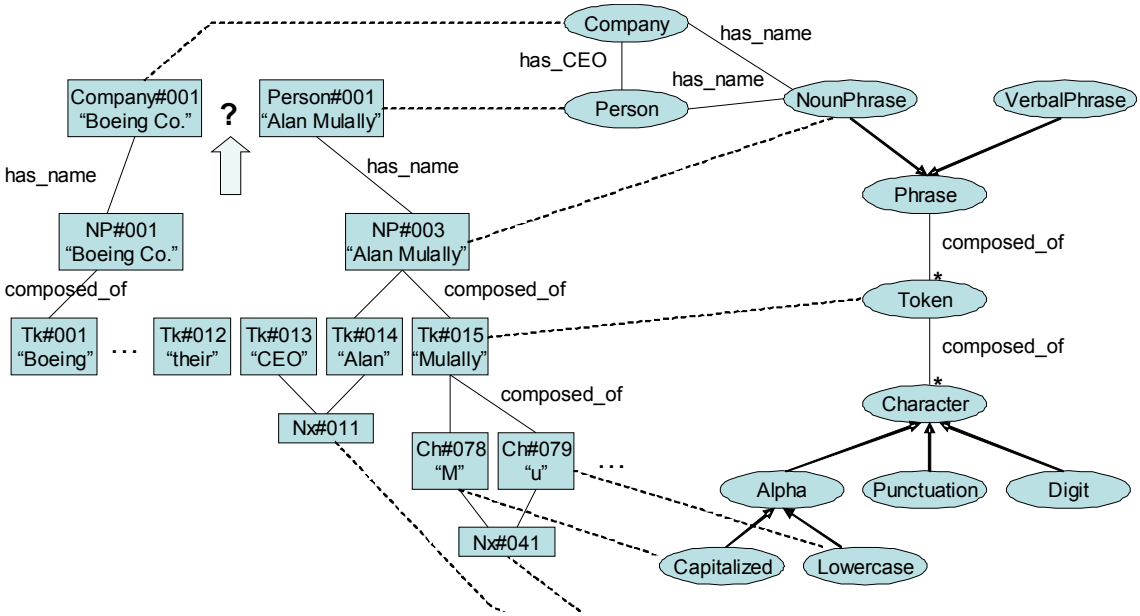

Figure 1. A simplified example of the proposed representation. Only very few nodes and edges of the graph are shown. The arrowed edges represent the usual "isa" relation, the dashed edges the "instance_of" relation. The corpus representation is on the left, while on the right is depicted the ontology that defines the structure of the intended data model representation. The original sentence is "Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results". Goal of the relation extraction algorithm is to infer the relation "has_CEO" between company "Boeing Co." and person "Alan Mulally" from the features in the representation.

The proposed representation is hierarchical in the sense that it is able to model the corpus at several levels, ranging from the character level to the document level, including the token, phrasal and sentence levels. On the other hand, the representation is uniform in the sense that it is able to capture various sets of features that characterize the corpus in diverse ways. For instance, in order to model the HTML formatting of the input documents, the ontological model in Figure 1 could be expanded to include entities like "header", "body", "table", etc. Also, the representation would be able to smoothly incorporate the input of a co-reference module, by including an edge between nodes "Boeing" and "their" in the example. Finally, such representation is also well suited to integrate such additional features as those provided by Semantic Web ontologies.

T-Rex features a modular architecture (see Figure 2). For each component type, e.g. Processor, Classifier, Combiner, there are several actual plug-ins implemented. Therefore, T-Rex supports several IE scenarios by mere parametrization by the user. Notice that the framework promotes a division into roughly two subsystems: the Processing subsystem composed of processors and feature extractors, and the Classification subsystem

composed of classifiers, feature selection algorithms and predictions' combiners. The boundary between these two sets of subsystems is defined by the data model. The Processing subsystem can be viewed as the NLP-dependent part of the framework, where corpora is analysed by one or more linguistic tools and a representation is put together into the data model. The Classification subsystem, on the other hand, can be seen as the ML-dependent part of the framework, where one or more classifiers can be run on the datasets generated from the data model features.

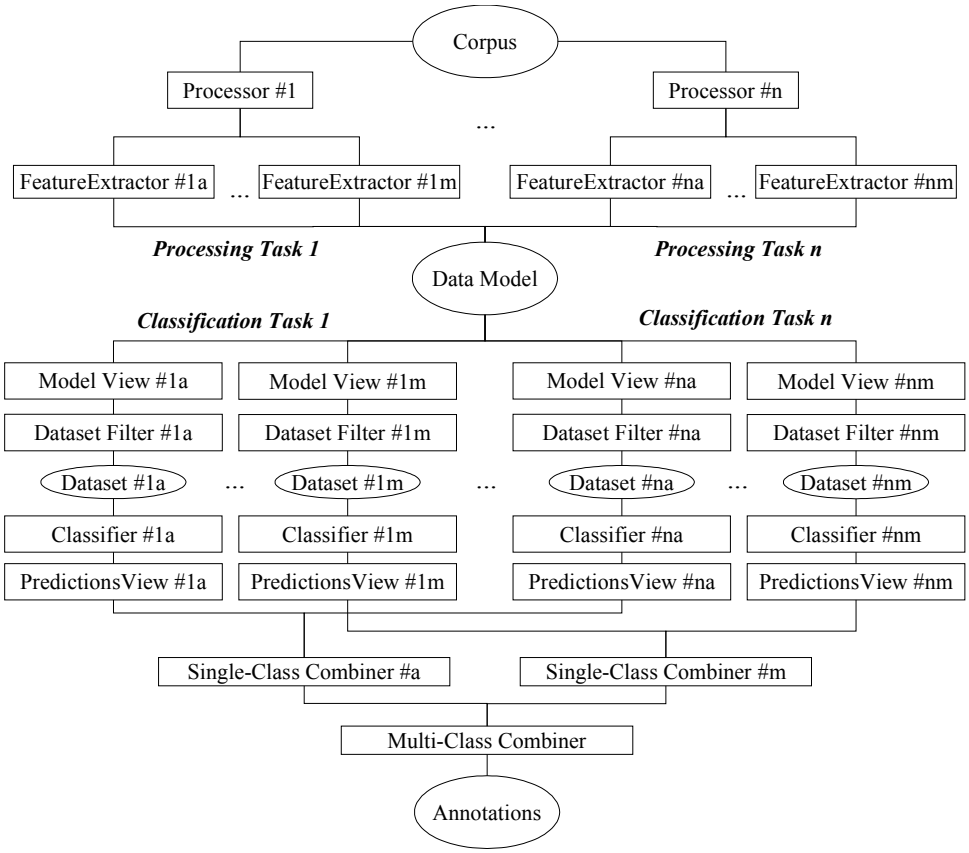

Figure 2. The architecture of T-Rex. Ellipses depict data while rectangles depict functional units. The *modus operandi* is roughly as follows. The corpus is first processed by a collection of NLP tools into a canonical data model which gathers all features extracted from the text. The data model is then transformed into datasets suitable for the collection of classifiers used. Finally, predictions are combined, first on a single-class, and then multi-class basis and the final annotations are produced.

3 Conclusions and Future Work

The current state-of-the-art accuracy for entity extraction tasks is generally around 90 percent for many systems and domains [11].The state-of-the-art performs even worse respecting relation extraction tasks. For instance, the top performer in the 2002 DARPA ACE evaluation got entity extraction scores of about 80%, but relation extraction scores of only roughly 60% [12]. Unfortunately, using a system that makes nearly one mistake out of two suggestions is hardly acceptable in real-world applications like that of producing semantic markup for the Semantic Web.

As future work, we will continue working towards more accurate IE algorithms implemented within the framework. Furthermore, we intend to soon start evaluating the use of T-Rex within the Armadillo web mining system.

References

1. N. Kushmerick and B. Thomas. Adaptive information extraction: Core technologies for information agents. In Intelligent Information Agents R&D in Europe: An AgentLink perspective (Klusch, Bergamaschi, Edwards & Petta, eds.). *Lecture Notes in Computer Science 2586*, Springer, 2003.
2. www.froogle.google.com
3. www.flipdog.com
4. S. Lawrence, C.L. Giles, K. Bollacker. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, Volume 32, Number 6, pp. 67-71, 1999.
5. F. Ciravegna, A. Dingli, D. Guthrie and Y. Wilks. Mining Web Sites Using Unsupervised Adaptive Information Extraction. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary, April 2003.
6. D. Roth and W. Yih. Probabilistic Reasoning for Entity and Relation Recognition. *Proceedings of the 20th International Conference on Computational Linguistics*, 2002.
7. D. Zelenko, C. Aone, A. Richardella. Kernel Methods for Relation Extraction. *JMLR Special Issue on Machine Learning Methods for Text and Images*. 3(Feb):1083-1106, 2003.
8. J. Suzuki, T. Hirao, Y. Sasaki, and E. Maeda. Hierarchical Directed Acyclic Graph Kernel - Methods For Structured Natural Language Data. In *Proceedings of the 41th Annual Meeting of Association for Computational Linguistics (ACL2003)*, 32--39, 2003.
9. R. Yangarber. Acquisition of Domain Knowledge. M.T. Pazienza (Ed.): SCIE 2002, LNAI 2700, pp. 1–28, Springer-Verlag, 2003.
10. J. Iria. T-Rex: A Flexible Relation Extraction Framework. In *Proceedings of the 8th Annual Colloquium for the UK Special Interest Group for Computational Linguistics (CLUK'05)*, Manchester, January 2005.
11. A. McCallum and D. Jensen. A Note on the Unification of Information Extraction and Data Mining using Conditional-Probability, Relational Models. *Workshop on Learning Statistical Models from Relational Data at IJCAI'03*, 2003.
12. DARPA. Darpa automatic content extraction program. 2002.