REEL: A Relation Extraction Learning Framework

Pablo Barrio
Columbia University
pjbarrio@cs.columbia.edu

Gonçalo Simões
INESC-ID and IST, Universidade de Lisboa
goncalo.simoes@tecnico.ulisboa.pt

Helena Galhardas
INESC-ID and IST, Universidade de Lisboa
helena.galhardas@tecnico.ulisboa.pt

Luis Gravano
Columbia University
gravano@cs.columbia.edu

ABSTRACT

We introduce the REEL (**RE**lation **E**xtraction **L**earning) framework, an open source framework that facilitates the development and evaluation of relation extraction systems over text collections. To define a relation extraction system for a new relation and text collection, users only need to specify the parsers to load the collection, the relation and its constraints, and the learning and extraction techniques to be used. This makes REEL a powerful framework to enable the deployment and evaluation of relation extraction systems for both application building and research.

1. INTRODUCTION

Relation extraction systems are sophisticated information extraction tools that automatically discover structured relations between entities in natural language text. For example, a properly trained relation extraction system would extract the tuple ⟨Mark Chapman, second-degree murder, 1981⟩ of the relation *Charged*(*Person*, *Charge*, *Date*) from the text excerpt "John Lennon's killer, Mark Chapman, was sentenced in 1981 to 20 years to life in prison after pleading guilty to second-degree murder." To extract such structured information from text documents, state-of-the-art relation extraction systems usually employ a variety of text processing tools, such as entity recognition and part-of-speech tagging, and many times require enforcing constraints on the extracted information, such as requiring that extracted entities be of a certain type or that entities in an extracted relation be mentioned within N words of each other [1].

Many relation extraction systems have been proposed in the literature [1]. However, few such systems are publicly available and, even when they are, it is usually problematic to adapt and evaluate them over new relations and text collections. To avoid implementing such complex systems from scratch, developers often rely on toolkits. One such toolkit is T-Rex [2], which splits the relation extraction task into relatively coarse modules for text processing and learning. Such coarse modules are hard to reuse across relation extraction tasks, and hence complicate the implementation of new systems. Also, T-Rex does not impose restrictions on the output of its modules, which complicates the experimental comparison of different extraction strategies and their output. As a result, to experimentally evaluate and compare relation extraction systems in T-Rex, developers must rely on ad-hoc solutions, which is far from ideal.

Other toolkits originally proposed for related text-centric tasks, such as text processing (e.g., UIMA [3]), machine learning (e.g., MALLET [4]), natural language processing (e.g., StanfordNLP [5], GATE [6]), and entity extraction (e.g., MinorThird [7]) provide low-level building blocks that are helpful for relation extraction, but lack the code and infrastructure to directly support relation extraction. To use these frameworks for relation extraction we could extend them by including the infrastructural elements missing in each framework. However, this would require in many cases a significant implementation effort and a drastic redesign of the toolkits, since we would have to incorporate full support for the missing pieces. A more promising approach, which we advocate in our work, is to integrate and complement valuable text processing toolkits—to exploit their powerful implementations of low-level text operations—and machine learning toolkits—to exploit their powerful implementations of relevant learning operations—for our extraction task.

2. THE REEL FRAMEWORK

We introduce the REEL (**RE**lation **E**xtraction **L**earning) framework, an open-source framework—publicly available at http://reel.cs.columbia.edu/ under the General Public License Version 3 (GPLv3)—to easily develop and evaluate relation extraction systems. REEL provides the code and infrastructure to: (i) handle various input text formats, enabling operations over different text collections; (ii) plug in appropriate text processing steps and tools, enabling diverse processing of the text with minimal effort; (iii) define and combine conceptual relation constraints that are automatically enforced; (iv) decouple learning and extraction from the text processing, enabling the straightforward integration and reusability of different extraction algorithms; and (v) uniformly execute and evaluate relation extraction systems, enabling the testing and fair assessment of these systems.

In contrast to existing toolkits, REEL effectively modularizes the key components involved in relation extraction. To define an extraction system for a new relation and new text collections, users only need to specify the parsers to load the collections, the relation and its constraints, and the learning and extraction techniques, which makes REEL a powerful

978-1-4799-5569-5/14/$31.00

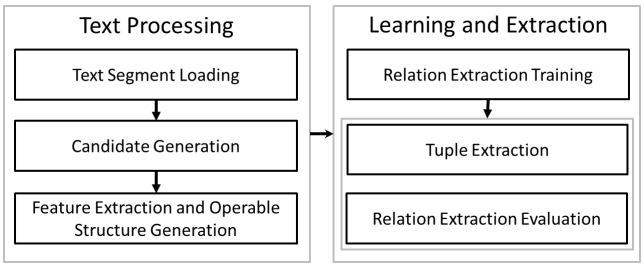

Figure 1: REEL system architecture.

framework to enable the deployment of relation extraction systems for both research and application building. Figure 1 shows REEL's flexible, modular architecture, with two types of components. The Text Processing components transform the input text documents into the input format for the relation extraction techniques. Then, the Learning and Extraction components learn, execute, and evaluate classification models that perform relation extraction.

The Text Processing components include the Text Segment Loading, Candidate Generation, and Feature Extraction and Operable Structure Generation components (see Figure 1). First, the Text Segment Loading component loads the documents in a text collection and transforms them into *text segments* (e.g., sentences) tagged with named entities. The Text Segment Loading component enables the integration of different text processing tools (e.g., XML parsers) to easily allow different types of collections. Second, the Candidate Generation component produces *candidate text sentences*. In a nutshell, a candidate text sentence is a mention of a potential tuple in a sentence that fulfills predefined constraints. This component enables a flexible definition of constraints over entities and relations (e.g., entities need to be of a certain type, say, *Person*, or entities need to be mentioned within N words of each other). Finally, the Feature Extraction and Operable Structure Generation component extracts the features required by a specific relation extraction algorithm [8, 9] and produces the data structures for the extraction algorithm (e.g., sequences or trees of features). These data structures, or *operable structures*, are a feature-enriched version of the candidate text segments on which the learning and extraction algorithms will operate. To produce operable structures, REEL provides a unified interface for extracting a wide and extensible variety of features and structures that different learning algorithms may require.

The Learning and Extraction components include the Relation Extraction Training, Tuple Extraction, and Relation Extraction Evaluation components. First, the Relation Extraction Training component automatically produces a *relation extraction model* using, as training input, labeled operable structures, which indicate whether the relation of interest holds among their entities. Second, the Tuple Extraction component uses this model to extract tuples corresponding to related entities. Notably, Tuple Extraction performs a classification task over unlabeled operable structures and produces tuples of entities that are likely related. Third, the Relation Extraction Evaluation component evaluates the relation extraction systems according to an easily extensible set of evaluation metrics. The most important characteristic of the learning and extraction components is that they provide a unified interface for different relation extraction techniques. This interface helps to train, execute, and evaluate the resulting models for different relation extraction techniques with minor changes in the code.

3. CONCLUSIONS

We introduced REEL, an open-source framework to easily develop and evaluate relation extraction systems. REEL provides end-to-end infrastructure to perform relation extraction tasks, and leverages powerful existing toolkits for both text processing and learning. Moreover, REEL effectively addresses the complex requirements of relation extraction and helps developers and researchers produce simple and easy-to-understand source code for their relation extraction systems. As part of the REEL distribution—available at http://reel.cs.columbia.edu/ as open source under the General Public License Version 3 (GPLv3)—we have included ready-to-use systems (e.g., [8, 9]); we have also integrated several text processing and learning toolkits, to illustrate how to incorporate and leverage external algorithms and toolkits. For further details about REEL, please refer to [10].

Acknowledgments: This material is based upon work supported by the National Science Foundation under Grant IIS-08-11038. This work was also supported by *Fundação para a Ciência e a Tecnologia*, under Project PEst-OE/EEI/LA0021/2013 and Ph.D. Grant SFRH/BD/61393/2009.

4. REFERENCES

[1] S. Sarawagi, "Information extraction," *Found. and Trends in Databases*, vol. 1, no. 3, pp. 261–377, Mar. 2008.

[2] J. Iria, "T-Rex: A flexible relation extraction framework," in *Proc. 8th Annu. Colloq. UK Special Interest Group for Computational Linguistics*, Manchester, UK, 2005.

[3] D. Ferrucci and A. Lally, "UIMA: An architectural approach to unstructured information processing in the corporate research environment," *Natural Language Eng.*, vol. 10, no. 3-4, pp. 327–348, Sep. 2004.

[4] A. K. McCallum. (2002) MALLET: A machine learning for language toolkit. [Online]. Available: http://mallet.cs.umass.edu

[5] C. D. Manning *et al.*, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting of the Assoc. Computational Linguistics: System Demonstrations*, Baltimore, MD, 2014, pp. 55–60.

[6] H. Cunningham *et al.*, *Text processing with GATE (version 6)*. Murphys, CA: Gateway Press CA, 2011.

[7] W. W. Cohen. (2004) Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. [Online]. Available: http://minorthird.sourceforge.net

[8] R. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *Proc. Conf. on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, Canada, 2005, pp. 724–731.

[9] R. Bunescu and R. J. Mooney, "Subsequence kernels for relation extraction," *Advances in Neural Inform. Process. Syst.*, vol. 18, pp. 171–178, May 2006.

[10] P. Barrio *et al.*, "REEL: A relation extraction learning framework," INESC-ID, Lisbon, Portugal, Tech. Rep. 15/2014, Jun. 2014. [Online]. Available: http://www.inesc-id.pt/ficheiros/publicacoes/10191.pdf