T-Rex: A Flexible Relation Extraction Framework

José Iria

Department of Computer Science, The University of Sheffield, UK

{j.iria@dcs.shef.ac.uk}

1 Introduction

In the wake of the explosive growth in the use of the computer as a communication device, has come a need for systems that help people cope with the sheer volume of information available. It is universally known that the Internet contains vast amounts of unstructured documents, but the same is also true for large organizations like publishing companies, government departments, airplane manufacturers, car manufacturers, and so forth. In many application domains, there is the potential to significantly increase the utility of available textual information by using automated methods for mapping parts of the unstructured text into a structured representation. This process is called Information Extraction (IE).

Within IE, the task of Entity Extraction is essentially a classification problem: given a piece of text in a document, the task consists in deciding whether it fits into some entity class. The task of Relation Extraction (REX), also known as event extraction or template filling, additionally aims to establish relations between the classified entities. The top performer in the 2002 DARPA ACE evaluation got entity extraction precision and recall scores of about 80%, but binary relation extraction scores of only roughly 60%. Using a system that makes nearly one mistake out of two suggestions is hardly acceptable in real-world applications. Relation extraction is therefore a difficult open research problem, with important applications in diverse fields, such as Knowledge Management and Web Mining.

2 T-Rex: Trainable Relation Extraction framework

The Trainable Relation Extraction framework has been developed as a testbed for experimenting with several algorithms for relation extraction. 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. For instance, by just concentrating in improving the *Combiner* component in the framework it was possible to get a substantial improvement in some of the corpora. The framework is general enough to support a variety of IE algorithms. As a first test, an entity extraction algorithm based on support vector machines was quickly developed using the framework. The algorithm achieves state-of-the-art results on typical corpora for the task.

The following sections describe the architecture of the T-Rex framework, as well the data representation model. A comparison of results obtained against those of the Amilcare [Cira01] system for a corpus typically for evaluation of IE systems is also presented.

2.1 Architecture

The architecture of T-Rex is depicted in Figure 2.1.

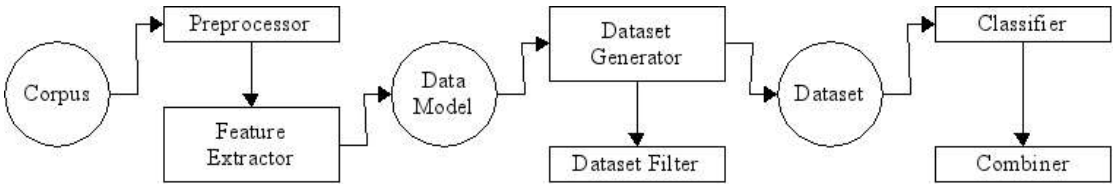

Figure 2.1 High-level view of T-Rex's architecture

T-Rex features a modular architecture. For each component type, e.g. *Processor*, *Classifier*, *Combiner*, there exist several components implemented. The idea is to have a framework generic enough to support the implementation of several types of Information Extraction systems by mere parametrization of the components. Notice that the framework promotes a division into roughly two sets of subsystems: the first, called the Processing system, is composed of a *Processor* component and a *FeatureExtractor* component, whilst the second, called the Classification system, is composed of a *DatasetGenerator*, a *FeatureFilter* and a *Classifier* component. 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 in 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 dataset generated from the data model.

2.2 The Data Model

The way the corpus is represented greatly affects the characteristics and capabilities of systems. In fact, in many systems representation and algorithm are tightly coupled, making it impossible to say whether the results obtained are due to the algorithm or the representation chosen. I am studying the use of graph-based representations for relation extraction. Employing a graph-based representation offers several advantages. It allows expressing in a uniform way arbitrary links between subgraphs, for example co-reference links, grammatical links, links related to HTML formatting and even the annotations of relations in the text (in the case of supervised learning). Additionally, it allows for the rapid prototyping of new algorithms given that potentially all the features can be captured in the representation and treated uniformly.

The data model is instantiated for a particular corpus by referring to the ontological model that gives structure to the intended representation. The model is depicted on the right hand side of Figure 2.2. It comprises both domain specific entities like “company” and “person” as well as syntactic/grammatical entities such as “phrase” and “token”. Besides entities, binary relations can also be modelled, as exemplified by the “next” and “adjacent” relations (in fact, all relations

are nodes in the graph, but for the sake of exposition only the relation "next" was depicted as such). On the left hand side of Figure 2.2 is depicted the corpus representation instantiated for some given corpus. The goal of the relation extraction algorithm is to infer from the features in the representation the relation "has_CEO" between Company "Boeing Co." and Person "Alan Mulally".

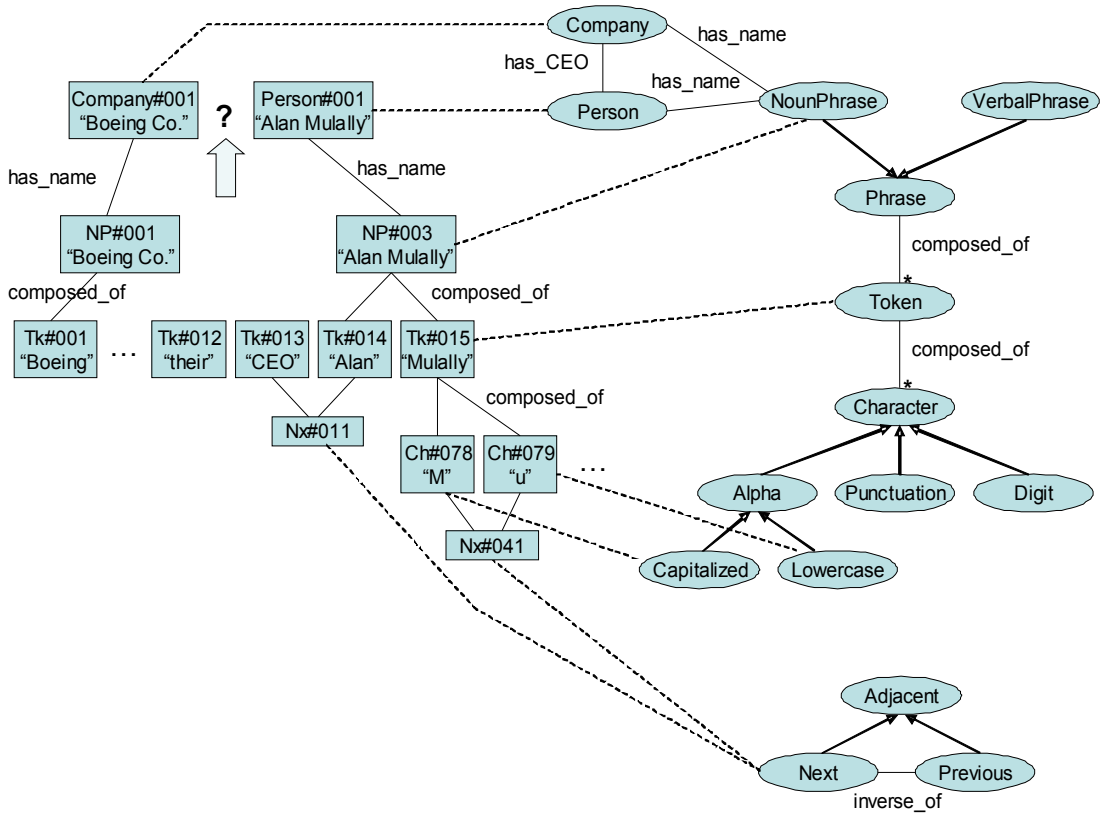

Figure 2.2. - A simplified example of the proposed corpus representation. Only very few nodes and edges of the graph are shown. The arrowed edges represent the usual "isa" relation, the dashed edges the "instance_of" relation. The corpus representation is on the left, while on the right is depicted the ontology that defines the structure of the intended corpus 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. 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 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 "Boeing" and "their" in the example. Finally, a graph-representation is also well suited to integrate such additional features as those provided by Semantic Web ontologies.

2.3 Evaluation

This section presents the results obtained by T-Rex in the "Seminar Announcements" corpus, typically used in the literature for evaluating the performance IE systems. In order to obtain the results shown bellow, T-Rex was parametrized so that it resembles the functionality of algorithm described in [FiKus04]. The following tables present the results obtained for both corpora. An extra column was added to show the f-measure obtained by Amilcare.

```
Classifier:          svmlight (-v 0 -t 0 -j 10)
Feature extractor:   extracts hierarchical features (with spaces, removing duplicates):
                     category orth string
Dataset generator:   GeneratorWholeWindow, 7 tokens window
Combiner:            Confidence-based predictions pairer (confThreshold: -0.5,
                     avgConfThreshold: 0.0)
Validation method:   10 predefined splits
```

Tag	*Precision*	*Recall*	*F-measure*	*Amilcare*
<stime>	94	96.75	95.35	92.73
<etime>	96.23	94.88	95.55	96.14
<location>	84.17	76.63	80.22	77.1
<speaker>	81.33	78.03	79.65	82.94
Micro-average	88.94	86.89	87.91	87.09

Table 2.1 - Results obtained by T-Rex on the Seminar Corpus

These results are consistent with the state of the art and slightly outperform those of Amilcare.

3 Future Work

As future work, I will consolidate the data model within T-Rex and will find out what types of features are best suited for the extraction of relations from textual information, covering different domains and different input structures. Secondly, I will implement new algorithms for relation extraction into the framework that use such representation formalism, borrowing insights from the successful approach introduced by the LP^2 algorithm [Cira01], but overcoming its limitations.

4 References

[Cira01] F. Ciravegna. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, Seattle, August 2001.

[FiKus04] A. Finn and N. Kushmerick. Multi-level Boundary Classification for Information Extraction. In *Proceedings of the European Conference on Machine Learning,* Pisa, 2004.