Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web

Giuseppe Rizzo, Marieke van Erp and Raphaël Troncy
Università di Torino, Turin, Italy
VU University, Amsterdam, The Netherlands,
EURECOM, Sophia Antipolis, France
giuseppe.rizzo@di.unito.it, marieke.van.erp@vu.nl, raphael.troncy@eurecom.fr

Abstract

Named entity recognition and disambiguation are of primary importance for extracting information and for populating knowledge bases. Detecting and classifying named entities has traditionally been taken on by the natural language processing community, whilst linking of entities to external resources, such as those in DBpedia, has been tackled by the Semantic Web community. As these tasks are treated in different communities, there is as yet no oversight on the performance of these tasks combined. We present an approach that combines the state-of-the art from named entity recognition in the natural language processing domain and named entity linking from the semantic web community. We report on experiments and results to gain more insights into the strengths and limitations of current approaches on these tasks. Our approach relies on the numerous web extractors supported by the NERD framework, which we combine with a machine learning algorithm to optimize recognition and linking of named entities. We test our approach on four standard data sets that are composed of two diverse text types, namely newswire and microposts.

Keywords: Named Entity Recognition, Named Entity Linking, Machine Learning, Newswire, Microposts

1. Introduction

Recognizing named entity mentions in text and linking them to entities on the Web of data is a vital, but not an easy task in information extraction. Since the 90's, recognizing and linking entities has been a popular research topic in the Natural Language Processing (NLP) community. Initially, research focused on identifying and classifying atomic information units in text (Named Entity Recognition and Classification, NER or NERC). Later on, research into linking NEs to external resources, either in dedicated corpora or on the Web, further developed. The primary goal of the Named Entity Linking (NEL) task is to disambiguate the recognized entity with an external definition and description. In this context, the knowledge base that is chosen affects the linking task in several ways, because it provides the final disambiguation point where the information is linked. Recent methods that have become popular in the Semantic Web community utilize knowledge bases such as DBpedia (Auer et al., 2007), Freebase[1] or YAGO (Suchanek et al., 2007) since they contain many entries corresponding to real world entities and are classified according to fine-grained classification schemes.

Many approaches rely on large amounts of manually annotated data to adapt to a different domain and are therefore mostly contained to research environments. Recently, different parties have started offering named entity recognizers and linkers. However, each of these Web extractors may be tuned for a different domain and it is unlikely that a user of these services has data from the exact same domain. Many of these extractors are bundled within the NERD API (Rizzo and Troncy, 2012), providing users the opportunity to easily query each of these services through the same setup and compare their outputs. However, up until now, a comprehensive evaluation of the different extractors was not available. Furthermore, some extractors may be better at some classes within the user's data, while a combination of extractors may yield an overall better performance. The general point is that each system is properly trained for doing a specific task and in a specific context. Combining different views can give a better and more exhaustive view of the overall annotation.

In this paper, we present an evaluation of both the NER and NEL tasks of the different extractors bundled in the NERD API, and we compare them with NERD-ML (van Erp et al., 2013), a system that learns how to combine the output of the different extractors. We present results for two domains: newswire and microposts. Our experiments show that the NERD extractors, taken individually, do not yield satisfactory results, while combined in NERD-ML, they can recognize and link named entities in both newswire and microposts reasonably well. However, this combination can only be obtained using machine learning algorithms tuned to a dataset. We also compute a theoretical upper bound limit that corresponds to an ideal combination of all extractors and we show that those results would come close to recognizing all entities in the domains.

The reminder of this paper is organized as follows: in Section 2., we further describe our motivation and some prior work. We present our named entity recognition experiments in Section 3., and our named entity linking experiments in Section 4.. Our results and their implications for further work in NER and NEL are detailed in Section 5.. We conclude and outline some future work in Section 6.. We provide all the necessary resources in order to replicate our experiments with this paper at `https://github.com/giusepperizzo/nerdml`.

2. Background and Motivation

The NER and NEL tasks have been addressed in different research fields such as the NLP, Web mining and Semantic Web communities. One of the first research papers in the NLP field, aiming at automatically identifying named

[1] `http://www.freebase.com`

entities in texts, was proposed by Rau (Rau, 1991). 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.* (Sekine and Nobata, 2004). A different approach was introduced when Supervised Learning (SL) techniques were used. The big change was the use of a large manually labeled data set. In the SL field, a human being usually annotates positive and negative examples so that the algorithm computes classification patterns. SL techniques exploit Hidden Markov Models (HMM) (Bikel et al., 1997), Decision Trees (Sekine, 1998), Maximum Entropy Models (Borthwick et al., 1998), Support Vector Machines (SVM) (Asahara and Matsumoto, 2003) and Conditional Random Fields (CRF) (Li, 2003). 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) and Unsupervised Learning (UL) approaches attempt to solve this problem by either providing a small initial set of labeled data to train and seed the system, or by resolving the extraction problem as a clustering one (Nadeau and Sekine, 2007). For instance, a user 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 (Alfonseca and Manandhar, 2002).

Scientific evaluation campaigns focused on newswire content, starting with MUC (Sundheim, 1993), CoNLL (Tjong Kim Sang, 2002; Tjong Kim Sang and Meulder, 2003) and ACE (NIST, 2005). They were proposed to compare the performance of various systems in a rigorous and reproducible manner. While an increasing popularity of NER focused on newswire content, relatively little previous work on building similar systems tuned for microposts such as tweets or similar text styles have been done. Hence, there are only relatively few recent shared tasks, such as the MSM 2013 and Microposts 2014 challenges, for evaluating the performance of various systems while annotating microposts in a rigorous and reproducible manner. Microposts are gaining more attention in the research domain as they are a highly popular source for opinions. As such, they promise great potential for researchers and companies alike to tap into a vast wealth of a heterogeneous and instantaneous barometer of what is currently trending in the world. However, due to their brief and fleeting nature, microposts provide a challenging playground for text analysis tools that are oftentimes tuned to longer and more stable texts. A first attempt was detailed in (Locke, 2009), in which the authors train a classifier with an annotated Twitter corpus to detect named entities of types `Person`, `Location` and `Organization`. The classifier performance was in average among the four classes of 40%.

The performance of a classifier is often proportional to the size of the training corpus. Therefore, several research efforts have been made for creating a wealth annotated tweet corpus, using Amazon Mechanical Turk (Finin et al., 2010). Beside the use of CRF in this task, a semi-supervised approach using k-Nearest Neighbor (k-NN) is proposed in (Liu et al., 2011). Generally, the linguistic variation in a micropost stream such as Twitter, with respect to a normal newswire corpus, tends to negatively affect the performance of state of the art NER systems. To reduce this problem, a system based on a set of features extracted from Twitter-specific POS taggers, a dedicated shallow parsing logic, and the use of Gazetteers generated from Freebase entities, that match best the fleeting nature of that informal messages is proposed in (Ritter et al., 2011).

The NER task is strongly dependent on the knowledge used to train the NE extraction algorithm. Recent methods, coming from the Semantic Web community, have been introduced to map entities to relational facts exploiting fine-grained ontologies from sources such as DBpedia, Freebase and YAGO. In addition to extracting and classifying a NE, efforts have been spent to develop methods for linking such an information to external resources. Disambiguation is one of the key challenges in this scenario and is founded on the fact that terms taken in isolation are naturally ambiguous: 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 methods generally try to find some clues in the surrounding text for contextualizing the ambiguous term and refine its intended meaning. Initially, the Web mining community used Wikipedia as the linking hub where entities were mapped (Milne and Witten, 2008; Kulkarni et al., 2009; Hoffart et al., 2011). A natural evolution of this approach resulted in disambiguating named entities with data from the LOD cloud. Mendes et al. (2011) proposed an approach to avoid named entity ambiguity using the DBpedia data set. Given the early stage of the research on microposts, a few attempts have been proposed to establish in-house gold standard corpora for microposts, and recently, it has been proposed a share task on benchmarking linking systems that deal specifically with microposts (Basave et al., 2014). Meij et al. (2012) created an in-house test collection of tweets linked to Wikipedia articles and they proposed a mixture of N-Gram and features extraction tuned to a Twitter stream. Disambiguating tweets using Wikipedia articles is also proposed by Guo et al. (2013), where the linking is done on a per-tweet basis. The authors proposed an evaluation based on a sampled Ritter's data set, where 473 tweets have been used for the experiment, showing that a bottleneck for the NEL task is due to the poor performance of the NER task in the first place.

3. Named Entity Recognition Experiments

In the named entity recognition experiments, we focus on identifying named entities and their types in newswire and microposts texts.

3.1. Data sets

The two use cases, newswire and microposts, were selected because of their diversity and current standing in the NER and NEL community. Newswire has been at the center of attention in named entity recognition since the start of NER

research as newswire articles are generally easy to obtain and contain a variety of different topics (Sundheim, 1993). Microposts have a much shorter history, but as they are a highly popular medium to share facts, opinions or emotions, they promise great potential for researchers and companies alike to tap into a vast wealth of a heterogeneous and instantaneous barometer of what is currently trending in the world. In this section, we describe our use cases and data sets. Both data sets are marked up with four types of named entities: persons, locations, organizations and miscellaneous. There is a slight difference in the definition of the miscellaneous category in the two domains. For the newswire data, this category includes songs, slogans, nationalities and languages[2] for a complete overview). In the microposts data set, languages are not included in the miscellaneous class. We will explain how this may affect classification in Section 5.. We describe below the characteristics of each data set.

3.1.1. Newswire (CoNLL-2003 Reuters corpus)

One of the most prominent data set in NER is the corpus that was created for the CoNLL-2003 Language-Independent Named Entity Recognition shared task(Tjong Kim Sang and Meulder, 2003). For this task, English and German news articles were annotated with named entities and made available to the research community to train and test their systems on. The English training data consists of 946 articles, containing $203,621$ tokens. The test data consists of 231 articles, containing $46,435$ tokens. The data was marked up manually with named entities of types person, location, organization and miscellaneous. Part-of-speech and chunk tags were added automatically. There is fairly little overlap of named entities between the training and test data sets: only 2.9% of the named entities that occur in the training data also occur in the test data.

	Articles	Tokens	NEs	PER	LOC	ORG	MISC
Training	946	203,621	23,499	6,600	7,140	6,321	3,438
Testing	231	46,435	5,648	1,617	1,668	1,661	702

Table 1: Statistics on number of articles, tokens, named entities for the Reuters data set.

3.1.2. Microposts (MSM'13 corpus)

The MSM'13 corpus was created for the Making Sense of Microposts Challenge 2013 (Basave et al., 2013) and consists of microposts collected from the end of 2010 to the beginning of 2011. The training set contains $2,815$ tweets, totalling $51,521$ tokens, and the test set contains $1,450$ tweets, totalling $29,085$ tokens. To anonymize the posts, username mentions are replaced with "_Mention_" and URLs with "_URL_". There is a fair bit more overlap of named entities between the training and test data: 8.1% of the named entities from the training data also occur in the test data.

For ease of use with our system, we converted the original tab-separated format consisting of tweet id, followed by entities contained in the tweet and the tweet text to the CoNLL IOB format (one token per line). The tweet id was preserved as an _ENDOFTWEET_<id> token.

[2]See the CoNLL annotation guidelines `http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt`

	Posts	Tokens	NEs	PER	LOC	ORG	MISC
Training	2,815	51,521	3,146	1,713	610	221	602
Testing	1,450	29,085	1,538	1,116	97	233	92

Table 2: Statistics on number of posts, tokens, named entities for the MSM data set.

NERD classes	CoNLL-2003	MSM'13
nerd:Person	PER	PER
nerd:Organization	ORG	ORG
nerd:Location	LOC	LOC
nerd:Amount	O	O
nerd:Animal	O	O
nerd:Event	MISC	MISC
nerd:Function	MISC	O
nerd:Product	MISC	O
nerd:Time	MISC	O
nerd:Thing	O	O
nerd:Movies	-	MISC
nerd:ProgrammingLanguage	-	MISC

Table 3: Entity classification alignment between NERD, CoNLL-2003 and MSM'13.

3.2. Experimental Setup

We followed 5 steps: *i)* collect NERD NE tags, *ii)* add extra linguistic features, *iii)* add NE tags from other NER approaches, *iv)* train classifier on training set, *v)* create classifiers.

i) As the output of the individual NERD extractors forms the heart of the features used by the machine learner, we first sent both training and test data sets to the NERD API to retrieve named entities from the NERD extractors. Due to file size limitations, we split up the Reuters data sets into parts of no larger than 50KB and sent each part separately to the NERD API. We made sure that the documents or microposts did not get split up. The extractors that were queried for the NER task are the following: AlchemyAPI, DBpedia Spotlight v0.6(setting: *confidence=0*, *support=0*, *spotter=CoOccurrenceBasedSelector*), Cicero[3], Lupedia, OpenCalais, Saplo, TextRazor, Wikimeta, and Yahoo!. Each extractor classifies entities according to its own schema which was harmonized to the NERD ontology v0.5[4]. We mapped the retrieved types to the four classes in the data sets (Table 3). As in any alignment task, the mapping from the NERD ontology and the four classes in the shared tasks is imperfect, as it narrows down the classification of the extractors from a fine-grained to a coarse-grained classification. The alignment is also error-prone due to the lack of documentation from some extractors. Resolving these issues is a topic for future work.

ii) In the Reuters data set, part-of-speech (POS) and chunk information is already present. For the microposts data, POS tags were added using the TwitterNLP tool (Owoputi et al., 2013). Additionally, we added 7 different features inspired by the features described in (Ritter et al., 2011): capitalization information (initial capital, allcaps, proportion of capitalised tokens in the micropost or proportion of tokens in the sentence of the news article), prefix (first three letters of the token), suffix (last three letters of the token),

[3]Cicero is the new name of the service previously known as Extractiv.

[4]`http://nerd.eurecom.fr/ontology/nerd-v0.5.n3`

and whether the token is at the beginning or end of the micropost or sentence of the news article.
iii) To evaluate the NERD extractors against an off-the-shelf NER system, the microposts were tagged by the system described in (Ritter et al., 2011). The 10 entity classes are mapped to the four classes in our data set. We also retrained the Stanford NER system on the MSM'13 data challenge training set, using parameters based on the `english.conll.4class.distsim.crf.ser.gz` properties file provided with the Stanford distribution. The newswire domain was tagged with the Stanford NER system (version 3.2.0) with the prepackaged `english.conll.4class.distsim.crf.ser.gz` model. The outputs from these systems were also used as features in the hybrid NERD-ML system.
iv) The output generated by the NERD extractors and the added features and tags from other NE systems are used to create feature vectors for the machine learning algorithm to find combinations of features and extractor outputs that improve the scores of the individual extractors. We experimented with several different algorithms and machine learning settings using WEKA-3.7.9[5].
v) For both data sets, we apply three different machine learning algorithms: Naive Bayes (NB), *k*-Nearest Neighbor (*k*-NN) and Support Vector Machines (SVM).

3.3. Feature Vector

As input of the classification models, for both training and testing, we attach to each phrase a feature vector that is depicted in Figure 1. A vector of linguistic features is generated for each input, together with the extractor types assigned to the phrase. The number of extractor types may vary, as described above. The feature vector is ended by the type assigned by the gold standard. These feature are literals.

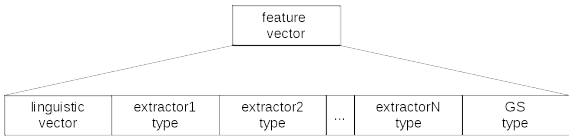

Figure 1: Feature Vector.

The linguistic vector (Figure 2) is composed of seven features. All of them are literals, expect for those labeled by (*) that are Boolean, and (**) that are double.

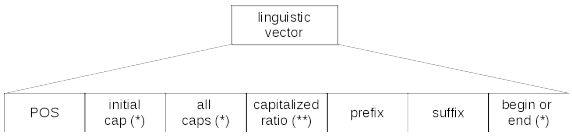

Figure 2: Linguistic Vector.

3.4. Theoretical limit

The task of the classifier is to decide whether a token belongs to a particular NE class. We have defined the concept of theoretical limit. This represents the optimal limit that NERD-ML can achieve if the combination of all the extractors is optimal. In other words, the theoretical limit is computed considering a well-classified token if at least one extractor has classified it properly.

3.5. Evaluation

Results are computed using the conlleval script[6] and plotted using R. Figure 3 shows the results of the individual NER extractors and a selection of the hybrid NERD-ML systems over the CoNLL-2003 Reuters corpus. The settings of the three runs of the hybrid NERD-ML system are:

NERD-ML-NB token, pos, initialcaps, allcaps, prefix, suffix, capitalfreq, start, AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, OpenCalais, Saplo, Yahoo!, Textrazor, Wikimeta, Stanford, class; ML=NB.

NERD-ML-kNN token, AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, OpenCalais, Saplo, Yahoo!, Textrazor, Wikimeta, Stanford, class; ML=*k*-NN, k=1, Euclidean distance.

NERD-ML-SVM AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, OpenCalais, Saplo, Textrazor, Wikimeta, Stanford, class; ML=SMO, polynomial kernel, standard parameters.

Figure 4 reports the results of the individual NER extractors and a selection of the hybrid NERD-ML systems over the MSM'13 corpus. The settings of the three runs of the hybrid NERD-ML system are:

NERD-ML-SVM_1 token, AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, OpenCalais, Saplo, Yahoo!, Textrazor, Wikimeta, Ritter, Stanford, class; ML=SMO, polynomial kernel, standard parameters.

NERD-ML-SVM_2 token, pos, initialcaps, suffix, proportion of capitals, AlchemyAPI, DBpedia Spotlight, Cicero, Opencalais, Textrazor, Wikimeta, Ritter, Stanford, class; ML=SMO, polynomial kernel, standard parameters.

NERD-ML-NB token, pos, initialcaps, allcaps, prefix, suffix, capitalfreq, start, AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, Opencalais, Textrazor, Ritter, Stanford, class. ML=NB.

3.6. Result Analysis

Results show the best settings achieved by the ML learning techniques used. With the CoNLL-2003 Reuters corpus, SVM outperforms both *k*-NN and Naive Bayes for classifying entities within the MISC class. *k*-NN and Naive Bayes work both well for recognizing PER and *k*-NN shows its strengths on detecting LOC. We can also observe a steady high recall when using Naive Bayes.
With the MSM'13 corpus, Naive Bayes consistently outperforms the others in terms of recall. Its reliability on detecting many entities affects however the general precision and F-measure. This result is also interesting in terms

[5] `http://www.cs.waikato.ac.nz/ml/weka`

[6] `http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt`

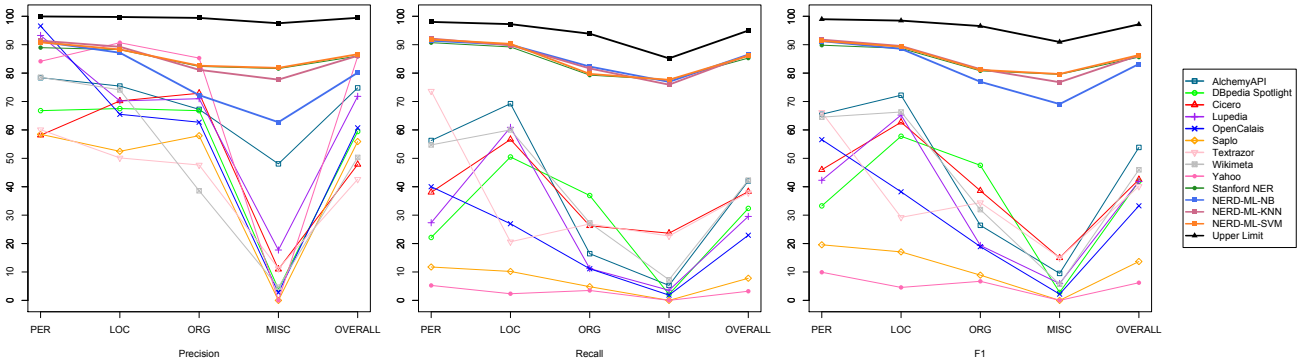

Figure 3: Precision, Recall and F-measure results for individual NERD extractors, Stanford and NERD-ML on CoNLL-2003 Reuters data set for different classes and overall. The black line denotes the upper limit the combined extractors can obtain.

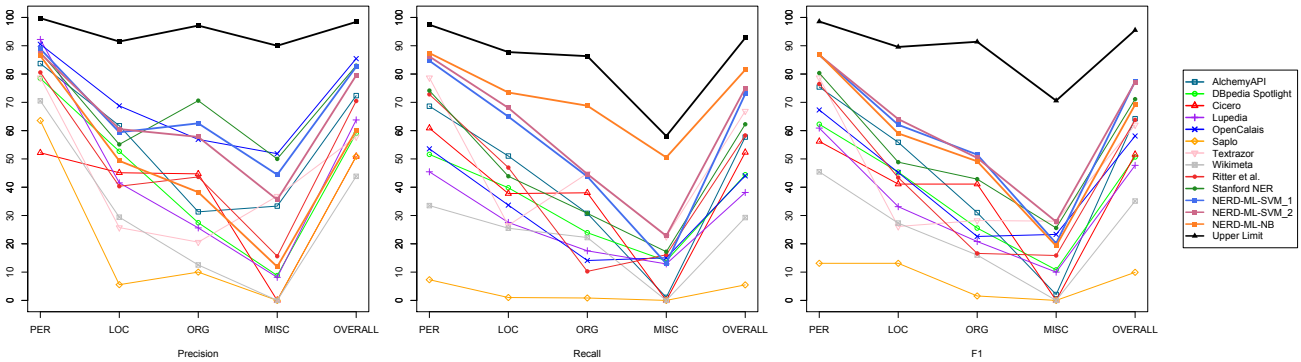

Figure 4: Precision, Recall and F-measure results for individual NERD extractors, Stanford and NERD-ML on MSM2013 data set for different classes and overall. The black line denotes the upper limit the combined extractors can obtain.

of computation since Naive Bayes generates a model that is faster and lighter to run than all the other investigated ML algorithms (in terms of time and memory). The two SVM settings preserve their strengths in terms of recall and the setting with all features outperforms the others. Naive Bayes seems to work better in this context of high noise and uncertain token distribution.

A further investigation has been conducted for evaluating how the size of the training set can affect the final results of the classifier. For doing this experiment, we selected the following settings:

CoNLL-2003 Reuters: token, AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, OpenCalais, Saplo, Yahoo!, Textrazor, Wikimeta, Stanford, class; ML= NB.

MSM'13: token, pos, initialcaps, allcaps, prefix, suffix, capitalfreq, start, AlchemyAPI, DBpedia Spotlight, Cicero, Lupedia, Opencalais, Textrazor, Ritter, Stanford, class. ML=SVM.

To gain some further insights into the influence of the amount of training data used, we carried out a series of experiments in which we incrementally built up the size of the training data in 10 steps. Figures 5 and 6 show the F-measures on the Reuters and MSM corpora averaged over 5 series of experiments in which we randomly added new data segments. The error bars show the variation in minimum and maximum scores achieved per step. We can observe that the performance of the classifier on the Reuters corpus improves by adding training data, which is what is to be expected if a training and test set are well balanced, and the features are chosen well.

For the MSM corpus, however, the reverse holds. Here, the classifier tends to worsen with an increase in training data. This may be explained by the fact that the data set is quite small, and there is also a fair bit overlap of named entities between the training and test sets, which may cause overfitting. Further research into the exact influence of data set size for this corpus are necessary, but as this would require a larger data set this is out of the scope of this contribution.

4. Named Entity Linking Experiments

In the named entity linking experiments, we focus on identifying links which disambiguate named entities to encyclopedic resources described in DBpedia.

4.1. Data sets

The two use cases, newswire and microposts, were selected to benchmark systems that do linking. We respectively used AIDA-YAGO2 (Hoffart et al., 2011) and #Microposts2014 Named Entity Extraction and Linking (NEEL) Challenge (#Microposts'14) (Basave et al., 2014) data sets.

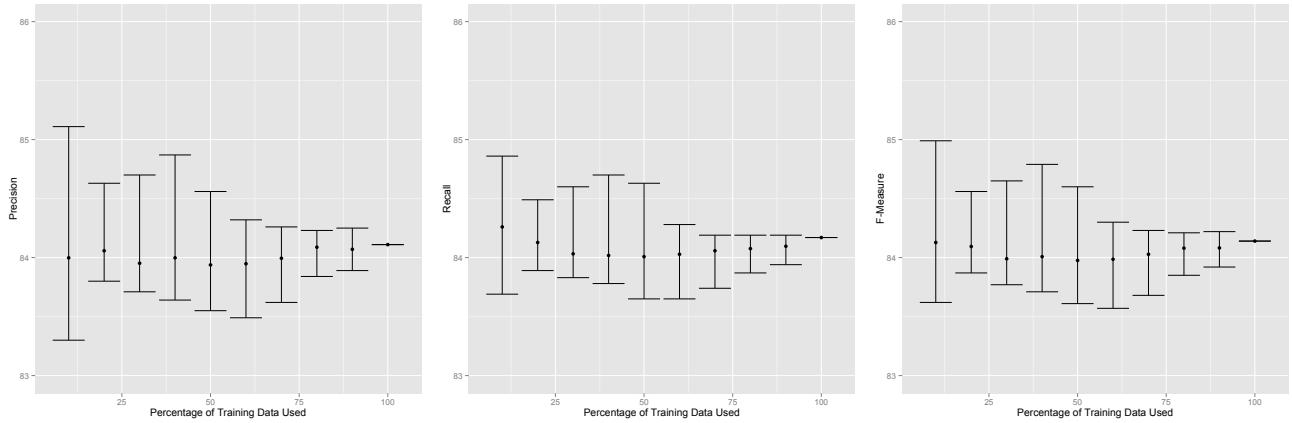

Figure 5: Learning curve on CoNLL-2003 Reuters data set.

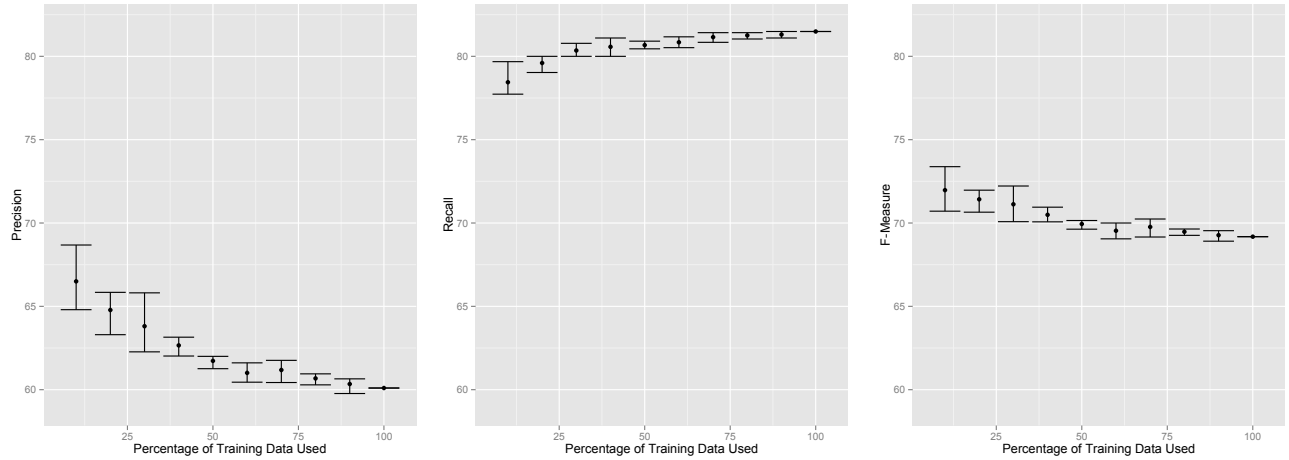

Figure 6: Learning curve on MSM'13 data set.

4.1.1. Newswire (AIDA-YAGO2 corpus)

Hoffart et al. (2011) added links to YAGO and Wikipedia to each entity mention in the CoNLL-2003 corpus. Table 1 reports the statistics of the data set.

4.1.2. Microposts (#Microposts'14 corpus)

The data set contains 3.5K tweets that cover event-annotated tweets collected for the period of July 15th, 2011 to August 15th, 2011. It extends over multiple noteworthy events including the death of Amy Winehouse, the London Riots and the Oslo bombing.

	Posts	Tokens	NEs	Links
Training	2,340	48,769	3,820	3,820
Testing	1,165	23,815	1,330	1,330

Table 4: Statistics on number of posts, tokens, named entities, and links for the #Microposts'14 data set.

4.2. Experimental Setup

We followed 3 steps: *i)* collect NERD NE links as output of the named entity extraction process for each document, *ii)* filter out the links which do not point to the two encyclopedic knowledge bases taken into account in this study, namely Wikipedia and DBpedia, and *iii)* map the Wikipedia links to their corresponding DBpedia resources.

i) We first split the data sets in, respectively, a set of articles (AIDA-YAGO2) and a set of tweets (#Microposts2014). We send each document or each tweet to the NERD API from which we collected the disambiguation URIs. The extractors that were queried are: AlchemyAPI, dataTXT(setting:*min_confidence=0.6*), DBpedia Spotlight v0.6(setting: *confidence=0*, *support=0*, *spotter=CoOccurrenceBasedSelector*), Lupedia, TextRazor, THD, Yahoo! and Zemanta.

ii) Not all these extractors use the same knowledge base to disambiguate the entities being recognized. Zemanta and AlchemyAPI point to open Web resources including Wikipedia and DBpedia resources. TextRazor disambiguates entities using Wikipedia resources while the others use DBpedia as the knowledge base. Consequently, comparing the ability of these extractors to correctly disambiguate a named entity is challenging and ambiguous.

iii) In our investigation, we consider a valid disambiguation resource if it points to the English Wikipedia/DBpedia knowledge base.

4.3. Evaluation

Results are computed using the neeleval script[7] and plotted using R. The evaluation is based on micro-average analysis. Table 5 reports the results of the individual NEL extractors over the AIDA-YAGO2 corpus, and Table 6 shows the results of the individual NEL extractors over the #Microposts'14 corpus.

	AlchemyAPI	*dataTXT*	*DBpedia*	*Lupedia*	*Textrazor*	*THD*	*Yahoo!*	*Zemanta*
p	**70.63**	39.20	26.93	57.98	49.21	32.50	61.24	35.58
r	14.05	**54.93**	42.21	29.90	51.66	40.10	9.65	7.78
f	23.43	45.75	32.88	39.45	**50.41**	35.90	16.68	12.77

Table 5: Breakdown per extractor regarding the NEL task over the AIDA-YAGO2 data set.

	AlchemyAPI	*dataTXT*	*DBpedia*	*Lupedia*	*Textrazor*	*THD*	*Yahoo!*	*Zemanta*
p	**72.22**	22.11	13.99	37.37	30.69	23.54	60.68	35.54
r	3.91	34.74	29.70	11.13	**34.89**	23.98	10.68	10.08
f	7.42	27.02	19.02	17.15	**32.65**	23.76	18.16	15.70

Table 6: Breakdown per extractor regarding the NEL task over the #Microposts2014 corpus.

4.4. Result Analysis

Results show strengths and weaknesses of these linkers depending on the corpus. AlchemyAPI has generally the best precision, while dataTXT and TextRazor have the best recall when linking named entities to the normalized DBpedia knowledge base for respectively the newswire and the microposts corpora. Overall, TextRazor is the one which shows the most stable and solid performance on both data sets when looking at the f-measure.

5. Discussion

The NER experiments presented in Section 3. highlight already the diversity of the domains that could be investigated in NER, analyzing search logs being now a natural evolution[8]. Even within the domains we present, the data sets are limited in size and coverage. In particular, the MSM'13 data set poses some serious limitations in terms of size and suffers from a fair overlap between the training and test data sets. This is probably due to the fact that the tweets for this data set span a very short period of time. Still, our experiments show that the combination of the different NERD extractors with a machine learner improves the performance of the individual extractors and manages to obtain very reasonable scores on the NER task.

As the results in both Figures 3 and 4 show, the miscellaneous class is the most difficult to recognize for both data sets. This is largely due to the fact that there is no consensus on what this class exactly entails. Although the MSM'13 annotation guidelines are based on the CoNLL guidelines, there is a difference in coverage. Also, different NERD classes needed to be mapped to this class (see Table 3) and some NERD extractors do not even recognize entities that do not fall within the person, location and organization classes (such as Cicero, Saplo, Wikimeta, and Yahoo!). Not having consensus across the different extractors about a particular class significantly increases the difficulty of recognizing this class.

Since the NEL challenge is still a recent task, there is yet no common agreement on the annotation level to adopt. In our setting, we used a per-tweet level annotation. The evaluation is based on micro-average, evaluating whether the pair, composed of an entity mention and a link, matches the one in the gold standard. Due to all the different settings provided by the NERD extractors, the results we obtained give just a raw idea of the task, lacking from the needed formalization already achieved in the NER task. This provides ample research avenues for future work.

6. Conclusion and Future Work

In this paper, we presented a thorough study of state-of-the-art NER and NEL systems on newswire and micropost content. We presented experiments and results that combine the state-of-the art from named entity recognition in the NLP domain and named entity linking from the Semantic Web community to gain more insights into the strengths and limitations of current approaches on these tasks. We introduced the concept of theoretical upper bound limit and we presented NERD-ML, an approach that unifies the benefits of a crowd entity recognizer through Web entity extractors combined with the linguistic strengths of a machine learning classifier. Our experiments show that by using NERD-ML, we outperform the state of the art in the NER task and we introduced a first harmonization of the NEL task.

Part of the ongoing work is to improve the NER results to get closer to our theoretical limit. In our incremental data set experiments, we observed how the size of the training corpus can influence the performance of a classifier. We plan to discover more insights from these results, digging more in the distribution of tokens and entities of the data sets used. Due to its early stage, the NEL task is the part to which we will dedicate more efforts, both in terms of harmonizing the results from the Web entity extractors and for performing a thorough evaluation over other corpora. The TAC KBP task dedicated to entity linking will also provide a good setting to experiment with NERD-ML.

Acknowledgements

The research leading to this paper was partially supported by the European Union's 7th Framework Programme via the projects LinkedTV (GA 287911) and NewsReader (ICT-316404).

7. References

Alfonseca, E. and Manandhar, S. (2002). An Unsupervised Method for General Named Entity Recognition And Au-

[7] https://github.com/giusepperizzo/neeleval

[8] ERD challenge: http://web-ngram.research.microsoft.com/erd2014/

tomated Concept Discovery. In *1st International Conference on General WordNet*.

Asahara, M. and Matsumoto, Y. (2003). Japanese Named Entity extraction with redundant morphological analysis. In *North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In *6th International Semantic Web Conference (ISWC'07)*.

Basave, A. E. C., Varga, A., Rowe, M., Stankovic, M., and Dadzie, A.-S. (2013). Making Sense of Microposts (#MSM2013) Concept Extraction Challenge. In *3rd Workshop on Making Sense of Microposts (#MSM2013)*.

Basave, A. E. C., Rizzo, G., Varga, A., Rowe, M., Stankovic, M., and Dadzie, A.-S. (2014). Making Sense of Microposts (#Microposts2014) Named Entity Extraction & Linking Challenge. In *4th Workshop on Making Sense of Microposts (#Microposts2014)*.

Bikel, D., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *5th International Conference on Applied Natural Language Processing*.

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). NYU: Description of the MENE Named Entity System as Used in MUC-7. In *7th Message Understanding Conference (MUC-7)*.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in Twitter data with crowdsourcing. In *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Guo, S., Chang, M.-W., and Kiciman, E. (2013). To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'13)*.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust Disambiguation of Named Entities in Text. In *Empirical Methods in Natural Language Processing (EMNLP'11)*.

Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (2009). Collective annotation of Wikipedia entities in Web text. In *15th ACM International Conference on Knowledge Discovery and Data Mining (KDD'09)*.

Li, A. M. W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *7th International Conference on Natural Language Learning at HLT-NAACL (CoNLL'03)*.

Liu, X., Zhang, S., Wei, F., and Zhou, M. (2011). Recognizing Named Entities in Tweets. In *Annual Meeting of the Association for Computer Linguistics (ACL'11)*.

Locke, B. (2009). Named Entity Recognition: Adapting to Microblogging. Master's thesis, University of Colorado.

Meij, E., Weerkamp, W., and de Rijke, M. (2012). Adding semantics to microblog posts. In *5th ACM Conference on Web Search and Data Mining (WSDM'12)*.

Mendes, P. N., Jakob, M., Garcia-Silva, A., and Bizer, C. (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. In *7th International Conference on Semantic Systems (I-Semantics'11)*.

Milne, D. and Witten, I. H. (2008). Learning to link with Wikipedia. In *17th ACM Conference on Information and Knowledge Management (CIKM'08)*.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

NIST. (2005). The ACE 2005 (ACE05) Evaluation Plan.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'13)*.

Rau, L. (1991). Extracting company names from text. In *7th IEEE Conference on Artificial Intelligence Applications*.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. In *Empirical Methods in Natural Language Processing (EMNLP'11)*.

Rizzo, G. and Troncy, R. (2012). NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In *13th Conference of the European Chapter of the Association for computational Linguistics (EACL'12)*.

Sekine, S. and Nobata, C. (2004). Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *4th International Conference on Language Resources and Evaluation (LREC'04)*.

Sekine, S. (1998). NYU: Description of the Japanese NE system used for MET-2. In *7th Message Understanding Conference (MUC-7)*.

Suchanek, F., Kasneci, G., and Weikum, G. (2007). Yago: a Core of Semantic Knowledge. In *16th International Conference on World Wide Web (WWW'07)*.

Sundheim, B. M. (1993). Overview of results of the MUC-6 evaluation. In *6th Conference on Message Understanding (MUC)*.

Tjong Kim Sang, E. F. and Meulder, F. D. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *17th Conference on Computational Natural Language Learning (CoNLL'03)*, Edmonton, Canada.

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *16th Conference on Computational Natural Language Learning (CoNLL'02)*, Taipei, Taiwan.

van Erp, M., Rizzo, G., and Troncy, R. (2013). Learning with the Web: Spotting Named Entities on the intersection of NERD and Machine Learning. In *3rd Workshop on Making Sense of Microposts (#MSM2013)*.