NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools

Giuseppe Rizzo
EURECOM / Sophia Antipolis, France
Politecnico di Torino / Turin, Italy
giuseppe.rizzo@eurecom.fr

Raphaël Troncy
EURECOM / Sophia Antipolis, France
raphael.troncy@eurecom.fr

Abstract

Named Entity Extraction is a mature task in the NLP field that has yielded numerous services gaining popularity in the Semantic Web community for extracting knowledge from web documents. These services are generally organized as pipelines, using dedicated APIs and different taxonomy for extracting, classifying and disambiguating named entities. Integrating one of these services in a particular application requires to implement an appropriate driver. Furthermore, the results of these services are not comparable due to different formats. This prevents the comparison of the performance of these services as well as their possible combination. We address this problem by proposing NERD, a framework which unifies 10 popular named entity extractors available on the web, and the NERD ontology which provides a rich set of axioms aligning the taxonomies of these tools.

1 Introduction

The web hosts millions of unstructured data such as scientific papers, news articles as well as forum and archived mailing list threads or (micro-)blog posts. This information has usually a rich semantic structure which is clear for the human being but that remains mostly hidden to computing machinery. Natural Language Processing (NLP) tools aim to extract such a structure from those free texts. They provide algorithms for analyzing atomic information elements which occur in a sentence and identify Named Entity (NE) such as name of people or organizations, locations, time references or quantities. They also classify these entities according to predefined schema increasing discoverability (e.g. through faceted search) and reusability of information.

Recently, research and commercial communities have spent efforts to publish NLP services on the web. Beside the common task of identifying POS and of reducing this set to NEs, they provide more and more disambiguation facility with URIs that describe web resources, leveraging on the web of real world objects. Moreover, these services classify such information using common ontologies (e.g. DBpedia ontology[1] or YAGO[2]) exploiting the large amount of knowledge available from the web of data. Tools such as AlchemyAPI[3], DBpedia Spotlight[4], Evri[5], Extractiv[6], Lupedia[7], OpenCalais[8], Saplo[9], Wikimeta[10], Yahoo! Content Extraction[11] and Zemanta[12] represent a clear opportunity for the web community to increase the volume of interconnected data. Although these extractors share the same purpose - extract NE from text, classify and disambiguate this information - they make use of different algorithms and provide different outputs.

This paper presents NERD (Named Entity Recognition and Disambiguation), a framework that unifies the output of 10 different NLP extrac-

[1] http://wiki.dbpedia.org/Ontology
[2] http://www.mpi-inf.mpg.de/yago-naga/yago
[3] http://www.alchemyapi.com
[4] http://dbpedia.org/spotlight
[5] http://www.evri.com/developer
[6] http://extractiv.com
[7] http://lupedia.ontotext.com/
[8] http://www.opencalais.com
[9] http://www.saplo.com/
[10] http://www.wikimeta.com
[11] http://developer.yahoo.com/search/content/V2/contentAnalysis.html
[12] http://www.zemanta.com

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 73–76,
Avignon, France, April 23 - 27 2012.

tors publicly available on the web. Our approach relies on the development of the NERD ontology which provides a common interface for annotating elements, and a web REST API which is used to access the unified output of these tools. We compare 6 different systems using NERD and we discuss some quantitative results. The NERD application is accessible online at `http://nerd.eurecom.fr`. It requires to input a URI of a web document that will be analyzed and optionally an identification of the user for recording and sharing the analysis.

2 Framework

NERD is a web application plugged on top of various NLP tools. Its architecture follows the REST principles and provides a web HTML access for humans and an API for computers to exchange content in JSON or XML. Both interfaces are powered by the NERD REST engine. The Figure 2 shows the workflow of an interaction among clients (humans or computers), the NERD REST engine and various NLP tools which are used by NERD for extracting NEs, for providing a type and disambiguation URIs pointing to real world objects as they could be defined in the Web of Data.

2.1 NERD interfaces

The web interface[13] is developed in HTML/-Javascript. It accepts any URI of a web document which is analyzed in order to extract its main textual content. Starting from the raw text, it drives one or several tools to extract the list of Named Entity, their classification and the URIs that disambiguate these entities. The main purpose of this interface is to enable a human user to assess the quality of the extraction results collected by those tools (Rizzo and Troncy, 2011a). At the end of the evaluation, the user sends the results, through asynchronous calls, to the REST API engine in order to store them. This set of evaluations is further used to compute statistics about precision scores for each tool, with the goal to highlight strengths and weaknesses and to compare them (Rizzo and Troncy, 2011b). The comparison aggregates all the evaluations performed and, finally, the user is free to select one or more evaluations to see the metrics that are computed for each service in real time. Finally, the application contains a help page that provides guidance and details about the whole evaluation process.

The API interface[14] is developed following the REST principles and aims to enable programmatic access to the NERD framework. GET, POST and PUT methods manage the requests coming from clients to retrieve the list of NEs, classification types and URIs for a specific tool or for the combination of them. They take as inputs the URI of the document to process and a user key for authentication. The output sent back to the client can be serialized in JSON or XML depending on the content type requested. The output follows the schema described below (in the JSON serialization):

```
entities: [{
   "entity":"Tim Berners-Lee",
   "type":"Person",
   "uri":"http://dbpedia.org/resource/
      Tim_berners_lee",
   "nerdType":"http://nerd.eurecom.fr/
      ontology#Person",
   "startChar": 30,
   "endChar": 45,
   "confidence": 1,
   "relevance": 0.5
}]
```

2.2 NERD REST engine

The REST engine runs on Jersey[15] and Grizzly[16] technologies. Their extensible framework allows to develop several components, so NERD is composed of 7 modules, namely: authentication, scraping, extraction, ontology mapping, store, statistics and web. The authentication enables to log in with an OpenID provider and subsequently attaches all analysis and evaluations performed by a user with his profile. The scraping module takes as input the URI of an article and extracts its main textual content. Extraction is the module designed to invoke the external service APIs and collect the results. Each service provides its own taxonomy of named entity types it can recognize. We therefore designed the NERD ontology which provides a set of mappings between these various classifications. The ontology mapping is the module in charge to map the classification type retrieved to the NERD ontology. The store module saves all evaluations according to the schema model we defined in the

[13] `http://nerd.eurecom.fr`

[14] `http://nerd.eurecom.fr/api/application.wadl`

[15] `http://jersey.java.net`

[16] `http://grizzly.java.net`

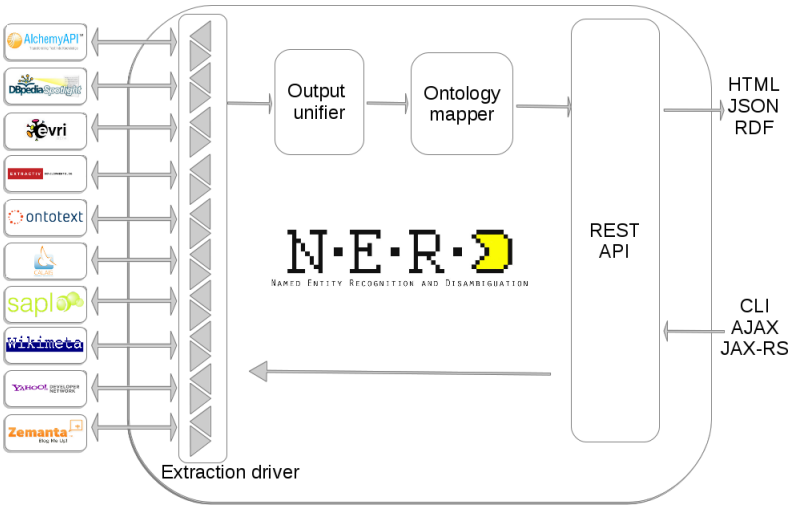

Figure 1: A user interacts with NERD through a REST API. The engine drives the extraction to the NLP extractor. The NERD REST engine retrieves the output, unifies it and maps the annotations to the NERD ontology. Finally, the output result is sent back to the client using the format reported in the initial request.

NERD database. The statistic module enables to extract data patterns from the user interactions stored in the database and to compute statistical scores such as Fleiss Kappa and precision/recall analysis. Finally, the web module manages the client requests, the web cache and generates the HTML pages.

3 NERD ontology

Although these tools share the same goal, they use different algorithms and their own classification taxonomies which makes hard their comparison. To address this problem, we have developed the NERD ontology which is a set of mappings established manually between the schemas of the Named Entity categories. Concepts included in the NERD ontology are collected from different schema types: ontology (for DBpedia Spotlight and Zemanta), lightweight taxonomy (for AlchemyAPI, Evri and Wikimeta) or simple flat type lists (for Extractiv, OpenCalais and Wikimeta). A concept is included in the NERD ontology as soon as there are at least two tools that use it. The NERD ontology becomes a reference ontology for comparing the classification task of NE tools. In other words, NERD is a set of axioms useful to enable comparison of NLP tools. We consider the DBpedia ontology exhaustive enough to represent all the concepts involved in a NER task. For all those concepts that do not appear in the NERD namespace, there are just sub-classes of parents that end-up in the NERD ontology. This ontology is available at `http://nerd.eurecom.fr/ontology`.

We provide the following example mapping among those tools which defines the `City` type: the `nerd:City` class is considered as being equivalent to `alchemy:City`, `dbpedia-owl:City`, `extractiv:CITY`, `opencalais:City`, `evri:City` while being more specific than `wikimeta:LOC` and `zemanta:location`.

```
nerd:City a rdfs:Class ;
  rdfs:subClassOf wikimeta:LOC ;
  rdfs:subClassOf zemanta:location ;
  owl:equivalentClass alchemy:City ;
  owl:equivalentClass dbpedia-owl:City ;
  owl:equivalentClass evri:City ;
  owl:equivalentClass extractiv:CITY ;
  owl:equivalentClass opencalais:City .
```

4 Ontology alignment results

We conducted an experiment to assess the alignment of the NERD framework according to the ontology we developed. For this experiment, we collected 1000 news articles of The New York Times from 09/10/2011 to 12/10/2011 and we performed the extraction of named entities with the tools supported by NERD. The goal is to explore the NE extraction patterns with this dataset and to assess commonalities and differences of the classification schema used. We propose the alignment of the 6 main types recognized by all tools using the NERD ontology. To conduct this experiment, we used the default configuration for all tools used. We define the following variables:

	AlchemyAPI	DBpedia Spotlight	Evri	Extractiv	OpenCalais	Zemanta
Person	6,246	14	2,698	5,648	5,615	1,069
Organization	2,479	-	900	81	2,538	180
Country	1,727	2	1,382	2,676	1,707	720
City	2,133	-	845	2,046	1,863	-
Time	-	-	-	123	1	-
Number	-	-	-	3,940	-	-

Table 1: Number of axioms aligned for all the tools involved in the comparison according to the NERD ontology for the sources collected from the *The New York Times* from 09/10/2011 to 12/10/2011.

the number n_d of evaluated documents, the number n_w of words, the total number n_e of entities, the total number n_c of categories and n_u URIs. Moreover, we compute the following metrics: word detection rate $r(w, d)$, i.e. the number of words per document, entity detection rate $r(e, d)$, i.e. the number of entities per document, entity detection rate per word, i.e. the ratio between entities and words $r(e, w)$, category detection rate, i.e. the number of categories per document $r(c, d)$ and URI detection rate, i.e. the number of URIs per document $r(u, d)$. The evaluation we performed concerned $n_d = 1000$ documents that amount to $n_w = 620,567$ words. The word detection rate per document $r(w, d)$ is equal to 620.57 and the total number of recognized entities n_e is $164,12$ with the $r(e, d)$ equal to 164.17. Finally $r(e, w)$ is 0.0264, $r(c, d)$ is 0.763 and $r(u, d)$ is 46.287.

Table 1 shows the classification comparison results. DBpedia Spotlight recognizes very few classes. Zemanta increases significantly classification performances with respect to DBpedia obtaining a number of recognized Person which is two magnitude order more important. AlchemyAPI has strong ability to recognize Person and City while obtaining significant scores for Organization and Country. OpenCalais shows good results to recognize the class Person and a strong ability to classify NEs with the label Organization. Extractiv holds the best score for classifying Country and it is the only extractor capable of extracting the classes Time and Number.

5 Conclusion

In this paper, we presented NERD, a framework developed following REST principles, and the NERD ontology, a reference ontology to map several NER tools publicly accessible on the web. We propose a preliminary comparison results where we investigate the importance of a reference ontology in order to evaluate the strengths and weaknesses of the NER extractors. We will investigate whether the combination of extractors may overcome the performance of a single tool or not. We will demonstrate more live examples of what NERD can achieve during the conference. Finally, with the increasing interest of interconnecting data on the web, a lot of research effort is spent to aggregate the results of NLP tools. The importance to have a system able to compare them is under investigation from the NIF[17] (NLP Interchange Format) project. NERD has recently been integrated with NIF (Rizzo and Troncy, 2012) and the NERD ontology is a milestone for creating a reference ontology for this task.

Acknowledgments

This paper was supported by the French Ministry of Industry (*Innovative Web* call) under contract 09.2.93.0966, "Collaborative Annotation for Video Accessibility" (ACAV).

References

Rizzo G. and Troncy R. 2011. *NERD: A Framework for Evaluating Named Entity Recognition Tools in the Web of Data*. 10^{th} International Semantic Web Conference (ISWC'11), Demo Session, Bonn, Germany.

Rizzo G. and Troncy R. 2011. *NERD: Evaluating Named Entity Recognition Tools in the Web of Data*. Workshop on Web Scale Knowledge Extraction (WEKEX'11), Bonn, Germany.

Rizzo G., Troncy R, Hellmann S and Bruemmer M. 2012. *NERD meets NIF: Lifting NLP Extraction Results to the Linked Data Cloud*. 5^{th} International Workshop on Linked Data on the Web (LDOW'12), Lyon, France.

[17] `http://nlp2rdf.org`