Inductive Entity Typing Alignment

Giuseppe Rizzo[1,2], Marieke van Erp[3], Raphaël Troncy[1]

[1] EURECOM, Sophia Antipolis, France
giuseppe.rizzo@eurecom.fr, raphael.troncy@eurecom.fr
[2] Università di Torino, Turin, Italy
[3] VU University Amsterdam, The Netherlands
marieke.van.erp@vu.nl

Abstract. Aligning named entity taxonomies for comparing or combining different named entity extraction systems is a difficult task. Often taxonomies are mapped manually onto each other or onto a standardized ontology but at the loss of subtleties between different class extensions and domain specific uses of the taxonomy. In this paper, we present an approach and experiments for learning customized taxonomy alignments between different entity extractors for different domains. Our inductive data-driven approach recasts the alignment problem as a classification problem. We present experiments on two named entity recognition benchmark datasets, namely the CoNLL2003 newswire dataset and the MSM2013 microposts dataset. Our results show that the automatically induced mappings outperform manual alignments and are agnostic to changes in the extractor taxonomies, implying that alignments are highly contextual.

1 Introduction

Named recognition and classification is an important task in providing more fine-grained access to textual resources than simple keyword search can offer. In recent years, many named entity recognition and classification tools have become available such as DBpedia Spotlight [8] and OpenCalais[4]. Each of these tools has a slightly different goal and different inner workings. Often, the entity schemas that these systems use internally are different, requiring prior alignment of the schemas in order to compare these systems. In previous work, we have manually mapped the taxonomies of 12 of these systems to a single ontology, namely the NERD ontology [13]. However, as these taxonomies evolve over time, mappings may need to be updated, which is an iterative and time consuming task. Furthermore, a single, static mapping to another taxonomy may result in loss of subtleties between different class extensions and domain specific uses of the taxonomy.

In this paper we show that it is possible to inductively learn mappings of entity types between various extractors available in the NERD framework and gold standard benchmark classes for well-defined entity classes such as person,

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

organization and location. Bin-classes such as 'miscellaneous' are more difficult to learn, but inspection of our mappings shows that the extractors uncover inconsistencies in the gold standard datasets that are being used. To assess the feasibility of the inductive approach, we use the learned mappings as input of the NER experiments reported in [12], and we observe improvements with respect to the baseline (computed using the manual mappings). The increase in performance is dependent on the used dataset, showing that this approach is better performing with the MSM2013 one.

The proposed approach enables us to create general conclusions based on the observation of individual cases. This is what is observed in the domain of the Natural Language Processing (NLP), in particular for the entity recognition, where the taxonomy is generally encoded implicitly in the data. The learning algorithms, after observing the distributions of tokens and types, train the classifier. Quite recently, with the introduction of the entity extractors, that together with the entity recognition also perform entity linking, the problem of using a rich upper level schema (the majority as advances proposed in the Linked Data movement) of the data has been exploited. Nowadays, the DBpedia Ontology, Freebase, and Schema.org (to name few) are schemas largely used by a plethora of commercial and research entity extractors. Most of these extractors can be used as off-the-shelf extractors, hence there is no chance to feed in a data schema that is different from the one internally modeled.

The remainder of this paper is organized as follows. In Section 2, we describe background and related work. In Section 3, we describe the datasets, the set of extractors together with their settings, and the data processing stage. In Section 4, we statistically motivate our approach and we provide two complementary approaches for learning inductively the alignments. In Section 5, we present the experiments and results, followed by a discussion in Section 6. We finish with conclusions and pointers for future work in Section 7.

2 Background and Related work

Matching different schemas for generating correspondences between elements is an extensively explored task. Schema matching is a critical step in many domains such as e-business, data warehouses and databases [3]. With the advent of the Semantic Web, schema matching has taken a central role in managing highly structured knowledge bases, such as DBpedia and YAGO. Several matching tools have been evaluated but typically under different conditions and for smaller match problems [2]. The OAEI yearly organizes shared tasks which include large ontologies, such as medical and library schemas. All these schemas come with a host of additional metadata, that is generally exploited by the matching tools. For instance, Cupid [7] combines a number of techniques such as linguistic matching, structure-based matching, constraint-based matching, and context-based matching at the schema element level and related metadata. A peculiarity of our work is its aim to be resilient to the schemas' heterogeneity, in terms of number of classes, number of hierarchical layers, and absence of meta-

data, conditions with which the discussed ontology matching approaches have difficulties.

Recently, the OAEI has introduced the Instance Matching challenge, which aims to evaluate tools able to identify similar instances, belonging to even different schemas among different RDF and OWL datasets. This notion grounds on the data interlinking movement, that has largely investigated the problem of detecting instances co-referring to the same real-world object is positively important in data integration. [16] and [10] propose a multi-layer approach for deciding whether or not two individuals are similar, based on contextual and semantic metadata. In particular, [16] proposes a tailored instance pipeline for RDF datasets composed of four stages ranging from data cleansing, unique subject matching, one-left object matching, and score matching. The scores, computed on the instances filtered by the previous stages working on the exact match, are weighted on the similarity of the metadata that surround them. [10] uses a two-stage approach composed of candidate generation and instance matching. The first phase clusters similar instances, to reduce the number of pairs. The second determines the equivalence of the individuals, measuring the TF-IDF cosine similarity at triple level for strings, inverted disparity for digits, and exact match for dates. The matching is independent from the initial schemas. Our work narrows down the instance matching task as a mere exact match of the same tokens (that occur in the same document, and at the same offset) potentially labeled using different schemas. We investigate the type distributions of the dataset labeled with the gold standard types and the one labeled with extractor types.

3 Experimental Setup

In our experiments, we use two entity classification benchmark datasets, namely CoNLL2003 and MSM2013. The corpora are annotated using off-the-self extractors that use different ontologies for classifying the entities, with some of the extractors using more than one ontology. Basic stats of the two datasets are shown in Table 1.

Table 1: Statistics on number of articles, tokens, named entities (in total and split out per class) for the CoNLL2003 and MSM2013 datasets.

CoNLL2003	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
MSM2013	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

3.1 Datasets

One of the most prominent datasets in NER is the corpus that was created for the CoNLL2003 Language-Independent Named Entity Recognition shared task [15]. There is fairly little overlap of named entities between the training and test datasets: only 2.9% of the named entities that occur in the training data also occur in the test data.

The MSM2013 corpus was created for the Making Sense of Microposts Challenge 2013 [1] and consists of microposts collected from the end of 2010 to the beginning of 2011. Similarly as for CoNLL2003, the MSM2013 has 8.1% overlap of named entities between the training and test data.

3.2 Extractors

The commercial and research tools that we evaluate via their Web APIs are AlchemyAPI,[5] dataTXT,[6] DBpedia Spotlight, Lupedia,[7] OpenCalais, TextRazor,[8] and Zemanta,[9]. For brevity, we refer to these using the uncapitalized spelling, and we shorten DBpedia Spotlight to dbspotlight. These extractors are selected for our experiments because they either utilize the DBpedia Ontology v3.8,[10] or the Freebase ontology[11] enabling us to more easily compare these extractors than the extractors that use a custom ontology. Furthermore, the DBpedia Ontology can be freely downloaded and browsed which enables us to perform experiments learning mappings at different levels in the taxonomic hierarchy (see Section 4).

The annotation results vary in terms of the schema used for classifying the phrases. For instance, the entity *Barack Obama* may be classified (depending on the context) as "Person" from alchemyapi, or as "OfficeHolder" by dbspotlight. This example shows at a first glance the subtle differences that exist while harmonizing different classification schemes. Zemanta officially claims it uses a sample of the Freebase types,[12] but in our experiments we observe that it uses a larger set of Freebase and DBpedia types.

We query these extractors by using the NERD framework [13] that acts as proxy as it harmonizes the retrieval of the annotations.

3.3 Data Preprocessing

We split each set into documents (CoNLL2003) and microposts (MSM2013). We then query the extractor e using the NERD framework, with the settings

[5] `http://www.alchemyapi.com`
[6] `https://dandelion.eu/products/datatxt`
[7] `http://lupedia.ontotext.com`
[8] `http://www.textrazor.com`
[9] `http://www.zemanta.com`
[10] `http://wiki.dbpedia.org/Ontology`
[11] `http://www.freebase.com`
[12] `http://developer.zemanta.com/docs/entity_type/` last access on April 29th, 2014.

described above. The retrieved output is parsed and converted in the CoNLL format, where the last column is dedicated to list the types T returned by e. Per each extractor, we generate one CoNLL file to list the T_{NERD} (NERD types), and one to list the native (source) types T_S returned by the extractor.

4 Inductive Typing Alignment

Let E denote the entity list, T the entity type list, S the source extractor types, and GS the types observed in the gold standard. $(E,T)_S$ indicates the ordered list of entities and types given by the source extractor, while O_S is the schema used by the source extractor to type the entities. We then define $A : T_S \rightarrow T_{GS}$ as the set of alignments given to which we apply a transformation from the T_S to the T_{GS}. Inspired by [14], we model the proposed inductive typing alignment as shown in Figure 1. *Inputs* are the ontology depth, the text token, and settings for the machine learning stage.

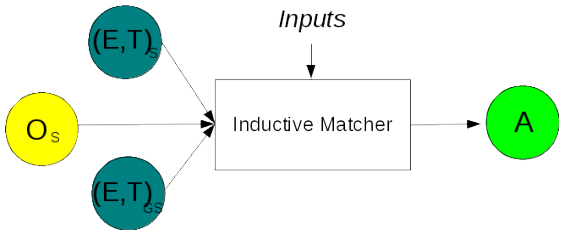

Fig. 1: The schemas matching chain.

The initial requirement for an inductive typing alignment is to rely on reasonable answers of a set of systems when performing on specific constraints and defined use cases. By the law of large numbers it is possible to infer the value range of a statistical parameter with a bounded high probability (usually from 95% to 99%) from independent random samples. Therefore, relying on a large number of observations, we can draw upon relations from different entity types. We split the inductive typing alignment into two separate tasks, the first a purely statistical approach, in which we extrapolate the evidence by observing the highest number of matches, and, the second, a machine learning approach, where a classification algorithm is trained using a set of mappings in order to infer the correct mapping for the test case.

4.1 Statistical Induction

Generally, by looking at the entity type distributions of a gold standard we can narrow down coarse-grained considerations of the dataset. Let us consider the gold standard schema as the central schema, and the extractor schemas as the

O_S. The entity surface forms work as matchers, so that we can cross the entity type distributions from the gold standard and the ones observed in the datasets described by O_S. Therefore, applying a frequency induction we imply alignments based on the peak of the distributions.

4.2 Machine Learning Induction

In our machine learning induction approach, we aim to learn which entity types as assigned by the extractor outputs correspond to which entity classes in the gold standard. We use Weka [6] v3.6.11 in our experiments. In all experiments we use separate training and test sets. We have experimented with various algorithms, but k-Nearest Neighbour (k-NN) [4] and Naive Bayes (NB) [9] are the best performing for our approach, and thus only results using these algorithms are reported.

For each extractor, we performed the following series of experiments for both k-NN with k set to 1 (called IB1 in Weka) and Naive Bayes.

NERDType we try to learn the mapping between the types assigned by the NERD ontology and the types in the gold standard dataset. This serves as a baseline to check whether the manually created mappings distribution in NERD for each extractor follows the same implicit patterns as the class distribution in the gold standard datasets.

URIType in these experiments, we try to learn the mapping between the entity type as given by the extractor and the gold standard type.

URIType First in these experiments, we try to learn the mapping between the superclass of the entity type as given by the extractor and the gold standard type.

URIType Second in these experiments, we go up one level in the extractor type ontology and try to learn the mapping between the super-super-class of the entity type as given by the extractor and the gold standard type

URIType Third in these experiments, we try to learn the mapping between the super-super-super class of the entity type as given by the extractor and the gold standard type.

It must be noted that the schemas for alchemyapi and opencalais are released in a textual format, hence we extrapolated them and created the OWLs.[13] Given the reduced depth (flat schema for opencalais, and 2 level hierarchy for alchemyapi) we could not perform experiments in which we traverse the hierarchy. Similarly, the unavailability of a machine readable Freebase schema obliged to consider the Freebase types as sequences of subtypes, separated by the terminator slash. This introduces a bias when the domain type corresponds to the identifier (for instance /person/person).

Figures 2 and 3 show the results of the mappings learnt for each extractor for the CoNLL2003 and MSM2013 datasets respectively.[14] For both datasets,

[13] `https://github.com/NERD-project/nerd-ontology`

[14] For reasons of space we only present the F-measures here, for an overview of the precision and recall see `https://github.com/giusepperizzo/nerd-inductive`.

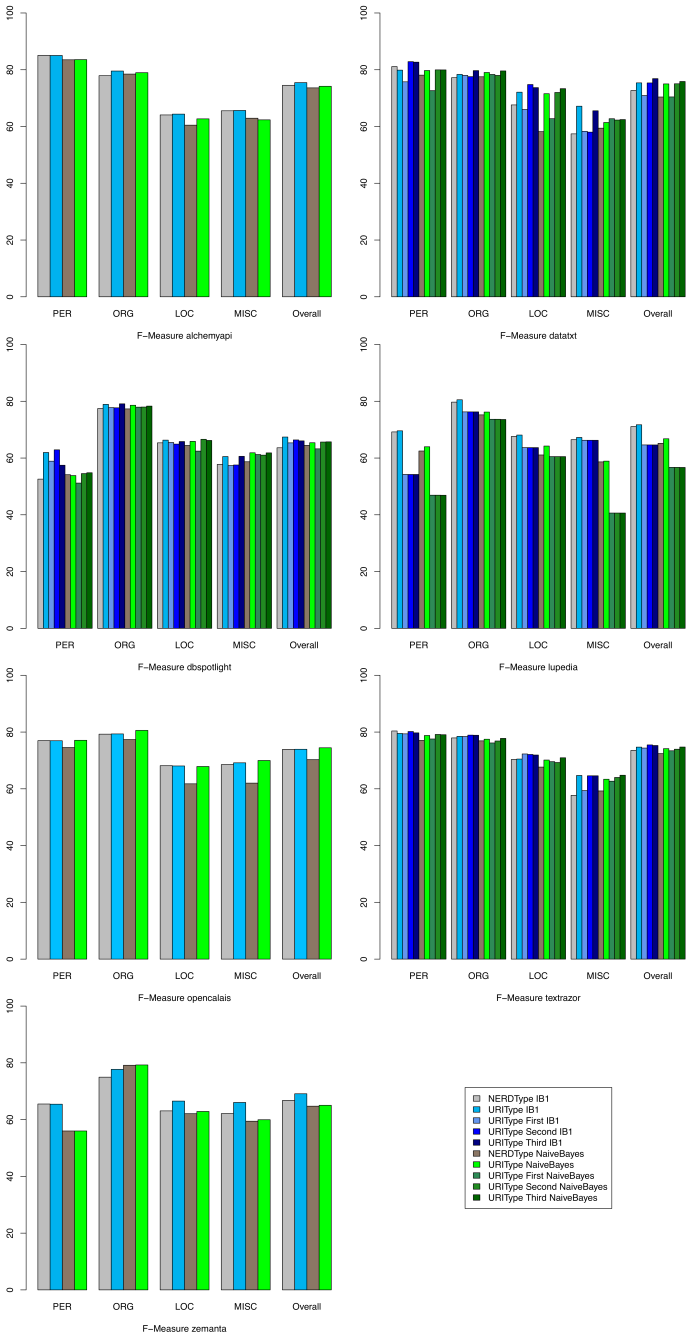

Fig. 2: F-scores of mapping experiments on the CoNLL2003 dataset on the person (PER), location (LOC), organisation (ORG), miscellaneous (MISC) and overall (Overall).

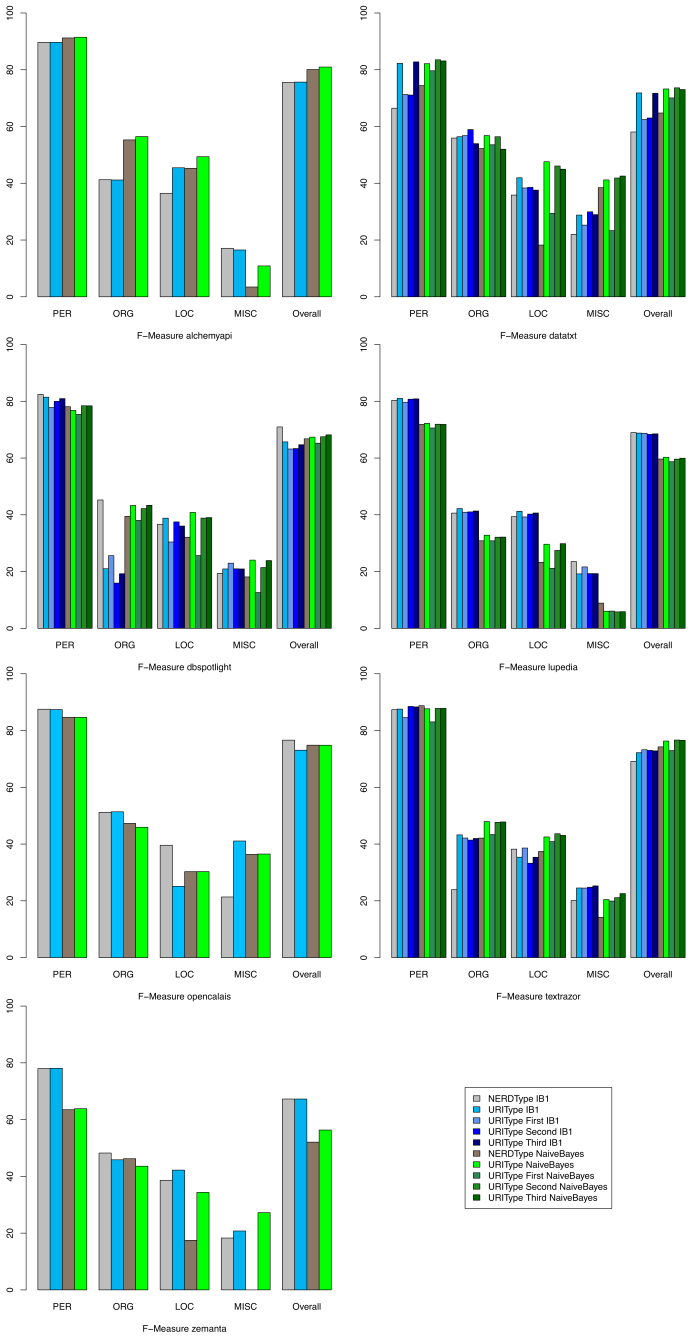

Fig. 3: Results of mapping experiments on the MSM2013 dataset on the person (PER), location (LOC), organisation (ORG), miscellaneous (MISC) and overall (Overall).

we see that the person class can be mapped to the different extractor schemas most easily. This is not surprising as this class is the least ambiguous. For the organization and location classes, the results drop, but this is mostly due to the recall of the extractors being quite low (see the recall statistics in Figures 3 and 4 of [12]). The miscellaneous class suffers from being a broad and underspecified class in both datasets, which affects both the recognition results as well as the typing and thus also the mapping. In the CoNLL specification for example, the miscellaneous class includes named sports events, whereas many of the extractors also annotate more generic event types such as *basketball championship*.

For both datasets, we find that for the extractors that use the DBpedia Ontology (datatxt, dbspotlight, lupedia and textrazor), the automatically learned mappings most often outperform the manual mappings of the NERD ontology, with the exception of the results for dbspotlight. This effect is more pronounced in the CoNLL2003 dataset than in the MSM2013 dataset. Another interesting thing to note is that the lupedia mappings can best be learnt using the IB1 algorithms, whereas the dbspotlight and datatxt mappings perform best when using the NaiveBayes classifier.

5 Evaluation and Results

Table 2a shows the results we achieve on applying the inductive approach on [12] for the extraction and classification of the CoNLL2003 corpus entities. As baseline, we report the results achieved by the same chain, but only using manual mappings. Results are computed using the conlleval script.[15] We observe that the C4.5 classifier is the best performing classifier for combining the linguistic features, the output of the Conditional Random Fields (CRF) [5], and the induced mappings from the investigated seven extractors (for both statically induction and machine learning induction), and hence for predicting the correct type. In this paper, the model used results less rich (with a reduced number of extractors) than our baseline approach.

The most striking finding from these experiments is that for the CoNLL2003 dataset, based on the recall the best mappings are obtained by using a simple frequency based statistical induction, where we choose the most frequently occurring type. This provides us with an increase of 0.45% in recall from the baseline. For the MSM2013 dataset however, the machine learning induction leads to the best results, with an increase in F1 of 0.68%, and an increase in recall of 8.59%. For both datasets, the figures report that the induction is generally promising for the bin classes, such as MISC. This is explained by the fact that the induction fills the gap left by the low number of examples used by the entity recognizer algorithm to build a model on top of that. The top mappings for the MISC class obtained by the datatxt Naive Bayes experiments are shown in Table 3.[16] Here we see the breadth of the MISC class and the differences in

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

[16] The complete mappings per extractor can be found at `https://github.com/giusepperizzo/nerd-inductive`.

Table 2: Precision, Recall and F1 results on CoNLL2003 (a) and MSM2013 (b) datasets for different classes and overall. Figures are in percentages. Boldface indicates the best score per measure.

(a)

		base-line	statistical induction	ml induction
PER	p	91.41	**91.56**	88.49
	r	92.15	**92.76**	90.70
	f	91.78	**92.16**	89.58
LOC	p	**89.27**	85.84	87.94
	r	89.81	**90.60**	89.58
	f	**89.54**	88.16	88.75
ORG	p	81.15	81.40	**82.14**
	r	**81.64**	80.21	79.54
	f	**81.39**	80.80	80.82
MISC	p	77.70	78.48	**81.50**
	r	75.93	**79.05**	78.32
	f	76.80	78.76	**79.88**
Overall	p	**86.09**	85.31	85.68
	r	86.35	**86.74**	85.56
	f	**86.22**	86.02	85.62

(b)

		base-line	statistical induction	ml induction
PER	p	88.90	88.59	**90.32**
	r	84.68	81.79	**90.00**
	f	84.74	85.05	**90.16**
LOC	p	59.43	**59.78**	51.41
	r	64.95	56.12	**74.49**
	f	**62.07**	57.89	60.83
ORG	p	**62.58**	56.02	61.83
	r	43.78	39.74	**49.15**
	f	51.52	46.50	**54.76**
MISC	p	**44.44**	20.90	18.67
	r	13.04	15.05	**30.11**
	f	20.17	17.50	**23.05**
Overall	p	**82.56**	79.32	76.79
	r	72.95	69.77	**79.22**
	f	77.46	74.24	**77.99**

the type of entities that fall within this class in the two datasets, supporting our case for customized mappings. It also shows the potential usefulness of having a more fine-grained class than MISC.

Table 3: Top mappings for MISC class as obtained in the datatxt third Naive-Bayes experiments. 'dbo:' is shorthand for `http://dbpedia.org/ontology`

CoNLL	MSM2013
dbo:Event,dbo:SportsEvent	dbo:Work,dbo:Film,
dbo:Country,dbo:Place,dbo:PopulatedPlace	dbo:Event,dbo:SportsEvent
dbo:EthnicGroup	dbo:Award
dbo:Language	dbo:Work,dbo:TelevisionSeason,
dbo:Event,dbo:SportsEvent,	dbo:Event,dbo:SportsEvent,
dbo:SoccerTournament	dbo:SoccerTournament
dbo:Award	dbo:Work,dbo:Film,dbo:TelevisionShow
dbo:Currency	dbo:Work,dbo:WrittenWork,dbo:Book

6 Discussion

The proposed approach inherits some limitations of the extractors used in this work. The annotations collected from the extractors are imperfect. The machine learning approach aims to compensate the system errors by remapping them to

the correct types. Another source of bias is the entity position, that is given by the majority of the extractors, while alchemyapi and opencalais leave the client to compute it. NERD attempts to reduce this ambiguity, recomputing the position just applying a rule-based logic. Four of the extractors potentially use more than one schema for the classification. Although this gives more information to the client, it affects the interpretation of the entity and, hence, introducing ambiguity in performing the further operations by intelligent systems plugged on. It is also unclear how some of the extractors exploit the taxonomies they use internally, which may cause suboptimal alignments. However, as some of these extractors are black boxes we can only infer how they operate by looking at the results.

Furthermore, the evaluation datasets used may not be optimal for evaluating these extractors. As mentioned in Subsection 4.2, the gold standard dataset is more conservative in its annotations, resulting in a lower precision for the extractors as they assume broader categories of entities. However, as creating gold standard benchmark datasets is a time consuming and complex task, there are not many around. Modeling choices influence the fit of the dataset for different tasks and it is inevitable that errors creep in, despite data often being annotated by multiple annotators. Minor errors may creep in, such as 'Keirin' being annotated as a location in the CoNLL dataset, whereas it should be a sport. In the same dataset, we also encounter rugby, tennis and soccer as usually not being annotated as an entity, but in some cases they are. Most of the extractors seem to tag these concepts. This presents us a with a mismatch between the dataset and the task the extractors were created for.

7 Conclusions and Future Work

We have shown an approach and experiments for learning customized taxonomy alignments between different entity extractors for different domains. We experimented with a statistical data-driven alignment, and a machine learning data-driven alignment on two NLP datasets, namely CoNLL2003 and MSM2013. We used the computed alignments as input of [12] and compared the overall results with the ones obtained just using a manual mapping. Results are encouraging and show the potentiality of the inductive approach, that remains strictly dependent on the used dataset. This validates the hypothesis that there is no one-size-fits-all approach to align different taxonomies. Part of our ongoing work is to improve the NER results to get closer to the theoretical limit presented in our previous work. In the ensemble learning domain, we plan to study the feature selection process further, and to estimate the influence of the size of the training corpus for building the classification model. We also plan to experiment with diverse datasets, covering other domains such as TV. A selection of further plots, not reported in this paper, together with the source code of our experiments, are available at `https://github.com/giusepperizzo/nerd-inductive`.

Acknowledgments

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).

References

1. Basave, A.E.C., Varga, A., Rowe, M., Stankovic, M., Dadzie, A.S.: Making Sense of Microposts (#MSM2013) Concept Extraction Challenge. In: Making Sense of Microposts (#MSM2013) Concept Extraction Challenge (2013)
2. Bellahsene, Z., Bonifati, A., Duchateau, F., Velegrakis, Y.: On Evaluating Schema Matching and Mapping. In: Schema Matching and Mapping. Data-Centric Systems and Applications (2011)
3. Bernstein, P.A., Madhavan, J., Rahm, E.: Generic Schema Matching, Ten Years Later. PVLDB 4(11), 695–701 (2011)
4. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Institute of Electrical and Electronics Engineers Transactions on Information Theory 13(1), 21–27 (1967)
5. Finkel, J.R., Grenager, T., Manning, C.: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In: 43^{rd} Annual Meeting on Association for Computational Linguistics (ACL '05) (2005)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
7. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic Schema Matching with Cupid. In: 7^{th} International Conference on Very Large Data Bases (VLDB'01) (2001)
8. Mendes, P.N., Jakob, M., Garcia-Silva, A., Bizer, C.: Dbpedia spotlight: Shedding light on the web of documents. In: 7^{th} International Conference on Semantic Systems (I-Semantics'11) (2011)
9. Mitchell, T.M.: Generative and discriminative classifiers: Naive bayes and logistic regression (October 2005), rough draft chapter intended for possible inclusion in a possible second edition of Machine Learning, T. M. Mitchell, McGraw Hill.
10. Nguyen, K., Ichise, R.: SLINT+ results for OAEI 2013 instance matching. In: 8^{th} International Workshop on Ontology Matching (OM-13) (2013)
11. Pereira Nunes, B., Dietze, S., Casanova, M., Kawase, R., Fetahu, B., Nejdl, W.: Combining a Co-occurrence-Based and a Semantic Measure for Entity Linking. In: 10^{th} Extended Semantic Web Conference (ESWC'13) (2013)
12. Rizzo, G., van Erp, M., Troncy, R.: Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web. In: 9^{th} International Conference on Language Resources and Evaluation (LREC'14) (2014)
13. Rizzo, G., Troncy, R.: NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In: 13^{th} Conference of the European Chapter of the Association for computational Linguistics (EACL'12) (2012)
14. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering 25(1) (2013)
15. Tjong Kim Sang, E.F., Meulder, F.D.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: 17^{th} Conference on Computational Natural Language Learning (CoNLL'03) (2003)
16. Zheng, Q., Shao, C., Li, J., Wang, Z., Hu, L.: RiMOM2013 results for OAEI 2013. In: 8^{th} International Workshop on Ontology Matching (OM-13) (2013)