LODIE: Linked Open Data for Web-scale Information Extraction

Fabio Ciravegna, Anna Lisa Gentile, and Ziqi Zhang

Department of Computer Science, The University of Sheffield, UK
{f.ciravegna,a.l.gentile,z.zhang}@dcs.shef.ac.uk

Abstract. This work analyzes research gaps and challenges for Web-scale Information Extraction and foresees the usage of Linked Open Data as a groundbreaking solution for the field. The paper presents a novel methodology for Web scale Information Extraction which will be the core of the LODIE project (Linked Open Data Information Extraction). LODIE aims to develop Information Extraction techniques able to (i) scale at web level and (ii) adapt to user information need. We argument that for the first time in the history of IE this will be possible given the availability of Linked Data, a very large-scale information resource, providing annotated data on a growing number of domains.

1 Introduction

Information Extraction (IE) is the technique for transforming unstructured textual data into structured representation that can be understood by machines. It is an essential technique to automatic knowledge capture, and has been an active research topic for decades. With the exponential growth of the Web, an unprecedented amount of data is made available online. Extracting information from this gigantic data source - or to be called Web-scale IE in the rest of this paper - in an effective and efficient way has been considered a major research challenge. Over the years, many different approaches [1–5] have been proposed. Nevertheless, the current state of the art has mainly addressed tasks for which resources for training are available (e.g. the TAP ontology in [1]) or use generic patterns to extract generic facts (e.g. [2], OpenCalais.com). The limited availability of resources for training has so far prevented the study of the generalised use of large-scale resources to port to specific user information needs.

This paper introduces the Linked Open Data Information Extraction (LODIE) project, a 3-year project that focuses on the study, implementation and evaluation of IE models and algorithms able to perform efficient user-centric Web-scale learning by exploiting Linked Open Data (LOD). Linked Data is [. . .] a recommended best practice for exposing, sharing, and connecting data [. . .] using URIs and RDF (www.linkeddata.org). LOD is ideally suited for supporting Web-scale IE adaptation because it is: (i) very large scale, (ii) constantly growing, (iii) covering multiple domains and (iv) being used to annotate a growing number of pages that can be exploited for training. The latter is particular interesting for IE: with the creation of schema.org, major players like Google,

Yahoo! and Bing are currently inviting Web content creators to include LOD-based microformats in their webpages in order to make the data and information contained understandable to search engines and Web robots. Similarly, RDFa is being adopted to produce annotations (http://www.w3.org/TR/xhtml-rdfa-primer). Researchers are starting to consider the use of LOD for Web-scale IE, however the approaches adopted so far are limited in scope to recognising tables [6], and extraction of specific answers from large corpora [7], but a generalised approach to the use of LOD for training large scale IE is still missing. LODIE will fill this gap by studying how an imprecise, redundant and large-scale resources like LOD can be used to support Web-scale user-driven IE in an effective and efficient way. The idea behind the project is to adapt IE methods to detailed user information needs in a completely automated way, with the objective of creating very large domain-dependent and task-dependent knowledge bases.

The remainder of this paper is organised as follows: Section 2 briefly introduces state of the art on Web-scale IE and the use of LOD in IE; Section 3 discusses the research gaps and challenges that LODIE aims to address; Section 4 introduces the LODIE methodology and architecture; Section 5 describes evaluation plan; and Section 6 concludes this paper.

2 Related Work

Adapting IE methods to Web-scale implies dealing with two major challenges: large scale and lack of training data. Traditional IE approaches apply learning algorithms that require large amount of training data, typically created by humans. However, creating such learning resources at Web-scale is infeasible in practice; meanwhile, learning from massive training datasets can be redundant and quickly become intractable [8].

Typical Web-scale IE methods adopt a light-weight iterative learning approach, in which the amount of training data is reduced to a handful of manually created examples called "seed data". These are searched in a large corpus to create an "annotated" dataset, whereby extraction patterns are generalised using some learning algorithms. Next, the learnt extraction patterns are re-applied to the corpus to extract new instances of the target relations or classes. Mostly these methods adopt a bootstrapping pattern where the newly learnt instances are selected to seed the next round of learning. This is often accompanied by some measures for assessing the quality of the newly learnt instances in order to control noisy data. Two well-known earlier systems in this area are Snowball [9] and KnowItAll [1, 2]. Snowball iteratively learns new instances of a given type of relation from a large document collection, while KnowItAll learns new entities of predefined classes from the Web. Both have inspired a number of more recent studies, including StatSnowball [10], ExtremeExtraction [4], NELL [3] and PROSPERA [5]. Some interesting directions undertaken by these systems include exploiting background knowledge in existing knowledge bases or ontologies to infer and validate new knowledge instances, and learning from negative seed data. While these systems learn to extract predefined types of information based on (limited) training data, the TextRunner [2] system proposes the "Open Information Extraction", a new paradigm that

exploits generic patterns to extract generic facts from the Web for unlimited domains without predefined interests.

The emergence of LOD has opened an opportunity to reshape Web-scale IE technologies. The underlying multi-billion triple store[1] and increasing availability of LOD-based annotated webpages (e.g., RDFa) can be invaluable resources to seed learning. Researchers are starting to consider the use of LOD for Web-scale information extraction. However, so far research in this direction has just taken off and the use of Linked Data is limited. Mulwad et al. [6] proposed a method to interpret tables based on linked data and extract new instances of relations and entities from tables. The TREC2011 evaluation on the Related Entity Finding task [7] has proposed to use LOD to support answering generic queries in large corpora. While these are relevant to our research, full user-driven complex IE task based on LOD is still to come.

LODIE will address these gaps by focussing on the following research questions: (i) How to let users define Web-IE tasks tailored to their own needs? (ii) How to automatically obtain training data (and filter noise) from the LOD? (iii) How to combine multi-strategy learning (e.g., from both structured and unstructured contents) to avoid drifting away from the learning task? (iv) How to integrate IE results with LOD?

3 LODIE - User-centric Web-scale IE

In LODIE we propose to develop an approach to Web-scale IE that enables fully automated adaptation to specific user needs. Users will be supported in defining their tasks using the LOD and IE methods and algorithms will be able to adapt to the new tasks using LOD as background knowledge. LOD will provide ontologies to formalise the user information need, and will enable seeding learning by providing instances (triples) and webpages formally annotated via RDFa or Microformats. Such background knowledge will be used to seed semi-supervised Web-scale learning. Output from the IE task will be both a set of instances to publish on the LOD, as well as a set of annotations which will provide provenance for the generated instances.

The use of an uncontrolled and constantly evolving, community provided set of independent Web resource for large-scale training is totally untapped in the current state of the art. Research has shown that the relation between the quantity of training data and learning accuracy follows a non-linear curve with diminishing returns [11]. On LOD the majority of resources are created automatically by converting legacy databases with limited or no human validation, thus errors are present [12]. Similarly, community-provided resources and annotations can contain errors, imprecision [13], spam, or even deviations from standards [14]. Also, large resources can be redundant, i.e. contain a large number of instances that contribute little to the learning task, while introducing considerable overhead. For example, the uptake of RDFa and microformat annotations is mainly happening at sites that generate webpages automatically, e.g. using a database back-end (e.g. eCommerce sites). Very regular annotations present very limited variability, and hence (i) high overhead for the learners (which will have to cope with thousands

[1] http://www4.wiwiss.fu-berlin.de/lodcloud

of examples providing little contribution) and (ii) the high risk of overfitting the model. For this reason, LODIE will put particular focus on measures and strategies to filter background knowledge to obtain noiseless and efficient learning.

The main contributions by LODIE will be:

- A method to formalise user requirements for Web-scale IE via LOD. We introduce methods based on ontology patterns [15] both to allow users to formalise their information needs and to identify relevant LOD resources to power adaptation to the task.
- Methods to evaluate the quality of LOD data and to select the optimal subset to seed learning. We introduce two measures: (i) Variability: to select seeds able to provide the learner with the optimal variety, so to avoid overfitting and overhead; this is expected to increase recall in extraction; (ii) Consistency to identify noisy data; this is expected to increase the precision of the IE process while reducing overhead during learning.
- The development of efficient, iterative, semi-supervised, multi-strategy Web-scale learning methods robust to noise and able to avoid drifting away when re-seeding. The methods will be able to exploit local and global regularities (e.g. page and site-wide regularities) as well redundancy in information [16].
- An evaluation process where we will test the above mentioned models in a number of tasks in order to compare them with the state of the art, both by defining tasks to be reused by other researchers and by participating in international competitions on large scale IE. The level of complexity of using large scale uncontrolled resources to seed Web-scale IE has never been previously addressed.

4 LODIE - Architecture and Methodology

We define Web-scale IE as a tuple: $< T, O, C, I, A >$ where: T is the formalisation of the user information needs (i.e. an IE Task); O is the set of ontologies on the LOD. C is a large corpus (typically the Web) which can be annotated already in part (C_L) with RDFa/Microformats; we refer to the unannotated part as C_U. I represents a collection of instances (knowledge base) defined according to O; I_L is a subset of I containing instances already present on the LOD; I_U is the subset of I containing all the instances generated by the IE process when the task is executed on C. A is a set of annotations and consists of two parts: A_L are found in C_L, and A_U are created by the IE process; A_U can be the final set or the intermediate sets created to re-seed learning.

The proposed method for IE applies a semi-supervised approach, based on identification of weak seeds for learning (high recall) followed by a filtering process that ensures only the candidates that are reasonably certain (precision) are used to (re)seed learning. We will work on an extension of the model we presented in [17] where (i) an initial set of seed instances I_L is identified, (ii) candidate annotation A_L and A_U are identified from C_L and C_U; (iii) a learning model is learned using (C_C, I_L, A_L, A_U), (iv) information is extracted by applying the model to C_C to generate I_U, A_U and (v) the new annotations are used to reseed another round of learning.

The overview of the LODIE approach is shown in Figure 1. The workflow includes the formalisation of the task T using LOD, the identification and optimisation of I and A to seed learning, the study of semi-supervised multi-strategy IE learning models, and the publication of A_U and I_U to LOD.

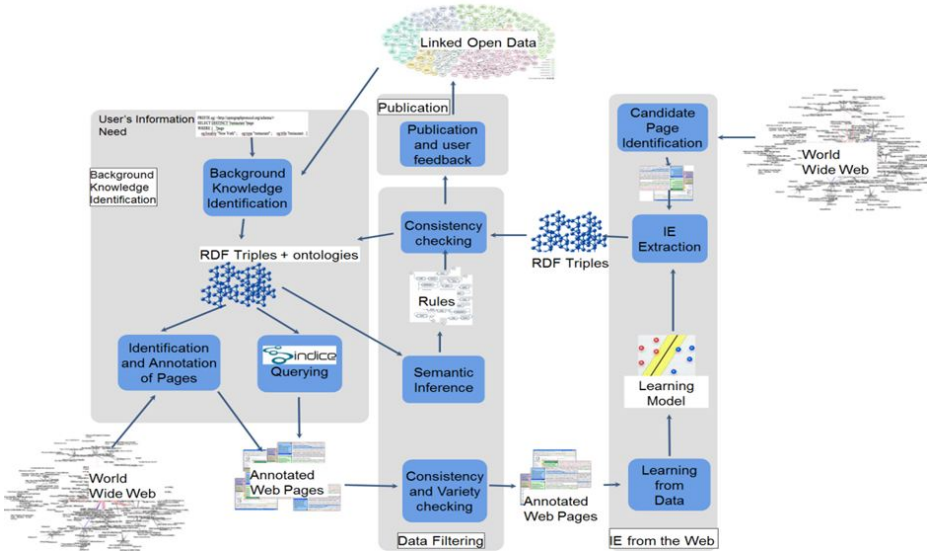

Fig. 1. Architecture diagram.

4.1 User needs formalisation

The first requirement for adapting Web-scale IE to specific user needs is to support users in formalising their information needs in a machine understandable format. Formally we define the user needs as a function: $T = f(O) \rightarrow O_L$ identifying a view on the LOD ontologies describing the information extraction task. T will be materialised in the form of an OWL ontology. We propose two ways to define T. The baseline strategy will be bottom up and will include: (i) identifying manually relevant ontologies and concepts on the LOD by using search engines like Swoogle (swoogle.umbc.edu) and Watson (watson.kmi.open.ac.uk) and (ii) manually defining a view on them using a standard tool like the Neon Toolkit (neon-toolkit.org). The second, more challenging strategy will be top-down and will be based on: (i) the formalisation of user needs using Content Ontology Design Patterns (Content ODP) [18] and (ii) the matching of the resulting ontology with existing LOD ontologies using Reeningeering ODPs.

An ODP is a reusable successful solution to a recurrent modelling problem. Content ODPs are patterns that describe a conceptualization addressing specific requirements,

e.g. in terms of competency questions or reasoning tasks. Content ODPs can be manifested as OWL ontologies, i.e. small OWL building blocks. User requirements will be described in terms of specializations of general Content ODPs, i.e. by specialising the general Content ODPs using user terminology. This will generate an ideal ontology describing the task. This ideal task will then have to be mapped to the reality of the LOD. First, relevant ontologies will be found using search engines like Swoogle and Watson. Then, transformational ODPs are used to turn the generated ontology into a view on the LOD by matching its concepts and relations with those actually found on the LOD. We will use Reeningeering Patterns, e.g. transformation recipes [19], currently proposed for semantically grounded triplifications. Reengineering Patterns will here be applied to map the user-generated semantically grounded ontology to an existing LOD ontology. This represent a kind of reverse approach than generally used in literature where Reengineering ODPs are used to map a database schema to a semantically grounded ontology. We will develop a user interface to define the IE task which will guide the user in an effective and efficient way. We will identify relevant Content and Reengineering ODP for the IE task and if necessary develop new ones. Application of patterns will be done using the Neon XD Tools plugin [20] and the Semion tool (stlab.istc.cnr.it/stlab/Semion).

4.2 Learning seed identification and filtering

A set of triples I_L relevant to the users need are identified as side effect of the definition of T: they can be retrieved from existing LOD knowledge bases associated with the types in T. We will use search engines like Sindice to identify RDFa and Microformat A_L which are associated to the types in T (if available). To these, we will add further candidates A_U identified by searching the Web for linguistic realisation of the triples I_L. In order to reduce noise due to ambiguity of the linguistic realisations [17], we will look for co-occurrence of known related instances in the same textual contexts (e.g. sentences [3]), and structural elements (e.g. tables and lists [21]) and apply focussing techniques (e.g. relevant ranking [7]).

These annotations together with A_L are used by the multi-strategy learning process to create new candidate annotations and instances. We have adopted similar approaches in [17] and [21]. Before feeding the identified annotations to the learning process, they will be filtered to ensure high quality in training data. This is achieved by using two measures, the measures of consistency and variability.

Filtering seeds - consistency measure: We will define a measure of consistency to filter A to prevent the learning algorithm to be misled by spurious data. Our hypothesis is that good data should present consistency with respect to the learning task. We will cast filtering as a problem of detecting noise in training data [22, 23]. These methods usually apply an ensemble of supervised classifiers to the training data and identify the noisy examples as those demonstrating high level of inconsistency in terms of the labels produced by classifiers. However in doing so, the classifiers used to detect noisy examples are constructed initially from a training set already containing noise, which may introduce bias in the process [23].

We propose to evaluate the consistency of the annotations by applying unsupervised clustering techniques and study the cluster membership of individual examples. We will map each $a \in A$ to a feature vector representing its form (superficial and semantic), other entities it appears with (together with their types and their reciprocal relations, from simple co-occurrence to specific relations), and other words it appears with in the sentence, etc. Then we will exploit unsupervised clustering techniques to split the data into clusters. Each generated cluster will be associated to a specific class by assigning the type of the largest majority of instances; ambiguous clusters will be discarded. The clustering procedure will be repeated iteratively under different settings; each a will then be assigned a value of consistency, which will be a function of how a consistently scores in the clusters associated to its actual type during the iterative process.

To minimise computation we will apply sampling methods to A to create a representative sample of manageable size. We will introduce methods to mathematically formulate the assessment of consistency based on an annotations cluster membership behaviour. The consistency score will be used to confirm the validity of a both before seeding (or re-seeding) learning and before the generation of the final set of annotations.

Optimising seeds - variability measure: Large numbers of examples in a very large resource like the LOD can contribute little to learning while substantially increase the computational overhead. The issue is increased when semi-supervised algorithms use self-learning (i.e. re-seeding) as strategy (e.g., [1, 3]) because, due to the nature of information redundancy on the Web, it is highly likely that a large portion of the reseeding data is also redundant. Very little has been done to prevent this issue in large scale IE. We hypothesize that good data should also present variability with respect to the learning task. Thus we introduce the notion of variability in the IE task and propose a novel measure to address this.

Given the annotations $t_A \subseteq A$ associated with one specific type t, we use the variability measure to evaluate t_A and select a subset $t_{A'} \subseteq t_A \subseteq A$ to (re-)seed learning for the type t. The measure of variability is adapted from the consistency measure. We will start by mapping each $a \in A$ to a feature vector representing its form in the same way as in the consistency measure. Then we will apply an agglomerative clustering algorithm [24] so that t_A will be clustered into a number of groups and the centroid of each cluster can be computed. The variability of the data collection t_A should reflect the number of clusters derived naturally and the distribution of members in each cluster. Intuitively, a higher number of clusters imply a higher number of groups of different examples, which ensures more extraction patterns to be learnt to ensure coverage; while even distribution of cluster members ensures the patterns can be generalised for each group. We hypothesize the variability of each $a \in A$ be dependent on the general variability of the collection, and on their distance to the centroid of each cluster because intuitively, the closer an element is to the centroid, the more representative it is for the cluster. We will introduce methods to mathematically formulate the variability based on these factors. At the end of the process we will have selected a subset $t_{A'} \subseteq t_A \subseteq A$.

4.3 Multi-strategy Learning

The seed data identified and filtered in the previous steps are submitted to a multi-strategy learning method, which is able to work in different ways according to the type of webpages the information is located in: (i) a model M_S able to extract from regular structures such as tables and lists; (ii) a model M_W wrapping very regular web sites generated by backing databases and (iii) a model M_T for information in natural language based on lexical-syntactic extraction patterns.

As for extracting from regular structures, following early work by [25, 26], we will adopt a strategy able to exploit the dependencies among entities expressed in one page/site to learn to extract from that page. As an example, for tables we will build a feature model based on text in each cell, as well as text from column label and text in the possibly related entities (text from cells in the same row). Moreover, when two or more annotations $a_W \in A$ of compatible type W appear in the same substructure (e.g. same column) in a document in C_U, and other candidates $a_X \in A$ of compatible type X bearing a relation r with a_W can be found in other parts of the same structure (e.g. other columns in the same table), we will hypothesize that all the other elements in those sub-structures will be of the type W and X and carry the same relation r. As a result, we will output a number of potential annotations $a \in A_U$ for each candidate in the table. To decide the best type assignment for each column we will initially experiment with strategies such as least common ancestors and majority [26] and compare and combine them with methods exploiting an enhanced feature model, that will take into account the semantics and restrictions in O [21].

For learning to wrap a site given one of its pages containing a potential reference to $a_{jW} \in A$, we will check if other pages from the same site are on the to do list for T and contain other $a_{iW} \in A$ of compatible type W in equivalent position (i.e. same XPath). If they do, we will suppose the site is to be wrapped and will extract from all the site pages that follow the identical XPath structure. As a result, we will output a number of potential annotations $a \in A_U$. Exploiting structural patterns of web pages for Information Extraction is often referred as wrapper induction [27]. We will experiment with both bottom-up and top-down strategies to wrapping [28] and combine structural and content elements from the pages.

Finally for all other cases, we will learn shallow patterns. As opposed to approaches based on complex machine learning algorithms (e.g. random walks in [24]), we will focus on lexical-syntactic shallow pattern generalization algorithms. The patterns will be generalised from the textual context of each $a \in A$ and will be based on features such as words (lexical), part of speech (syntactic) and expected semantics such as related entity classes. We will base the algorithm on our previous research in [21]. The innovation will be focused on modifying the algorithm to account for negative examples, and enriching the pattern representation with semantics mined from external knowledge resources, such as fine-grained entity labels as in [29]. The patterns are then applied to other webpages to create new candidate annotations.

At the end of this process, we concatenate the candidate annotations extracted by each learning strategy and create a collection of candidates $a \in A_U$. These will refer to instances already known (I_L) as well as new instances (I_U). The goal of the next steps will be to create new triples in I. Also to form an iterative semi-supervised learning

pattern, the annotations will be selected to reseed the new round of learning. In order to prevent the learner from drifting away if noisy data is permitted to creep in re-seeding [30, 28, 3], we will filter the candidate A_U by their consistency, and optimize them by their variability (see previous section). The learning continues in a never-ending fashion to consistently update the knowledge base [3]. Results will be visualized as soon as they are produced and users will be able to subscribe to their queries and be notified and updated when new facts of interest are mined.

4.4 Publication of new triples in the LOD

We will develop methods to enable the learned knowledge to be published and integrated into the LOD by exposing a SPARQL endpoint. In order to do so, the candidates A_U identified by IE will be assigned to a URI, i.e. a unique identifier. We call this step disambiguation [17]. The core of our disambiguation process will be exploiting features to obtain the optimal representation of each candidate set. We will use both co-occurrence based features (gathered from the context of occurrence of a given noun phrase) and relational features (obtained by exploring relational properties in the ontologies) [31]. As scalability is a major requirement both in terms of T and C, we will explore methods with minimum requirements in computational terms such as simple feature overlapping based methods [32] and string distance metrics [13]. We will compare their effectiveness with that of more computationally intensive machine learning methods such as HMM [17], random walks [24] etc.

Finally, in order to correct mistakes and improve the quality of both data and learning, a user-friendly interface will be created to enable users to provide feedback by correcting mistakes in both the knowledge base and the annotations. Strategies such as those employed by WIQA (Information Quality Assessment Framework) [33] using different information filtering policies will be employed. Corrections made by users are collected as feedback to the learning process. These are fed into new learning cycles and it is anticipated that with minimum and voluntary user feedback the learning process can improve over time. It has been shown that learning systems benefit largely from very little human supervision [11].

5 Evaluation

In order to test the effectiveness of the IE algorithms we will test both the suitability of the approach to formalise the user needs and the suitability of the approach to IE.

As for the definition of user needs, we will test the approach by giving a task described in natural language to experts in IE with a reasonable understanding of LOD and asking them to define an equivalent IE task. Evaluation will consider (i) feasibility and efficiency: can a user develop a task in a reasonable time with limited overhead using ontology patterns? We will test this by timing the task and comparing with the use of the baseline method; (ii) effectiveness: is the result really representative of the user needs? Are the resulting task ontology and the associated triples/annotations suitable to

seeding IE? This will be assessed in two ways: on the one hand users will have to judge the resulting T, A_L and I_L as relevant to their needs; on the other hand we will evaluate the returned triples in terms of usefulness to learning using the quality measures described below.

As for the effectiveness of the IE process, we will measure empirically different aspects of the learning strategy, from the different algorithms, different versions of the measures, etc. To separate this aspect of evaluation from the user evaluation, we will define a new task based on population of sections of the schema.org ontology and we will test the effectiveness of the IE system in different configurations. Typically standard measures of evaluation for IE are based on precision; since the unbounded domain and sheer amount of data on the Web makes it largely impossible to study other measures such as recall. However, besides precision, we will attempt also a partial evaluation of recall by providing the system with just a fraction of the available A_L and checking recall with respect to the A_L not provided for training. Moreover, we plan to participate in comparative large scale IE evaluations such as the TAC Knowledge Base Population [34] or the TREC Entity Extraction task [7] to compare our technology with the state of the art.

6 Conclusion

LODIE is a project addressing complex challenges that we believe are novel and of high interest to the scientific community. It is timely because (i) for the first time in the history of IE a very large-scale information resource is available, covering a growing number of domains and (ii) of the very recent interest in the use of Linked Data for Web extraction. Potential for exploitation is very high. A number of challenges are ahead and require the use of technologies from fields such as knowledge representation and reasoning, IE and machine learning. We intend to use knowledge patterns to formalise user requirements for Web-scale IE. We will develop efficient iterative semi-supervised multi-strategy Web-scale learning methods robust to noise and able to avoid drifting away when re-seeding. Particular focus will be put on efficient and robust methods: we will develop and test methods to evaluate the quality of LOD data for training and to select the optimal subset to seed learning.

Acknowledgments

The LODIE project (Linked Open Data Information Extraction) is funded by the Engineering and Physical Sciences Research Council, Grant Reference: EP/J019488/1.

References

1. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-Scale Information Extraction in KnowItAll (Preliminary Re-

sults). In: WWW2004 Proceedings of the 13th international conference on World Wide Web. (2004) 100–110
2. Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction for the web. In: IJCAI'07 Proceedings of the 20th international joint conference on Artifical intelligence. (2007) 2670–2676
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr, E.R.H., Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning. In: Proceedings of the Conference on Artificial Intelligence (AAAI). (2010) 1306–1313
4. Freedman, M., Ramshaw, L.: Extreme extraction: machine reading in a week. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, ACL (2011) 1437–1446
5. Nakashole, N., Theobald, M.: Scalable knowledge harvesting with high precision and high recall. WSDM '11 Proceedings of the fourth ACM international conference on Web search and data mining (1955) (2011) 227–236
6. Mulwad, V., Finin, T., Syed, Z., Joshi, A.: Using linked data to interpret tables. In: First International Workshop on Consuming Linked Data (COLD2010). (2010)
7. Balog, K., Serdyukov, P.: Overview of the TREC 2010 Entity Track. In: Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010), NIST (2011)
8. Joachims, T.: Making large scale SVM learning practical. In B. Scholkprof, C.J.C.Borges, Smola, A., eds.: Advances in kernel methods. MIT Press, Cambridge, MA, USA (1999) 169–184
9. Agichtein, E., Gravano, L., Pavel, J.: Snowball: a prototype system for extracting relations from large text collections . ACM SIGMOD . . . (2001) 612
10. Zhu, J.: StatSnowball : a Statistical Approach to Extracting Entity. In: WWW '09 Proceedings of the 18th international conference on World wide web. (2009) 101–110
11. Thompson, C.A., Hall, V., Mooney, R.J.: Active Learning for Natural Language Parsing and Information Extraction LEARNING SYSTEMS. In: Proceedings of the Sixteenth International Conference on Machine Learning. ICML 99. (1999) 406–414
12. Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., Aumueller, D.: Triplify Light-Weight Linked Data Publication from Relational Databases. WWW '09 Proceedings of the 18th international conference on World wide web (2009) 621–630
13. Lopez, V., Nikolov, A., Sabou, M., Uren, V.: Scaling up question-answering to linked data. In: Proceedings of the 17th international conference on Knowledge engineering and management by the masses. EKAW10. (2010) 193–210
14. Halpin, H., Hayes, P., McCusker, J.: When owl: sameas isn't the same: An analysis of identity in linked data. In: Proceedings of 9th International Semantic Web Conference ISWC 2010. (2010) 305–320
15. Gangemi, A., Presutti, V.: Towards a pattern science for the Semantic Web. Semantic Web **0** (2010) 1–7
16. Blanco, L., Bronzi, M., Crescenzi, V., Merialdo, P., Papotti, P.: Redundancy-driven web data extraction and integration. Procceedings of the 13th International Workshop on the Web and Databases - WebDB '10 (2010)
17. Rowe, M., Ciravegna, F.: Disambiguating identity web references using Web 2.0 data and semantics. Web Semantics: Science, Services and Agents on the World Wide Web **8**(2-3) (July 2010) 125–142
18. Presutti, V., Gangemi, A.: Content ontology design patterns as practical building blocks for web ontologies. In: Conceptual Modeling-ER 2008. (2008) 128–141
19. Nuzzolese, A., Gangemi, A.: Fine-tuning triplification with Semion. Proceedings of the 1st Workshop on Knowledge Injection and Extraction from LD at EKAW 2010 (2010)

20. Blomqvist, E., Presutti, V., Daga, E., Gangemi, A.: Experimenting with eXtreme Design. In: Proceedings of the 17th international conference on Knowledge engineering and management by the masses. EKAW10. (2010) 120–134
21. Ciravegna, F., Chapman, S., Dingli, A.: Learning to harvest information for the semantic web. The Semantic Web: (1) (2004)
22. Jiang, Y., Zhou, Z.h.: Editing Training Data for kNN Classifiers with. In: ISNN 2004, International Symposium on Neural Networks. (2004) 356–361
23. Valizadegan, H., Tan, P.: Kernel Based Detection of Mislabeled Training Examples. In: Proceedings of the Seventh SIAM International Conference on Data Mining. (2007) 309–319
24. Iria, J., Xia, L., Zhang, Z.: Wit: Web people search disambiguation using random walks. SemEval '07 Proceedings of the 4th International Workshop on Semantic Evaluations (June) (2007) 480–483
25. Milne, D., Witten, I.: Learning to link with wikipedia. In: CIKM '08 Proceedings of the 17th ACM conference on Information and knowledge management. (2008) 509–518
26. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and Searching Web Tables Using Entities , Types and Relationships. Proceedings of the VLDB Endowment **3**(1-2) (2010) 1338–1347
27. Kushmerick, N.: Wrapper Induction for information Extraction. In: IJCAI97. (1997) 729–735
28. Dalvi, N., Kumar, R., Soliman, M.: Automatic wrappers for large scale web extraction. Proceedings of the VLDB Endowment **4**(4) (2011) 219–230
29. Kazama, J., Torisawa, K.: Inducing gazetteers for named entity recognition by largescale clustering of dependency relations. In: Proceedings of ACL-08: HLT. (2008) 407–415
30. Curran, J., Murphy, T., Scholz, B.: Minimising semantic drift with mutual exclusion bootstrapping. In: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics. (2007) 172–180
31. Krishnamurthy, J., Mitchell, T.: Which noun phrases denote which concepts? In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Volume 15213. (2011) 570–580
32. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, London, UK, Springer-Verlag (2002) 136–145
33. Bizer, C., Cyganiak, R.: Quality-driven information filtering using the WIQA policy framework. Web Semantics: Science, Services and Agents on the World Wide Web **7**(1) (January 2009) 1–10
34. Ji, H., Grishman, R.: Knowledge base population: Successful approaches and challenges. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - HLT 11. (2011) 1148–1158