NERD: Evaluating Named Entity Recognition Tools in the Web of Data

Giuseppe Rizzo[1,2] and Raphaël Troncy[1]

[1] EURECOM, Sophia Antipolis, France, <raphael.troncy@eurecom.fr>
[2] Politecnico di Torino, Torino, Italy, <giuseppe.rizzo@polito.it>

Abstract. The Web of data promotes the idea that more and more data are interconnected. A step towards this goal is to bring more structured annotations to existing documents using common vocabularies or ontologies. Semi-structured texts such as scientific, medical or news articles as well as forum and archived mailing list threads or (micro-)blog posts can hence be semantically annotated. Named Entity (NE) extractors play a key role for extracting structured information by identifying features, also called entities, and by linking them to other web resources by means of typed inferences. In this article, we propose a thorough evaluation of five popular Linked Data entity extractors which expose APIs: AlchemyAPI, DBPedia Spotlight, Extractiv, OpenCalais and Zemanta. We present NERD, an evaluation framework we have developed and the results of a controlled evaluation performed by human beings that consists in assigning a Boolean value to three criteria: entity detection, entity type and entity disambiguation.

Key words: Entity extraction, Linked Data, Natural Language Processing, Evaluation of Linked Data entity extraction tools

1 Introduction

The Web of Data is often illustrated as a fast growing cloud of interconnected datasets representing information about barely everything [5]. The Web hosts also millions of semi-structured texts such as scientific or medical 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 author but that remains mostly hidden to computing machinery. Natural Language Processing (NLP) and information extractors aim to bring back such a structure from those free texts. They provide algorithms for extracting semantic units identifying the name of people, organizations, locations, time references, quantities, etc. and classifying them into predefined categories or content types. They improve the ability of content searching, finding meaningful relationships between the entities extracted.

Since the 90's, an increasing emphasis has been given to the evaluation of NLP techniques. Hence, the Named Entity Recognition (NER) task has been developed as an essential component of the Information Extraction field. In parallel, a number of services have been developed to extract structured information from resources published on the Web. Recently, those tools have been transformed into web services, opening their APIs for public research or commercial use and contributing to the development of a new set of semantic applications. Tools such as AlchemyAPI[3], DBpedia Spotlight[4], Extractiv[5], OpenCalais[6] and Zemanta[7] represent a clear opportunity for the Semantic Web community to increase the volume of interconnected data. Although these tools share the same purpose – extracting semantic units from text – they make use of different algorithms and training data. They generally provide a similar output composed of a set of extracted named entities, their type and potentially a URI disambiguating each named entities ($o = (NE, type, URI)$). These services have their own strengths and shortcomings but, to the best of our knowledge, no scientific evaluation has ever been conducted to understand the conditions under which each tool is the most appropriate. This paper attempts to fill this gap.

We have developed NERD (Named Entity Recognition and Disambiguation), a web-based application which enables human beings to evaluate the five most used Linked Data named entity extractors. The user is invited to *i)* submit a URI of a textual document, *ii)* select a particular tool and *iii)* rate the accuracy of the results following the three criteria ($NE, type, URI$). All user interactions are then stored in a database. We propose a OWL ontology containing the set of mappings of all entity categories that those tools are able to detect. The NERD application uses this ontology and the evaluations submitted by users to generate comparison and analysis reports that take into account the authority and the genre of the web resources or the entity type.

The remainder of this paper is organized as follows. We present an overview of NLP techniques and Linked Data NLP tools in section 2. We briefly describe the architecture of the NERD application in section 3. We detail the evaluation methodology and the results we obtained in section 4. We discuss those results and argue for the development of a gold standard in section 5. Finally, we give our conclusions and outline future work in Section 6.

2 Named Entity Extractors

The goal of a Named Entity (NE) extractor (part of the NLP tools family) is to extract named entities. The first definition of a NE was coined by Grishman *et al.* as an information unit such as the name of a person or an organization, a location, a brand, a product, a numeric expression including time, date, money and percent found in a sentence [7]. Key features to assess NLP tools are configuration variables and output variables. In this section, we provide a brief overview of the state of the art and we distinguish two set of NLP tools: the ones that

[3] http://www.alchemyapi.com

[4] http://dbpedia.org/spotlight

[5] http://extractiv.com

[6] http://www.opencalais.com

[7] http://www.zemanta.com

are just able to identify information units and classifying them in a taxonomy of categories and the ones that can additionally provide a link pointing to a web resource that disambiguates the named entity.

2.1 Information Extraction and NER

One of the first research papers in the NLP field aiming at automatically identifying named entities in texts was proposed by Rau [13]. This work relies on heuristics and definition of patterns to recognize company names in texts. The training set is defined by the set of heuristics chosen. This work evolved and was improved later on by Sekine *et al.* [15]. A different approach was introduced when Supervised Learning (SL) techniques were used. The big disruptive change was the use of a large dataset manually labeled. In the SL field, a human being usually trains positive and negative examples so that the algorithm computes classification patterns. SL techniques exploit Hidden Markov Models (HMM) [3], Decision Trees [14], Maximum Entropy Models [4], Support Vector Machines (SVM) [2] and Conditional Random Fields (CRF) [9]. The common goal of these approaches is to recognize relevant key-phrases and to classify them in a fixed taxonomy. The challenges with SL approaches is the unavailability of such labeled resources and the prohibitive cost of creating examples. Semi-Supervised Learning (SSL) approach and Unsupervised Learning (UL) approach attempt to solve this problem by either providing a small initial set of labeled data to train and seed the system [8], or by resolving the extraction problem as a clustering one. For example, one can try to gather named entities from clustered groups based on the similarity of context. Other unsupervised methods may rely on lexical resources (e.g. WordNet), lexical patterns and statistics computed on large annotated corpus [1].

Besides the different learning approaches, the Named Entity recognition tools vary in terms of the language they can support. While each language has its own syntax and semantics that may affect the way the entities can be extracted, Palmer *et al.* have used statistical methods for finding named entities in newswire articles for Chinese, English, French, Japanese, Portuguese and Spanish [12]. They found that the difficulty of the NER task was different for the six languages but that a large part of the task could be performed with simple methods. However, the results were affected by low F-measure and an absence of mapping between entities to types. In the remaining of this paper, we only consider the English language in order to remove one variable in our evaluation. The NERD framework is however independent of the language relying solely on the capabilities of the underlying named entity extractors.

2.2 NER Web Services

In addition to detect a NE and its type, the NLP community has developed methods to disambiguate the information unit with a valid URI embracing the Linked Data movement. Disambiguation is one of the key challenges in this scenario and its foundation stands on the fact that terms taken in isolation are naturally ambiguous. Hence, a text containing the term `London` may refer to the city `London in UK` or to the city `London in Minnesota, USA`, depending on the surrounding context. Similarly, people, organizations and companies can have multiple names and nicknames. These systems generally try to find in the surrounding text some clues for contextualizing the ambiguous term and refine its intended meaning. Therefore, a NE extraction workflow consists in analyzing some input content for detecting named entities, assigning them a type weighted by a confidence score and by providing a list of URIs for disambiguation. AlchemyAPI, DBpedia Spotlight, Extractiv, OpenCalais and Zemanta all exploit this idea. These web services differentiate from many aspects. All of them, except DBpedia Spotlight, are commercial, although they provide a restricted free access. While AlchemyAPI supports eight different languages, OpenCalais and DBpedia Spotligth support three languages (resp. English, French, Spanish and English, Spanish, Portuguese), the others work only with English content. The taxonomy of named entity types that those tools can extract is also different. DBpedia Spotlight has an exhaustive taxonomy since it adopts the DBpedia classes schema, while Zemanta provides an almost flat list of categories. We will describe in the Section 4.1 how we have interlinked those taxonomies. All those tools are finally capable to provide URI disambiguation, being web resources from datasets part of the LOD cloud (e.g. DBpedia, Freebase, GeoNames, LinkedIMDB) or other resources (e.g. Shopping.com). In the Table 1, we report factual information about these five web services.

	AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
Language Support	English, French, German,Italian, Portuguese, Russian, Spanish, Swedish	English, Spanish, Portuguese	English	English, French, Spanish	English
Entity type number	272	272	6	39	81
LOD Dataset number	7	1	1	9	1

Table 1. Factual information about 5 popular Linked Data NE web services.

2.3 NER Web Services Comparison

The creators of the DBpedia Spotlight service have compared their service with a number of other NER extractors (OpenCalais, Zemanta, Ontos Semantic API[8], The Wiki Machine[9], AlchemyAPI and M&W's wikifier[11]) according to a particular annotation task [10]. The experiment consisted in evaluating 35 paragraphs from 10 articles in 8 categories selected from the "The New York Times" and has been performed by 4 human raters. The final goal was to create wiki links. The

[8] `http://www.ontos.com`

[9] `http://thewikimachine.fbk.eu/`

experiment showed how DBpedia Spotlight overcomes the performance of other services to complete this task. The "golden standard" does not adhere to our requirement because it annotates unit information with just Wikipedia resource and it does not link the annotation to the NE and their type. For this reason, we differentiate from this work by building a proposal for a "golden standard" where we combine NE, type and URI as well as a relevance score of this pattern for the text.

Other attempts of comparisons are stressed in two blog posts. Nathan Rixham[10] and Benjamin Nowack[11] have both reported in their blogs their experiences in developing a prototype using Zemanta and OpenCalais. They observe that Zemanta aims at recommending "tags" for the analyzed content while OpenCalais focuses on the extraction of named entities with their corresponding types. They argue that Zemanta tends to have a higher precision for real things while the performance goes down for less popular topics. When OpenCalais provides a Linked Data identifier or more information about the named entity, it rarely makes a mistake. OpenCalais mints new URIs for all named entities and sometimes provides `sameAs` links with other linked data identifiers. In contrast, Zemanta does not generate new URIs but suggests (multiple) links that represent the best named entity in a particular context. In another blog post, Robert Di Ciuccio[12] reports on a simple benchmarking test of five NER APIs (OpenCalais, Zemanta, AlchemyAPI, Evri, OpenAmplify and Yahoo! Term Extraction) over three video transcripts in the context of ViewChange.org. The author argues that Zemanta was the clear leader of the NLP API field for his tests, observing that OpenCalais was returning highly relevant terms but was lacking disambiguation features and that AlchemyAPI was returning disambiguated results but that the quantity of entities returned was low. Finally, Veeeb provides a simple tool enabling to visualize the raw JSON results of AlchemyAPI, OpenCalais and Evri[13]. Bartosz Malocha developed in EURECOM a similar tool for Zemanta, AlchemyAPI and OpenCalais[14]. We conclude that to the best of our knowledge, there have been very few research efforts that aim to compare systematically and scientifically Linked Data NER services. Our contribution fills this gap. We have developed a framework enabling the human validation of NER web services that is also capable to generate an analysis report under different conditions.

3 NERD: A Framework for Comparing NER Web Services

NERD (Named Entity Recognition and Disambiguation)[15] is a web application plugged on top of various named entities extractors. It allows the user to analyze

[10] `http://webr3.org/blog/experiments/linked-data-extractor-prototype-details/`

[11] `http://bnode.org/blog/2010/07/28/linked-data-entity-extraction-with-zemanta-and-opencalais`

[12] `http://blog.viewchange.org/2010/05/entity-extraction-content-api-evaluation/`

[13] `http://www.veeeb.com/examples/flex/nlpapicompare/nlpCompare.html`

[14] `http://entityextraction.appspot.com/`

[15] `http://semantics.eurecom.fr/nerd`

any textual resource published on the web and accessible with a URI, and to extract from the text the named entities detected, typed and disambiguated by five NER APIs. It provides a user interface for assessing the performance of each of those five tools according to the pattern $(NE, type, URI)$. All user interactions are collected and stored in a database. The framework can finally generate analysis reports and comparison of tools using the NERD ontology[16].

The NERD system architecture is composed of two parts. The back-end is developed in Java and runs on an Apache-Tomcat application server combined with a MySQL database. It has the role to scrape a Web page given as input and to connect the NERD service to all NE extractors. The front-end is developed in HTML/Javascript and has the main role to offer a user interface to the user for assessing the performance of NE tools and visualize analysis reports.

4 Evaluation

We conducted an evaluation of those NLP web services in order to understand their strengths and weaknesses. As we have pointed out, those tools have been developed for different use cases and have different configuration variables. In order to compare their raw results, we develop the NERD ontology, a set of mappings established manually by two ontology engineers between the taxonomy of NE types (section 4.1). We present then the setup of our experiment (section 4.2) before detailing the evaluation results for a controlled (section 4.3) and an uncontrolled (section 4.4) evaluations.

4.1 Preliminaries

One of the differences among these NE extractors is the taxonomy used to represent entity types. DBpedia Spotlight classifies named entities using the DBpedia ontology[17]. In contrast, AlchemyAPI, Extractiv, OpenCalais and Zemanta define their own taxonomies of type, respectively AlchemyAPI schema[18], Extractiv dictionary[19], OpenCalais classes[20] and Zemanta entity types[21]. The most complete taxonomy is provided by AlchemyAPI, which has a very large number of classes similar to DBPedia Spotlight.

We develop the NERD ontology by manually aligning the different classes using their definitions and providing a best coverage of the principal axioms. For the sake of brevity, Table 2 reports only on the `owl:equivalentClass` axioms. However, the NERD ontology provides other mapping axioms using the `rdfs:subClassOf`. The ontology is available at `http://semantics.eurecom.fr/nerd/ontology`.

[16] http://semantics.eurecom.fr/nerd/ontology/

[17] http://dbpedia.org/ontology/

[18] http://www.alchemyapi.com/api/entity/types.html

[19] http://wiki.extractiv.com/w/page/29179775/Entity-Extraction

[20] http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/entity-index-and-definitions

[21] http://developer.zemanta.com/docs/entity_type/

AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
Continent	Continent	CONTINENT	Continent	-
Country	Country	COUNTRY	Country	-
City	City	CITY	City	-
Mountain	-	MOUNTAIN	-	-
Lake	Lake	LAKE	-	-
Company	Company	-	Company	company
Person	Person	PERSON	Person	person
Athlete	Athlete	-	-	-
Politician	Politician	-	-	-
BasketballPlayer	Basketball Player	-	-	-
Movie	Film	MOVIE	Movie	film
Automobile	Automobile	-	-	-

Table 2. `owl:equivalentClass` axioms established manually among the most frequent categories collected in the experiment evaluation.

We use the API documentation pages to identify the entity types that the tools are able to extract and we mint new URI for grouping the axioms definitions. As shown in the listing below, the `nerd:City` class is considered as being equivalent to `alchemy:City`, `dbpedia-owl:City`, `extractiv:CITY` and `opencalais:City` while being more specific than `zemanta:location`.

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

4.2 Method

We conduct two sort of evaluations: *i)* a controlled experiment where 4 participants had to rate the output pattern given by NERD for the same 10 English news articles from 5 different categories selected from BBC and The New York Times; *ii)* an uncontrolled experiment where 17 participants were asked to evaluate a total of 53 English news articles selected randomly from 4 sources: CNN, BBC, The New York Times and Yahoo! News. In both cases, each participant received first a training session consisting in explaining the various functionalities of the tools and the purpose of evaluating the accuracy of the NE detection task, typing and disambiguation[22]. In all experiments, NERD invoked the Linked Data NER extractors using their standard configurations.

The assessment of the output $o = (NE, type, URI)$ of each tool consisted in rating those three criteria with a Boolean value: true (resp. false) if the detected entity was indeed present in the article; true if the assigned type of the named entity is correct in the context of the article; true if the URI provided is an

[22] The application provides a detailed help page

accurate disambiguation of the named entity detected. Furthermore, the users were asked to judge subjectively if the pair $(NE, type)$ was actually relevant for the text being analyzed. In the case where the participant did not assess a result, it would be considered as false. In the case where no type or no disambiguation URI was provided by the tool, it would be considered as false as well.

		1	2	3	4	5	6	7	8	9	10	agreement
AlchemyAPI	NE	0.33	0.12	-0.05	0.08	1	0.13	-0.04	0.03	1	0.19	slight
	Type	0.29	0.47	1	0.22	0.81	0.19	-0.12	0.18	-0.02	-0.08	poor
	URI	0.8	0.68	0.73	0.73	0.83	0.84	1	0.9	0.89	0.7	substantial
	rel	0.34	0.36	0.08	0.07	-0.15	0.28	-0.08	-0.07	1	0.19	slight
Extractiv	NE	0.1	0.05	-0.05	-0.05	0.11	0.09	-0.15	-0.05	0.05	0.1	slight
	Type	0.5	0.65	0.29	0.77	0.37	0.36	0.35	0.54	0.84	0.71	substantial
	URI	0.86	0.65	0.89	0.77	0.77	0.76	0.67	0.81	1	0.85	almost
	rel	0.51	0.2	0.08	0.2	0.32	0.35	-0.1	0.16	0.15	0.17	slight
OpenCalais	NE	0.04	0.05	-0.22	0.05	0.47	0.16	-0.08	0.07	-0.07	0.22	fair
	Type	0.86	0.67	-0.12	0.67	0.64	0.65	-0.11	0.13	-0.03	0.56	moderate
	URI	1	1	1	1	1	1	1	1	1	1	almost
	rel	0.1	-0.06	-0.06	0.2	0.23	0.26	0.25	-0.07	-0.07	0.39	fair
Spotlight	NE	-0.14	-0.07	-0.1	0.02	0.22	0.52	-0.19	0.03	0	-0.11	poor
	Type	0.64	0.84	0.77	0.78	0.82	-0.05	0.3	0.66	0.2	0.29	fair
	URI	0.02	0.16	-0.05	0.25	0.42	0.58	-0.16	0.21	0.67	-0.1	poor
	rel	0.27	-0.05	-0.04	0.04	0.44	0.78	-0.06	0.01	0.04	-0.14	poor
Zemanta	NE	-0.05	0.13	-0.23	-0.05	1	-0.03	-0.18	-0.11	-0.08	0.13	slight
	Type	0.3	0.06	0.23	-0.29	-0.29	-0.33	-0.05	1	0.44	-0.05	poor
	URI	0.01	0.38	0.13	0.16	0.37	-0.05	1	1	-0.08	0.22	fair
	rel	-0.03	0.02	-0.11	-0.03	1	-0.05	-0.07	-0.11	0.01	0.27	fair

Table 3. Fleiss's Kappa score computed for each extractor and per involved fields (NE,Type,URI,relevant). The agreement column shows the Fleiss's Kappa interpretation among all 10 articles.

4.3 Controlled Experiment Results

This experiment consisted of asking 4 participants to evaluate the output pattern given by the analysis of the same 10 news articles, each article being rated 5 times (1 for each NE extractor), yielding a total number of analysis of 200. News articles have been selected from the following categories: world, business, sport, science, health from two sources: BBC and The New York Times. The average word number per article is 981.

Some of the extractors (e.g. DBpedia Spotlight and Extractiv) provide NE duplicates because they compute the NE extraction task for each statement of the text. Instead, the others run the extraction task on the whole text, removing intrinsically the duplicates. In order not to bias the statistics, we first removed all duplicates. The final number of unique entities detected was 4641 with an average number of entity per article equal to 23.2.

Agreement Investigation We compute the Fleiss's Kappa score [6] in order to assess the agreement among the four raters. We interpret this score using the normalization proposed by Sim *et al.*'s classification [16]. Table 3 shows the average agreement for each extractor used in the experiment according to all analysis. Low agreement level is obtained for the NE detection and its relevance for all extractors. Instead, an overall agreement is reached for AlchemyAPI, Extractiv and OpenCalais when users evaluated the Type and URI field. DBpedia Spotlight presents substantial agreement among all raters for the type field, instead low agreement for other fields due, essentially, to the heterogeneous results provided by the extractor (i.e. entity list includes named entities and often topic concepts affecting the overall evaluation). Instead, Zemanta shows an interesting agreement when URI field is evaluated. Table 4 details the agreement score grouped by the source. Table 5 presents the average agreement according to the categories involved in the experiment. Scores are similar for all categories, showing how this experiment reached a good level of agreement for the type and URI evaluation task.

		AlchemyAPI	Extractiv	OpenCalais	Spotlight	Zemanta	agreement
BBC	NE	0.45	0.01	0.03	-0.04	0.09	slight
	Type	0.39	0.47	0.25	0.55	0.13	fair
	URI	0.85	0.84	1	0.18	0.28	substantial
	rel	0.24	0.19	0.09	0.13	0.16	slight
NYTimes	NE	0.11	0.03	0.11	0.08	0.02	slight
	Type	0.19	0.61	0.54	0.5	0.08	fair
	URI	0.77	0.77	1	0.22	0.34	substantial
	rel	0.16	0.21	0.14	0.13	0.02	slight

Table 4. Fleiss's Kappa score computed for each extractor and per involved fields (NE,Type,URI,relevant) grouped by source.

Statistic Results The precision value, p, is computed with the average of the precision for each field of the output triple $o = (NE, type, URI)$. The relevant score, is computed considering the user rating of each pair $(NE, type)$. According to Table 6, AlchemyAPI has the best overall performances both in terms of precision and relevant score. In Table 7, we focus on the detailed precision value of each output o_i. The results are a bit more contrasted. AlchemyAPI, although preserving good performance in NE extraction and accurate typing, has a clear weakness to link the NE to a web resource. URI disambiguation is better performed by Zemanta and DBpedia Spotlight. Moreover, Zemanta has a good reliability to recognize NE in contrast to DBpedia Spotlight. However, both lack the rich type classification. For what concerns DBpedia Spotlight, this result contrasts with the large ontology used to classify the extracted NEs. OpenCalais and Extractiv demonstrate good results in the type identification task.

		AlchemyAPI	Extractiv	OpenCalais	Spotlight	Zemanta	agreement
business	NE	0.01	-0.05	-0.08	-0.04	-0.14	poor
	Type	0.61	0.53	0.28	0.78	-0.03	moderate
	URI	0.73	0.83	1	0.1	0.15	moderate
	rel	0.08	0.14	0.07	0	-0.07	slight
health	NE	0.59	0.07	0.08	-0.06	0.03	slight
	Type	-0.05	0.78	0.27	0.25	0.2	fair
	URI	0.8	0.92	1	0.28	0.07	substantial
	rel	0.59	0.16	0.16	-0.05	0.14	fair
science	NE	0.0	-0.1	-0.01	-0.08	0.14	poor
	Type	0.03	0.44	0.01	0.48	0.47	fair
	URI	0.95	0.74	1	0.02	1	substantial
	rel	-0.08	0.03	0.09	-0.03	-0.09	poor
sport	NE	0.57	0.1	0.32	0.37	0.49	fair
	Type	0.5	0.37	0.64	0.38	-0.31	fair
	URI	0.84	0.77	1	0.5	0.16	substantial
	rel	0.06	0.33	0.24	0.61	0.47	fair
world	NE	0.22	0.08	0.04	-0.11	0.04	slight
	Type	0.38	0.58	0.76	0.74	0.18	moderate
	URI	0.74	0.76	1	0.09	0.19	moderate
	rel	0.35	0.35	0.02	0.11	0	slight

Table 5. Fleiss's Kappa score computed for each extractor and per involved fields (NE,Type,URI,relevant) grouped by article category.

	AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
overall precision	0.7054	0.4915	0.611	0.5396	0.6463
relevant score	0.9005	0.5525	0.6805	0.8224	0.8800

Table 6. Aggregate result comparisons considering the average of the precision and recall for all submitted runs in the controlled experiment.

	p_{name}	p_{type}	p_{uri}
AlchemyAPI	0.9440	0.8938	0.2783
DBPedia Spotlight	0.5995	0.0922	0.7828
Extractiv	0.7713	0.6768	0.3849
OpenCalais	0.8687	0.75	0.0
Zemanta	0.9031	0.1403	0.8954

Table 7. Precision results of NE extraction, type classification and URI selection on all NE extractors evaluated in the controlled experiment.

Configuration parameters affect the general behaviour of these NE extractors. We investigate whether an extractor shows better results or not for a particular genre of text such as news articles. We group all submitted runs according to both authorities: BBC and the New York Times. The results shown that AlchemyAPI has, again, the best performances in terms of NE extraction and type classification (Table 8), but its contribution to select URIs is very low. Previous results showed a good performance of DBpedia in URI disambiguation, but this result is affected by drops with The New York Times articles while showing slight increase for the BBC news articles. Zemanta keeps good performance of URI disambiguation for both authorities.

	BBC			NY Times		
	p_{ne}	p_{type}	p_{uri}	p_{ne}	p_{type}	p_{uri}
AlchemyAPI	0.9676	0.9380	0.4013	0.9235	0.8702	0.2069
Dbpedia Spotlight	0.5955	0.1252	0.7677	0.6197	0.0635	0.0635
Extractiv	0.7674	0.7169	0.3633	0.7891	0.6520	0.3882
OpenCalais	0.9056	0.8755	0.0	0.8340	0.6851	0.0
Zemanta	0.9075	0.1413	0.8938	0.8950	0.1450	0.8950

Table 8. Comparison of NE extraction, type and URI precision among all NE extractors according to the source authority in the controlled experiment.

The alignment proposed in the NERD ontology provides the least common denominator of those NE extractors. Grouping evaluation results under the same classes may provide the ability to assess which ones provide more specific (and precise) results for the most common categories. The alignments we propose enable to make such a comparison. We report on the three most used classes in the task of NE extraction: `Person`, `Organization` and `City` in Table 9. According to the previous analysis, AlchemyAPI has the highest precision value for the name and URI type extraction. Extractiv performs well with the Organization type. Surprisingly, DBpedia Spotlight seems unable to classify NEs for all analyzed types. The value NG is used when the category has not been identified.

	City			Organization			Person		
	p_{ne}	p_{type}	p_{uri}	p_{ne}	p_{type}	p_{uri}	p_{ne}	p_{type}	p_{uri}
AlchemyAPI	1.0	0.9778	0.5111	0.9122	0.8851	0.2568	0.9682	0.9136	0.0772
Dbpedia Spotlight	NG	NG	NG	NG	NG	NG	NG	NG	NG
Extractiv	0.9583	0.9375	0.6458	1	1	1	0.7941	0.7574	0.2132
OpenCalais	0.8958	0.8125	0.0	0.9911	0.9643	0.0	0.95	0.9429	0.0
Zemanta	0.9219	0.3437	0.7969	NG	NG	NG	NG	NG	NG

Table 9. Comparison of NE extraction, type and URI precision among all NE extractors according to the three types `Person`, `Organization` and `City` in the controlled experiment.

4.4 Uncontrolled Experiment Results

To complement the controlled experiment, we also performed a user test scenario where we left raters to free select English news articles from 4 different sources. The 17 users performed 94 runs and each article was assessed by at least 2 different tools. The overall number of entities extracted is 1616, with an average of 34 entities per article. Table 10 presents the overall evaluation for the five tools considered and it shows commonalities according to what showed in the controlled experiment. Table 11 details the item comparison among those extractors. These results are comparable with what obtained in the controlled scenario, expect for the p_{NE} of the DBpedia Spotlight extractor. This is mainly due to the different way to consider a NE for the two set of raters.

	AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
overall precision	0.7415	0.5925	0.6612	0.5402	0.6657
relevant score	0.8916	0.4632	0.6635	0.7006	0.8342

Table 10. Aggregate result comparisons considering the average of the precision and recall for all submitted runs for the uncontrolled experiment.

	p_{ne}	p_{type}	p_{uri}
AlchemyAPI	0.9808	0.9038	0.34
DBPedia Spotlight	0.7448	0.2378	0.7951
Extractiv	0.7357	0.6991	0.5489
OpenCalais	0.8793	0.7414	0.0
Zemanta	0.8760	0.3471	0.7739

Table 11. Precision results of NE extraction, type classification and URI selection on all NE extractors evaluated for the uncontrolled experiment.

We also group ratings according to the authority and the category in this experiment. Table 12 shows the precision of those tools when they extract news articles from the same news article publisher. AlchemyAPI has again the best performance in terms of NE extraction and type classification, but its contribution to select URIs is very small for all of them. Previous results showed a good performance for DBpedia in URI disambiguation (CNN and NY Times articles) while showing lower performance for BBC and Yahoo! News articles. Zemanta keeps good performance of URI disambiguation for all three authorities (CNN, NYTimes and Yahoo! News), but under-performs with BBC data. Finally, Extractiv outperforms OpenCalais only when it works with the CNN dataset in terms of NE extraction, type classification and URI disambiguation. The precision of OpenCalais in terms of name extraction and type classification goes under its average when it works with Yahoo! News.

		BBC	CNN	NYTimes	Yahoo! News
AlchemyAPI	p_{ne}	0.98	0.9818	0.9411	1.0
	p_{type}	0.96	0.9496	0.8824	1.0
	p_{uri}	0.2892	0.3087	0.1176	1.0
DBpedia Spotlight	p_{ne}	0.4296	0.72	0.9314	0.4166
	p_{type}	0.2574	0.32	0.432	0.3333
	p_{uri}	0.5741	0.92	0.8557	0.4166
Extractiv	p_{ne}	0.6108	0.8538	0.5306	0.6967
	p_{type}	0.5398	0.9045	0.4614	0.6311
	p_{uri}	0.3622	0.7793	0.3857	0.4590
OpenCalais	p_{ne}	0.8660	0.8153	0.9167	0.5833
	p_{type}	0.7202	0.6939	0.85	0.5
	p_{uri}	0.0	0.0	0.0	0.0
Zemanta	p_{ne}	0.7	0.6785	0.89	0.84
	p_{type}	0.4	0.2048	0.5	0
	p_{uri}	0.5	0.7476	0.8500	0.90

Table 12. Comparison of NE extraction, type and URI precision among all NE extractors according to the source authority for the uncontrolled experiment.

Table 13 shows the comparison results when NE are grouped according to the NERD ontology. Due to the different news articles corpus used, the results show several changes with respect to what was obtained in the controlled experiment. However, some common aspects emerge: AlchemyAPI preserve its high value performances, except for the URI disambiguation, for all category involved. In contrast, Extractiv shows very precise NE detection for Organization but poor performance for classifying Person and City. Homogeneous results are obtained when OpenCalais is used. DBpedia Spotlight recognized just the Organization NE while Zemanta missed all categories.

5 Discussion

NE extractors are more and more popular within the Semantic Web community with the promise to have a huge impact on the volume of interconnected data. In this context, DBpedia Spotlight and Zemanta provide an important step forward by linking and disambiguating extracted named entities to resources already identified in the LOD cloud. They outperform AlchemyAPI and OpenCalais in this task. Using the complete DBpedia ontology, DBpedia Spotlight may potentially give a precise evaluation of the entity type. Up to now, indeed, when the type is associated, DBpedia Spotlight gives a very deep class hierarchy which helps a computer machinery to structure better the text. The type generation still remains a tricky point for most of them especially for Zemanta. Indeed, it returned empty category type evaluations for each NE for most of our attempts. In this case, NERD used the URI type to classify the NE.

The NE extraction seems a very mature task since most of the NE extracted are meaningful information unit from the text. This extraction, sometimes, gen-

		City	Organization	Person
AlchemyAPI	p_{ne}	1	1	0.9804
	p_{type}	0.8889	0.9545	0.9608
	p_{uri} 0.8889	0.3809	0.2083	
DBpedia Spotlight	p_{ne}	1	1.0	NG
	p_{type}	1	1.0	NG
	p_{uri}	1	1.0	NG
Extractiv	p_{ne}	0.9583	1.0	0.6294
	p_{type}	0.875	1.0	0.7203
	p_{uri}	0.6875	0.5	0.3732
OpenCalais	p_{ne}	0.9	0.9091	0.8378
	p_{type}	0.55	0.7879	0.7838
	p_{uri}	0.0	0.0	0.0
Zemanta	p_{ne}	0.8788	NG	NG
	p_{type}	0.5757	NG	NG
	p_{uri}	0.6452	NG	NG

Table 13. Comparison of NE extraction, type and URI precision among all NE extractors according to the category name.

erates named entity duplicates. Generally, those services tokenize the text in a list of exclusive statements, recognizing the dot as a terminator character of a sentence. In this way, results are affected by multiple named entity occurrences, which are sometimes needed and sometimes not[23]. Co-references (e.g. the pronoun `he`) is also often considered as a named entity. OpenCalais has, instead, a different approach to resolve multiple occurrences, removing them in the extraction results. OpenCalais differs from others also for the URI generation. Indeed, it provide a disambiguation mechanism which links each information unit with web resources in its authority domain and provides occasionally same as link to other LOD datasets.

Finally, this work has evidenced the need for the construction of a common gold standard. Although all raters were trained regarding how a NE detection should be evaluated, they show disagreement in performing this task. Most of them evaluated a NE detection as a true positive when the term was a relevant topic for the article, while others have a stricter definition of a NE. DBpedia Spotlight is an example of extractors which provides a lot a detailed list of concepts rather than NE. E.g., for the article `http://www.bbc.co.uk/news/world-us-canada-14361383`, it extracts topics such as "debt", "economy", etc. that could well index the article but that are not named entities for other raters. This is the reason why, during the controlled experiment, raters did not agree to the identification of the NE.

[23] E.g., `http://omg.yahoo.com/news/jackass-star-died-from-pa-crashs-impact-fire/65597` is about both Jackass the movie, and Jackass the TV Series

6 Conclusion and Future Work

In this paper, we presented an experimental evaluation of human driven named-entity extraction performed by the Named Entity Recognition and Disambiguation (NERD) web application. The evaluation was performed considering precision of Named Entities extraction, precision of the classification of the information unit into categories, precision of the disambiguation of the Named Entity with web resources and the relevant score. Experiment results showed the strengths and weaknesses of five different tools. Overall, AlchemyAPI seems the best solution to extract named entities and to categorize them in a deep ontology. Through the ability to infer data from the LOD cloud, DBpedia Spotlight and Zemanta infer meaningful URIs. Finally, experiments are polarized using the authority as a key selection in the data choice and grouped in similar categories. Our goal was to assess the performance variations according to the type of the extracted named entities and whether NE extractors can provide more specific (and precise) results under the same category or not. Finally, an important research question is addressed: how to evaluate those NE extractors? It becomes crucial to create a sharable “ground truth”, where each NE, type and URI are evaluated in a controlled experiment by human beings. In this work we proposed a first step towards the creation of such a gold standard dataset.

Future work will include the release of a REST API for the NERD framework to the Semantic Web community and to improve the dataset with more user experiences. In terms of manual evaluation, Boolean decision is not enough for judging a NER tool. For example a named entity type might not be wrong, but not precise enough (Obama is not only a person, he is also known as the American President). Another improvement of the system is to allow the input of additional items or correct miss-understanding or ambiguous items. Finally, we plan to implement a “smart” extractor service, which takes into account extraction evaluations coming from all raters to assess new evaluation tasks. The idea is to study the role of the relevant field in order to create a set of not-discovered NE from one tool, but which may be find out by other tools.

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). The authors would also like to thank Xueyi Yang and Mintao Ye from EURECOM for fruitful discussions on the design and development of the NERD interfaces.

References

1. Enrique Alfonseca and Suresh Manandhar. An Unsupervised Method for General Named Entity Recognition And Automated Concept Discovery. In *1^{st} International Conference on General WordNet*, 2002.

2. Masayuki Asahara and Yuji Matsumoto. Japanese Named Entity extraction with redundant morphological analysis. In *International Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*, pages 8–15, Edmonton, Canada, 2003.
3. Daniel Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *5^{th} International Conference on Applied Natural Language Processing*, pages 194–201, Washington, USA, 1997.
4. Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. NYU: Description of the MENE Named Entity System as Used in MUC-7. In *7^{th} Message Understanding Conference (MUC-7)*, 1998.
5. Richard Cyganiak and Anja Jentzsch. Linking Open Data cloud diagram. LOD Community (`http://lod-cloud.net/`), 2010.
6. Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
7. Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: a brief history. In *16^{th} International Conference on Computational linguistics (COLING'96)*, pages 466–471, Copenhagen, Denmark, 1996.
8. Heng Ji and Ralph Grishman. Data selection in semi-supervised learning for name tagging. In *Workshop on Information Extraction Beyond The Document*, pages 48–55, Sydney, Australia, 2006.
9. Andrew McCallumand Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *7^{th} International Conference on Natural Language Learning at HLT-NAACL (CONLL'03)*, pages 188–191, Edmonton, Canada, 2003.
10. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *7^{th} International Conference on Semantic Systems (I-Semantics)*, 2011.
11. David Milne and Ian H. Witten. Learning to link with wikipedia. In *17^{th} ACM Conference on Information and Knowledge Management (CIKM'08)*, pages 509–518, Napa Valley, CA, USA, 2008.
12. David Palmer and David Day. A statistical profile of the Named Entity task. In *5^{th} International Conference on Applied Natural Language Processing*, pages 190–193, Washington, USA, 1997.
13. L.F. Rau. Extracting company names from text. In *7^{th} IEEE Conference on Artificial Intelligence Applications*, volume i, pages 29–32, 1991.
14. Satoshi Sekine. NYU: Description of the Japanese NE system used for MET-2. In *7^{th} Message Understanding Conference (MUC-7*, 1998.
15. Satoshi Sekine and Chikashi Nobata. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *4^{th} International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, 2004.
16. J. Sim and C.C. Wright. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85(3):257–268, January 2005.