Integrating Open and Closed Information Extraction: Challenges and First Steps

Arnab Dutta[1], Mathias Niepert[2], Christian Meilicke[1], Simone Paolo Ponzetto[1]

[1] Research Data and Web Science, University of Mannheim, Germany
[2] Computer Science and Engineering, University of Washington, USA
{arnab,christian,simone}@informatik.uni-mannheim.de
mniepert@cs.washington.edu

Abstract. Over the past years, state-of-the-art information extraction (IE) systems such as NELL [5] and ReVerb [9] have achieved impressive results by producing very large knowledge resources at web scale with minimal supervision. However, these resources lack the schema information, exhibit a high degree of ambiguity, and are difficult even for humans to interpret. Working with such resources becomes easier if there is a structured information base to which the resources can be linked. In this paper, we introduce the integration of open information extraction projects with Wikipedia-based IE projects that maintain a logical schema, as an important challenge for the NLP, semantic web, and machine learning communities. We describe the problem, present a gold-standard benchmark, and take the first steps towards a data-driven solution to the problem. This is especially promising, since NELL and ReVerb typically achieve a very large coverage, but still still lack a full-fledged clean ontological structure which, on the other hand, could be provided by large-scale ontologies like DBpedia [2] or YAGO [13].

Keywords: Information extraction, Entity Linking, Ontologies

1 Introduction

Research on information extraction (IE) systems has experienced a strong momentum in recent years. While Wikipedia-based information extraction projects such as DBpedia [1, 17] and YAGO [25, 13] have been in development for several years, systems such as NELL [5] and ReVerb [9] that work on very large and unstructured text corpora have more recently achieved impressive results. The developers of the latter systems have coined the term *open* information extraction (OIE), to describe information extraction systems that are not constrained by the boundaries of encyclopedic knowledge and the corresponding fixed schemata that are, for instance, used by YAGO and DBpedia. The data maintained by OIE systems is important for analyzing, reasoning about, and discovering novel facts on the web and has the potential to result in a new generation of web search engines [7]. At the same time, the data of *open* IE projects would benefit from a corresponding logical schema even if it was incomplete and

light-weight in nature. Hence, we believe that the problem of integrating open and schema-driven information extraction projects is a key scientific challenge. In order to integrate existing IE projects we have to overcome a difficult problem of linking different manifestations of the same real world object, or more commonly the task of entity resolution. The fact that makes this task challenging is that triples from such systems are underspecified and ambiguous. Let us illustrate this point with an example triple from NELL where two terms (subject and object) are linked by some relationship (predicate):

`agentcollaborateswithagent(royals, mlb)`

In this triple, `royals` and `mlb` are two terms which are linked by some relation `agentcollaborateswithagent`. Interpreting these terms is difficult since they can have several meanings, including very infrequent and highly specialized ones, which are sometimes difficult to interpret even for humans. Here, `royals` refers to the baseball team *Kansas City Royals* and `mlb` to *Major League Baseball*.

In general, due to the fact that information on the Web is highly heterogeneous, there can be a fair amount of ambiguity in the extracted facts. The problem becomes even more obvious when we encounter triples like:

`bankbankincountry(royal, ireland)`

Here, `royal` refers to a different real-world entity, namely the *Royal Bank of Scotland*. Hence, it is important to uniquely identify the terms in accordance with the contextual information provided by the entire triple. In this paper, we aim at aligning such polysemous terms from *open* IE systems to instances from a *closed* IE system, while focusing on NELL and DBPEDIA in particular.

The remainder of the paper is organized as follows, in Section 2 we introduce the information extraction projects relevant to our work. We present our baseline algorithm for finding the best matching candidates for a term in Section 3 and in Section 4 introduce a gold standard for evaluating its performance. In Section 5 we report performance results of the proposed approach. In Section 6 we discuss related work on information extraction and entity linking. Finally, we conclude the paper in Section 7.

2 Information Extraction Projects: A Brief Overview

The **N**ever **E**nding **L**anguage **L**earning [5] (NELL) project's objective is the creation and maintenance of a large-scale machine learning system that continuously *learns* and extracts structured information from unstructured web pages. Its extraction algorithms operate on a large corpus of more than 500 million web pages[1] and not solely on the set of Wikipedia articles. The NELL system was bootstrapped with a small set of classes and relations and, for each of those, 10-15 positive and negative instances. The guiding principle of NELL is to build several semi-supervised machine learning [6] components that accumulate instances of the classes and relations, re-train the machine learning algorithms with these instances as training data, and re-apply the algorithms to

[1] `http://lemurproject.org/clueweb09/`

extract novel instances. This process is repeated indefinitely with each re-training and extraction phase called an iteration. Since numerous extraction components work in parallel and extract facts with different degrees of confidence in their correctness, one of the most important aspects of NELL is its ability to combine these different extraction algorithms into one coherent model. This is also accomplished with relatively simple linear machine learning algorithms that weigh the different components based on their past accuracy.

NELL has been running since 2010, initially fully automated and without any human supervision. Since it has experienced concepts drift for some of its relations and classes, that is, an increasingly worse extraction performance over time, NELL now is given some corrections by humans to avoid this long-term behavior. NELL does not adhere to any of the semantic web standards such as RDF or description logic.

DBpedia [1, 17] is a project that aims at automatically acquiring large amounts of structured information from Wikipedia. It extracts information from infobox templates, categories, geo-coordinates, etc.. However, it does not learn relations from the Wikipedia categories. This template information is mapped to an ontology. In addition, it has a fixed set of classes and relations. Moreover, the ontology is with more than 1000 different relations much broader than other existing ontologies like YAGO [25] or semantic lexicons like BabelNet [19].

DBPEDIA represents its data in accordance with the best-practices of publishing linked open data. The term *linked data* describes an assortment of best practices for publishing, sharing, and connecting structured data and knowledge over the web [2]. DBPEDIA's relations are modeled using the resource description framework (RDF), a generic graph-based data model for describing objects and their relationships. The entities in DBPEDIA have unique URIs. This makes it appropriate as our reference knowledge base to which we can link the terms from NELL. In the case of the examples from Section 1, by linking the terms appropriately to DBPEDIA, we are able to attach an unambiguous identifier to them which was initially missing.

`royals` ⇒ `http://dbpedia.org/resource/Kansas_City_Royals`

`royal` ⇒ `http://dbpedia.org/resource/The_Royal_Bank_of_Scotland`

3 Methodology

Wikipedia is an exhaustive source of unstructured data which has been extensively used to enrich machines with knowledge [15]. In this work we use Wikipedia as an entity-tagged corpus [4] in order to bridge knowledge encoded in NELL with DBPEDIA. Since there is a corresponding DBPEDIA entity for each Wikipedia article [2], we can in fact formulate our disambiguation problem as that of linking entities mentioned within NELL triples to their respective Wikipedia articles. Our problem is that, due to polysemy, often a term from NELL can refer to several different articles in Wikipedia or, analogously, instances in DBPEDIA. For

instance, the term *jaguar* can refer to several articles such as the car, the animal and so on.

In this work we accordingly explore the idea of using Wikipedia to find out the most probable article for a given term. Wikipedia provides regular data dumps and there are off-the-shelf preprocessing tools to parse those dumps. We used WikiPrep [11, 10] for our purpose. WikiPrep removes redundant information from the original dumps and creates more relevant XML dumps with additional information like the number of pages in each category, incoming links to each Wikipedia article and their anchor text, and a lot more[2]. In our work, we are primarily interested in the link counts, namely the frequency of anchor text labels pointing to the same Wikipedia page. Table 1 shows some of the articles the anchors *jaguar* or *lincoln* are referring to. Intuitively, out of all the outgoing links from the anchor term *jaguar*, 1842 links pointed to the article *Jaguar Cars* and so on. Essentially, these anchors are analogous to the NELL terms. Based on these counts, we create a ranked list of articles for a given anchor[3].

anchor	Article	Link count
jaguar	Jaguar Cars	1842
jaguar	Jaguar Racing	440
jaguar	Jaguar	414
...	...	...
lincoln	Lincoln, England	1844
lincoln	Lincoln, Nebraska	920
lincoln	Lincoln (2012 film)	496
...	...	...

Table 1. Snippet of those articles linked to using the anchor *jaguar* and *lincoln*.

As seen in Table 1, the output from WikiPrep can often be a long list of anchor-article pairs and some of them having as low as just one link count. Accordingly, we adopt a probabilistic approach in selecting the best possible DBpedia instance. For any given anchor in Wikipedia, the fraction of articles the links points to is proportional to the probability that the anchor term refers to the particular article [23]. More formally, suppose some anchor e refers to N articles $A_1, \ldots, A_N$ with $n_1, \ldots, n_N$ respective links counts, then the conditional probability P of e referring to A_j is given by, $P(A_j|e) = n_j / \sum_{i=1}^{N} n_i$. We compute the probabilities for each terms we are interested in and from the ranked list of descending $P(A_j|e)$, top-k candidates are selected. The choice of k is described in Section 4. We apply this idea on the Nell data set. For each Nell triple, we take the terms occurring as subject and object, and apply the procedure above.

4 Creating a Gold Standard

Nell provides regular data dumps[4] consisting of facts learned from the Web. Based on this data we create a frequency distribution over the predicates. To this end, we first clean up the data from the dumps (since these contain additional

[2] `http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/`

[3] Note that, while there are alternative data sets such as the Crosswiki data [23], in this work we opted instead for exploiting only Wikipedia internal-link anchors since we expect them to provide a cleaner source of data.

[4] `http://rtw.ml.cmu.edu/rtw/resources`

Top predicates	Instances	Random predicates	Instances
generalizations	1297709	**personleads-organization**	716
proxyfor	5540	countrylocatedingeopoliticallocation	632
agentcreated	4354	**actorstarredinmovie**	537
subpartof	3262	**athleteledsportsteam**	294
atlocation	2877	personbornincity	285
mutualproxyfor	2803	**bankbankincountry**	246
locationlocatedwithinlocation	2159	**weaponmadeincountry**	188
athleteplayssport	2076	athletebeatathlete	148
citylocatedinstate	2010	**companyalsoknownas**	107
professionistypeofprofession	1936	**lakeinstate**	105
subpartoforganization	1874		
bookwriter	1809		
furniturefoundinroom	1674		
agentcollaborateswithagent	1541		
animalistypeofanimal	1540		
agentactsinlocation	1490		
teamplaysagainstteam	1448		
athleteplaysinleague	1390		
worksfor	1303		
chemicalistypeofchemical	1303		

Table 2. The 30 most frequent predicates found in NELL. The set of predicates we randomly sampled for the gold standard are in bold.

information, such as, for instance, iteration of promotion, best literal strings, and so on[5], which are irrelevant to our task). In Table 2, we list the 30 most frequent predicates. Since the gold standard should not be biased towards predicates with many assertions we randomly sampled 12 predicates from the set of predicates with at least 100 assertions (highlighted in bold in the table). In this paper, we focus on this smaller set of predicates due to the time consuming nature of the manual annotations we needed to perform. However, we plan to continuously extend the gold standard with additional predicates in the future.

For each NELL predicate we randomly sampled 100 triples. We assigned each predicate and the corresponding list of triples to an annotator. Since we wanted to annotate a large number of triples within an acceptable time frame, we first applied the method described in Section 3 to generate possible mapping candidates for the NELL subject and object of each triple. In particular, we generated the top-3 mappings, thereby avoiding generation of too many possible candidates, and presented those candidates to the annotator. Note that in some cases (see Table 3), our method could not determine a possible mapping candidate for a NELL instance. In this case, the triple had to be annotated without presenting a matching candidate for subject or object or both. In our setting, each annotation instance falls under one of the following three cases:

(i) One of the mapping candidates is chosen as the correct mapping, i.e., the simplest case.
(ii) The correct mapping is not among the presented candidates (or no candidates have been generated). However, the annotator can find the correct

[5] `http://rtw.ml.cmu.edu/rtw/faq`

Nell-Subject	Nell-Object	DBP-Subject	DBP-Object
stranger	albert-camus	The_Stranger_(novel)	Albert_Camus
	1st cand.	Stranger_(comics)	Albert_Camus
	2nd cand.	Stranger_(Hilary_Duff_song)	-
	3rd cand.	Characters_of_Myst	-
gospel	henry_james	?	Henry_James
	1st cand.	Gospel_music	Henry_James
	2nd cand.	Gospel	Henry_James_(basketball)
	3rd cand.	Urban_contemporary_gospel	Henry_James,_1st_Baron...
riddle_master	patricia_a__mckillip	The_Riddle-Master_of_Hed	Patricia_A._McKillip
	1st cand.	-	Patricia_A._McKillip
	2nd cand.	-	-
	3rd cand.	-	-
king_john	shakespeare	King_John_(play)	William_Shakespeare
	1st cand.	John,_King_of_England	William_Shakespeare
	2nd cand.	King_John_(play)	Shakespeare_quadrangle
	3rd cand.	King_John_(1899_film)	Shakespeare,_Ontario

Table 3. Four annotation examples of the `bookwriter` predicate (we have removed the URI prefix `http://dbpedia.org/resource/` for better readability).

mapping after a combined search in DBpedia, Wikipedia or other resources available on the Web.

(iii) The annotator cannot determine a DBpedia entity to which the given Nell instance should be mapped. This was the case when the term was too ambiguous, underspecified, or not represented in DBpedia. In this case the annotator marked the instance as unmatchable ('?').

Table 3 shows four possible annotation outcomes for the `bookwriter` predicate. The first example illustrates case (i) and (ii). The second example illustrates case (i) and (iii). With respect to this example the annotator could not determine the reference for the Nell term `gospel`. The third example illustrates a special case of (ii) where no mapping candidate has been generated for the Nell term `patricia_a__mckillip`. The fourth example shows that the top match generated by our algorithm is not always the correct mapping, but might also be among the other alternatives that have been generated.

5 Experiments

5.1 Evaluation Measures

In the following, we briefly re-visit the definitions of precision and recall and explain their application in our evaluation scenario. Let A refer to the mappings generated by our algorithm, and G refer to mappings in the gold standard. Precision is defined as $prec(A, G) = |A \cap G|/|A|$ and recall as $rec(A, G) = |A \cap G|/|G|$. The F_1 measure is the equally weighted harmonic mean of both values, i.e., $F_1(A, G) = 2 * prec(A, G) * rec(A, G)/(prec(A, G) + rec(A, G))$.

If an annotator assigned a question mark, then the corresponding Nell term could not be mapped and it does not appear in the gold standard G. This can again be seen in Table 3, where we present the mappings generated by

Predicate	Unmatched	Reason/Observation
agentcollaborateswithagent	15%	only first names or surnames are given
lakeinstate	14%	non-existent entities for the lakes in context
personleadsorganization	9.5%	non-existent entities or too ambiguous
bookwriter	8.5%	obscure books, writer ambiguous
animalistypeofanimal	8%	uncommon description of animal types
teamplaysagainstteam	6.5%	ambiguity between college and college team
companyasloknownas	5.5%	ambiguous names
weaponmadeincountry	3.5%	non-existent entities in DBpedia
actorstarredinmovie	3%	ambiguity between film, acts, or play
bankbankincountry	2%	ambiguous and non-existent entities
citylocatedinstate	0.5%	too general entity
athleteledsportsteam	0%	well defined names of persons

Table 4. The percentage of entities per predicate that could not be matched by a human annotator.

our algorithm for four triples, as well as the corresponding gold-standard annotations. If the mapping A consists of top-k possible candidates, computing precision and recall on the examples, we have the precision value for $k = 1$ as $prec@1 = 4/7 \approx 57\%$ and $rec@1 = 4/7 \approx 57\%$. Note that precision and recall are not the same in general, because $|A| \neq |G|$ in most cases. More generally, we are interested in $prec@k$, the fraction of top-k candidates that are correctly mapped and $rec@k$, the fraction of correct mappings that are in the top-k candidates. For $k = 3$, we have $prec@3 = 5/17 \approx 29\%$ and $rec@3 = 5/7 \approx 71\%$.

It can be expected that $prec@1$ will have the highest score and $rec@1$ will have the lowest score. When we analyze A with $k > 1$, we focus mainly on the increase in recall. Here we are in particular interested in the value of k for which the number of additionally generated correct mappings in A is negligibly small compared to the mappings generated in A for $k + 1$.

When generating the gold standard, we realized that finding the correct mappings is often a hard task and sometimes even difficult for a human annotator. We had also observed that the problem of determining the gold standard varies strongly across the properties we analyzed. For some of the properties we could match all (or nearly all) subjects and objects in the chosen triples, while for other properties up to 15% of the instances could not be matched.

Table 4 presents the percentage of entities that could not be matched by the annotators, together with the main reason the annotators provided when they could not find a corresponding entity in DBpedia. A typical example of a problematic triple from the `agentcollaborateswithagent` property is

```
agentcollaborateswithagent(world, greg)
```

In this case, the mapping for subject and object was annotated with a question mark. We also observed cases in which an uncommon description was chosen that had no counterpart in DBpedia. Some examples from the predicate `animalistypeofanimal` are the labels `furbearers` or `small_mammals`.

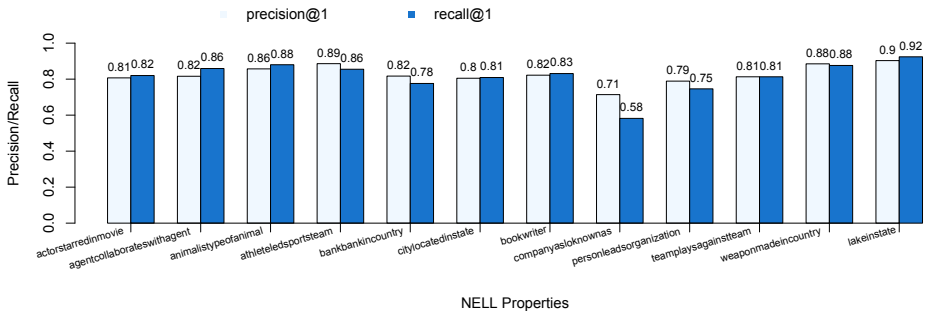

Fig. 1. *prec*@1 and *rec*@1 of our proposed method.

5.2 Results and Discussion

We run our algorithm against the gold standard[6], and report the precision and recall values. In Figure 1, we show the precision and recall values obtained on the set of NELL predicates. These values are for top-1 matches. Precision and recall vary across the predicates with `lakeinstate` having the highest precision. Using micro-average method, for the top-1 matches we achieved a precision of 82.78% and an average recall of 81.31% across all the predicates. In the case of macro-averaging, instead, we achieved precision of 82.61% and recall of 81.42%.

In Figure 2, we show the values for *rec*@2, *rec*@5 and *rec*@10 compared to *rec*@1, the recall values reported in Figure 1. By considering more possible candidates with increasing k, every term gets a better chance of being matched correctly, thus explaining the increases in *rec*@k with k. However, it must be noted, that for most of the predicates the values tend to saturate after *rec*@5. This reflects that after a certain k any further increase in k does not alter the correct mappings, since our algorithm already provided a match within top-1 or top-2 candidates. Still, for some we observe an increase even at *rec*@10 because there can be still a possibility of one correct matching candidate lying at a much lower rank in the top-k list of candidates.

In Figure 3, we plot the micro-average values of the precision, recall and F_1 scores over varying k. We attain the best F_1 score of 0.82 for $k = 1$ and the recall values tend to saturate after $k = 5$.

This raises an important question regarding the upper bound of recall of our algorithm. In practice, we cannot achieve a recall of 1.0 because we are limited from factors like:

– the matching candidate being never referred to by the terms. For example, `gs` refers to the company *Goldmann Sachs*, but it never appeared even in all the possible candidates, since *Goldmann Sachs* is never referred to with `gs` in Wikipedia.

[6] The data are freely available at `http://web.informatik.uni-mannheim.de/data/nell-dbpedia/NellGoldStandard.tar`.

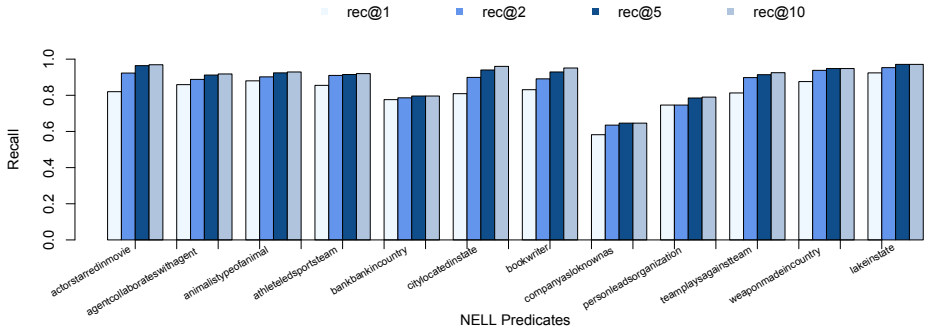

Fig. 2. Comparison of $rec@1$ against $rec@k$.

– persons being often referred to by the combination of their middle and last name. For e.g. `hussein_obama`. It is actually talking about President Barack Obama, but with our approach we cannot find a good match.
– misspelled words. We have entities like `missle` instead of `missile`.

However, there are ways to further improve the recall of our method like, for instance, by means of string similarity techniques – e.g., Levenshtein edit distance. A similarity threshold (say, as high as 95%) could then be tuned to consider entities which only partially match a given term. Another alternative would be to look for sub-string matches for the terms with middle and last names of persons. For instance, `hussein_obama` can have a possible match if terms like `barrack_hussein_obama` has a candidate match. In addition, a similarity threshold can be introduced in order to avoid matching by arbitrary longer terms.

In general, thanks to the annotation task and our experiments we were able to acquire some useful insights about the data set and the proposed task.

– Predicates with polysemous entities, like `companyalso knownas`, usually have lower precision. The triples for this predicate had a wide usage of abbreviated terms (the stock exchange codes for the companies) and that accounts for a lower precision value.
– The NELL data is skewed towards a particular region or type. The triples involving

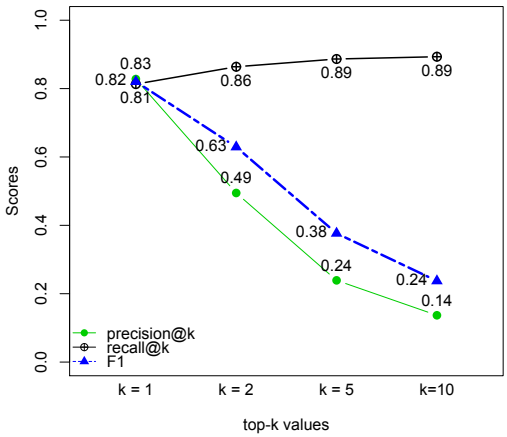

Fig. 3. Micro-average $prec@k$, $rec@k$ and F_1.

persons and sports primarily refer to basketball or baseball. Similarly, for `lakeinstate`, nearly all the triples refer to lakes in United States.

6 Related Work

Key contributions in information extraction have concentrated on minimizing the amount of human supervision required in the knowledge harvesting process. To this end, much work has explored unsupervised bootstrapping for a variety of tasks, including the acquisition of binary relations [3], facts [8], semantic class attributes and instances [20]. Open Information Extraction further focused on approaches that do not need any manually-labeled data [9], however, the output of these systems still needs to be disambiguated by linking it to entities and relations from a knowledge base. Recent work has extensively explored the usage of distant supervision for IE, namely by harvesting sentences containing concepts whose relation is known and leveraging these sentences as training data for supervised extractors [27, 14]. Talking of integration of open and closed IE projects, it is worthwhile to mention the work of [21] where matrix factorization technique was employed for extracting relations across different domains. They proposed an *universal schema* which supports cross domain integration.

There has been some work on instance matching in the recent past. Researchers have transformed the task into a binary classification problem and solved it with machine learning techniques [22]. Some have tried to enrich unstructured data in form of text with Wikipedia entities [18]. However, in our approach we consider the context of the entities while creating the gold standard which makes it bit different from these above mentioned entity linking approaches. Also, there are tools like *Tìpalo* [12] for automatic typing of DBPEDIA entities. They use language definitions from Wikipedia abstracts and use WordNet in the background for disambiguation. PARIS [24] takes a probabilistic approach to align ontologies utilizes the interdependence of instances and schema to compute probabilities for the instance matches. Lin et al. [16] provide a novel approach to link entities across million documents. They take web extracted facts and link the entities to Wikipedia by means of information from Wikipedia itself, as well as additional features like string similarity, and most importantly context information of the extracted facts. The Silk framework [26] discovers missing links between entities across linked data sources by employing similarity metrics between pairs of instances.

7 Conclusions

In this paper, we introduced a most-frequent-entity baseline algorithm in order to link entities from an open domain system to a closed one. We introduced a gold standard for this task and compared our baseline against it. In the near future, we plan to extend this work with more complex and robust methods, as well as extending our methodology to cover other open IE projects like ReVerb.

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ive. Dbpedia: A nucleus for a web of open data. In *Proceedings of 6th International Semantic Web Conference joint with 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, pages 722–735, 2007.
2. Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia – A crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.
3. Sergey Brin. Extracting patterns and relations from the world wide web. In *Selected papers from the International Workshop on The World Wide Web and Databases*, WebDB '98, pages 172–183. Springer-Verlag, 1999.
4. Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL-06*, pages 9–16, 2006.
5. Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proc. of AAAI-10*, pages 1306–1313, 2010.
6. Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 1st edition, 2010.
7. Oren Etzioni. Search needs a shake-up. *Nature*, 476(7358):25–26, 2011.
8. Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in KnowItAll: (preliminary results). In *Proc. of WWW '04*, pages 100–110, 2004.
9. Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proc. of EMNLP-11*, pages 1535–1545, 2011.
10. Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proc. of AAAI-06*, pages 1301–1306, 2006.
11. Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI-07*, pages 1606–1611, 2007.
12. Aldo Gangemi, AndreaGiovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. Automatic typing of DBpedia entities. In *The Semantic Web – ISWC 2012*, volume 7649 of *Lecture Notes in Computer Science*, pages 65–81. Springer, Berlin and Heidelberg, 2012.
13. Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2012.
14. Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. Learning 5000 relational extractors. In *Proc. of ACL-10*, pages 286–295, 2010.
15. Eduard Hovy, Roberto Navigli, and Simone Paolo Ponzetto. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27, 2013.
16. Thomas Lin, Mausam, and Oren Etzioni. Entity linking at web scale. In *Proc. of AKBC-WEKEX '12*, pages 84–88, 2012.
17. Pablo Mendes, Max Jakob, and Christian Bizer. Dbpedia: A multilingual cross-domain knowledge base. In *Proc. of LREC-12*, 2012.
18. David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proc. of CIKM '08*, pages 509–518, 2008.

19. Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.
20. Marius Paşca. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In *Proc. of WWW '07*, pages 101–110, 2007.
21. Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, 2013.
22. Shu Rong, Xing Niu, EvanWei Xiang, Haofen Wang, Qiang Yang, and Yong Yu. A machine learning approach for instance matching based on similarity metrics. In *The Semantic Web – ISWC 2012*, volume 7649 of *Lecture Notes in Computer Science*, pages 460–475. Springer, Berlin and Heidelberg, 2012.
23. Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *Proc of LREC-12*, pages 3168–3175, 2012.
24. Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168, November 2011.
25. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA, 2007. ACM Press.
26. Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk - A Link Discovery Framework for the Web of Data. In *Proc. of LDOW '09*, 2009.
27. Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In *Proc. of ACL-10*, pages 118–127, 2010.