Integration of Large Scale Knowledge Bases using Probabilistic Graphical Models

Arnab Kumar Dutta
Data and Web Science
Universität Mannheim
68159 Mannheim, Germany
arnab@informatik.uni-mannheim.de

ABSTRACT

Over the recent past, information extraction (IE) systems such as NELL and REVERB have attained much success in creating large knowledge resources with minimal supervision. But, these resources in general, lack schema information and contain facts with high degree of ambiguity which are often difficult to interpret. Whereas, Wikipedia-based IE projects like DBPEDIA and YAGO are structured, have disambiguated facts with unique identifiers and maintain a well-defined schema. In this work, we propose a probabilistic method to integrate these two types of IE projects where the structured knowledge bases benefit from the wide coverage of the semi-supervised IE projects and the latter benefits from the schema information of the former.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: [Representations (procedural and rule-based)]; I.2.3 [**Deduction and Theorem Proving**]: [Uncertainty, fuzzy, and probabilistic reasoning]

Keywords

Data Integration, Probabilistic Inference, Knowledge Bases

1. INTRODUCTION

1.1 Motivation

Research in information extraction (IE) systems has experienced a strong momentum in recent years. While Wikipedia based information extraction projects such as DBPEDIA [1, 16] and YAGO [23] 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) [2] 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 [8]. In this context, there are latent advantages in integrating these two IE systems in producing better knowledge repositories which can be harvested to further enable state-of-the-art performance on a wide range of NLP applications. This is especially promising, since NELL and REVERB typically achieve a very large coverage, but still lack a full-fledged, clean ontological structure which, on the other hand, could be provided by large-scale ontologies like DBPEDIA or YAGO.

WSDM'14, February 24–28, 2014, New York, New York, USA.

http://dx.doi.org/10.1145/2556195.2556202.

1.2 Problem Statement

In order to integrate existing IE projects, we have to solve the problem of entity resolution. The fact that makes this task challenging is that triples from OIE systems are often underspecified and ambiguous. For instance, consider a NELL triple, where two terms (subject and object) are linked with some relationship (predicate):

```
agentcollaborateswithagent(times, bob herbert)
```

Here, `times` and `bob herbert` are two terms which are linked by a predicate called `agentcollaborateswithagent`. The problem of interpreting the terms is difficult since the term `times` by itself can have several meanings. Interpreting this sentence's meaning poses a challenging task even to humans. Here, `times` refers to the newspaper company *The New York Times* and `bob herbert` refers to the journalist *Bob Herbert*.

Furthermore, the task of integration gets more complicated as the triples extracted from the web are often not certain. OIE systems attach a *degree of truth* to the facts. Even, if the disambiguated terms refer to the correct entities in the given context, it is difficult to determine the correctness of the triple itself. This becomes evident with the following example,

```
0.86:  agentcollaborateswithagent(fox, obama)
```

In this given context, the triple is of significantly low confidence and even if `fox` refers to the *news corporation* and `obama` to *Barack Obama*, the certainty of the fact is questionable. Thus, the free text terms within OIE facts can have multiple meanings and varying confidences across different contexts. In this work, we take some steps towards overcoming this problem by providing a probabilistic approach for the integration of different large-scale knowledge bases.

Anchor	Article	Link Counts
carbon	Carbon	2256
carbon	Carbon County, Pennsylvania	120
...	...	...
lincoln	Lincoln, England	1844
lincoln	Lincoln (2012 film)	496
...	...	...

Table 1: Snippet of articles linked to using the anchors *carbon* and *lincoln*.

2. PROPOSED SOLUTION

We propose a three step approach towards a solution of the problem. In Section 2.1 we discuss a way to create probabilistic set of `sameAs` hypotheses between instances from NELL and DBPEDIA. In Section 2.2 we formulate the problem using Markov Logic Networks [19], which is an efficient way of combining logic with uncertainty. Finally, in Section 2.3 we filter out incorrect `sameAs` hypotheses, if any, by solving an inference task on a Markov network created from our markov logic formulation. In the process, we want to find an answer to the question if it is possible with probabilistic reasoning to exploit the inconsistencies in linking entities occurring in web-extracted facts to improve the integration task. Note that, although NELL has a schema of its own, we do not exploit it keeping in mind a generalized approach towards the problem, since other OIE projects, like REVERB lack such a schema.

2.1 Linking terms

In this work we use Wikipedia as an entity-tagged corpus [4] in order to bridge knowledge encoded in NELL with DBPEDIA. It contains information about each article, anchors present, outgoing links etc. Note that, there are alternative data sets such as the CROSSWIKI data [21], but we opted instead for exploiting only Wikipedia internal-link anchors, which are expected to be a cleaner source of data. Since there is a corresponding DBPEDIA instance for each Wikipedia article [3], we can safely assume an article to be an exact counterpart of the DBPEDIA instance. However, the problem is that, due to polysemy, often a term from NELL can refer to several different articles in Wikipedia or, analogously, instances in DBPEDIA.

Table 1 shows two examples, where a particular anchor can refer to multiple Wikipedia articles with varying number of link counts[1]. This fosters the idea to adopt a probabilistic approach in selecting the best possible DBPEDIA resource. 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 [21]. Formally, if 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 a ranked list of $P(A_j|e)$, top-k candidates are selected.

We apply this procedure on every term occurring as subject and object in a NELL triple. These terms are analogous to the anchors in Wikipedia. We generate a set of `sameAs` hypotheses between the NELL terms and the most probable DBPEDIA instance with confidences given by P. For further details and empirical results, refer to the paper [7].

[1]We used WikiPrep [13, 12] for parsing the Wikipedia dumps and finding the link counts for anchor-article pairs.

Axiom Type	Atom	Example of a Ground Atom
$\mathtt{A} \sqsubseteq \mathtt{B}$	$\mathtt{csub}(\mathtt{A},\mathtt{B})$	*csub(Student,Person)*
$\mathtt{A} \sqsubseteq \neg\mathtt{B}$	$\mathtt{cdis}(\mathtt{A},\mathtt{B})$	*cdis(Vehicle,Actor)*
$\mathtt{P} \sqsubseteq \mathtt{Q}$	$\mathtt{psub}(\mathtt{P},\mathtt{Q})$	*psub(husbandof, hasspouse)*
$\mathtt{a} : \mathtt{C}$	$\mathtt{isType}(\mathtt{C},\mathtt{a})$	*isType(President, Obama)*
$\mathtt{a} \neq \mathtt{b}$	$\mathtt{diffFrom}(\mathtt{a},\mathtt{b})$	*diffFrom(Obama, Bush)*
$\exists\mathtt{R}.\top \sqsubseteq \mathtt{A}$	$\mathtt{dom}(\mathtt{R},\mathtt{A})$	*dom(lakeinstate, BodyOfWater)*
$\top \sqsubseteq \forall\mathtt{R}.\mathtt{A}$	$\mathtt{ran}(\mathtt{R},\mathtt{A})$	*ran(lakeinstate, Place)*

Table 2: *unweighted atomic formulae* used for the modeling.

2.2 Modeling with Markov Logic Networks

For the integration task, we create a knowledge base $\mathcal{K}$, consisting of two types of information. An *uncertain* type consisting of `sameAs` hypotheses described in the previous section. The other being the *certain* type which encodes the information from the *closed* domain IE, DBPEDIA in particular. In our case, we employ Markov Logic Network (MLN) which provides the perfect template to model scenarios consisting of certain and uncertain aspects, as present in our knowledge base.

Unweighted Atoms

We translate the axioms of the DBPEDIA ontology[2] to *atomic formulae* or *atoms*[3] as indicated by Table 2. These atoms are defined for both the terminological axioms ($\mathcal{T}$-box) and the assertions involving instances ($\mathcal{A}$-box) in $\mathcal{K}$. Since these atoms model the information from DBPEDIA, we want them to hold under all circumstances, hence they are unweighted. All possible instantiations of the atoms lead to *ground atoms* or specifically *hard evidences*. Examples of such hard evidences are shown in Table 2. The atoms are formulated using the ontology provided by DBPEDIA with the exception of class disjointness. Particularly, atoms of the form `cdis(A, B)` were taken from the works of [11], where disjointness axioms were learnt automatically.

Weighted Atoms

Weighted atomic formulae represent the *uncertain* information in $\mathcal{K}$. In our linking task, we create a set of `sameAs` hypotheses having confidence values attached. Using MLN, we define weighted `sameAs` atoms as

$$w : \mathtt{sameAs}(\mathtt{a},\mathtt{b})$$

where, w are the confidences as computed in Section 2.1. Instantiating the atoms with a set of constants leads to grounding them and creates *soft evidences*, for instance `0.498 : sameAs(Special Air Service, sas)`.

[2]http://downloads.dbpedia.org/3.8/dbpedia_3.8.owl.bz2

[3]Classes are denoted with A, B, C, D, properties with R, S and instances with a, b, c.

The property assertions from OIE can also be modeled in a similar fashion as

$$w : \texttt{pAss}(\texttt{P}, \texttt{a}, \texttt{b})$$

where, w are the confidence values obtained from the web extracts of the OIE projects. Grounding these type of atoms leads to the set of soft evidences like
`0.967 : pAss(bookwriter, pickwick papers, dickens)`.
Note that every ground atom, is binary valued.

Unweighted Rules

The atoms, in disjunction with each other, enable us to formulate a set of first order logic *rules*. Generally, a knowledge base is a conjunction of all such rules. Hence, $\mathcal{K}$ can have an exponential number of states with every possible combination of a boolean value assignment to the ground atoms. Every such state is a *world*. A possible world is necessarily a truth assignment to the ground atoms.

Unweighted rules are the *hard* constraints which should hold in every world. If they are violated, the world is improbable. By translating the DBpedia axioms into atoms, we loose the semantics associated with the Description Logic (DL) axioms in DBpedia. Hence, there is a need to add explicitly deduction rules to exploit the inference mechanisms implicit in the DL semantics.

$$\Rightarrow \texttt{csub}(\texttt{C}, \texttt{D}) \tag{1a}$$
$$\Rightarrow !\texttt{cdis}(\texttt{C}, \texttt{C}) \tag{1b}$$
$$\texttt{csub}(\texttt{B}, \texttt{C}) \wedge \texttt{csub}(\texttt{C}, \texttt{D}) \Rightarrow \texttt{csub}(\texttt{B}, \texttt{D}) \tag{1c}$$
$$\texttt{csub}(\texttt{A}, \texttt{B}) \wedge \texttt{cdis}(\texttt{B}, \texttt{C}) \Rightarrow \texttt{cdis}(\texttt{A}, \texttt{C}) \tag{1d}$$
$$\texttt{dom}(\texttt{R}, \texttt{C}) \wedge \texttt{csub}(\texttt{C}, \texttt{B}) \Rightarrow \texttt{dom}(\texttt{R}, \texttt{B}) \tag{1e}$$
$$\texttt{ran}(\texttt{R}, \texttt{C}) \wedge \texttt{csub}(\texttt{C}, \texttt{B}) \Rightarrow \texttt{ran}(\texttt{R}, \texttt{B}) \tag{1f}$$

These rules are a subset of the rules used in [10]. We further extend our rule set with support for inference with the $\mathcal{A}$-box.

$$\texttt{csub}(\texttt{C}, \texttt{B}) \wedge \texttt{isType}(\texttt{C}, \texttt{a}) \Rightarrow \texttt{isType}(\texttt{B}, \texttt{a}) \tag{2a}$$
$$\texttt{pAss}(\texttt{R}, \texttt{a}, \texttt{b}) \wedge \texttt{psub}(\texttt{R}, \texttt{S}) \Rightarrow \texttt{pAss}(\texttt{S}, \texttt{a}, \texttt{b}) \tag{2b}$$
$$\texttt{pAss}(\texttt{R}, \texttt{a}, \texttt{b}) \wedge \texttt{dom}(\texttt{R}, \texttt{C}) \Rightarrow \texttt{isType}(\texttt{C}, \texttt{a}) \tag{2c}$$
$$\texttt{pAss}(\texttt{R}, \texttt{a}, \texttt{b}) \wedge \texttt{ran}(\texttt{R}, \texttt{C}) \Rightarrow \texttt{isType}(\texttt{C}, \texttt{b}) \tag{2d}$$

So far we have defined rules required to capture the general behavior of the $\mathcal{A}$-box and the $\mathcal{T}$-box. But there needs to be a restriction which ensures that an incorrect hypothesis introduces an inconsistency into our knowledge base.

$$\Rightarrow \texttt{sameAs}(\texttt{a}, \texttt{a}) \tag{3a}$$
$$\texttt{sameAs}(\texttt{a}, \texttt{b}) \Rightarrow \texttt{sameAs}(\texttt{b}, \texttt{a}) \tag{3b}$$
$$\texttt{sameAs}(\texttt{a}, \texttt{b}) \wedge \texttt{sameAs}(\texttt{b}, \texttt{c}) \Rightarrow \texttt{sameAs}(\texttt{a}, \texttt{c}) \tag{3c}$$
$$\texttt{isType}(\texttt{C}, \texttt{a}) \wedge \texttt{sameAs}(\texttt{a}, \texttt{b}) \Rightarrow \texttt{isType}(\texttt{C}, \texttt{b}) \tag{3d}$$
$$\texttt{cdis}(\texttt{C}, \texttt{B}) \wedge \texttt{isType}(\texttt{C}, \texttt{a}) \Rightarrow !\texttt{isType}(\texttt{B}, \texttt{a}) \tag{3e}$$
$$\texttt{diffFrom}(\texttt{a}, \texttt{b}) \Rightarrow !\texttt{sameAs}(\texttt{a}, \texttt{b}) \tag{3f}$$

(3e) enforces a conflict into the modeling by stating that if two classes are disjoint with each other they cannot be the type of the same instance. (3f) limits the possibility for an entity representing two different real world instances.

Weighted Rules

These are *soft* rules, which can be violated but if they are, that world becomes less probable compared to the world where they hold true. In our modeling, we have not used any soft rules. This remains as an open issue which is discussed further in Section 4.

2.3 Inference

Here, we discuss the idea behind solving the problem as an inference task in MLN. In particular, we are interested in the maximum a-posteriori (MAP) state. The aim is to find the most probable world. Usually, we observe the state of some variables and try to infer the state of the hidden variables. Let, E be the observed variables and Q be the set of hidden variables, then a MAP state tries to find an assignment for Q so that $\underset{Q}{\text{argmax}}\, P(Q|E)$, i.e. the assignment which makes P maximum. Such an assignment is the most probable world.

In our problem scenario, only the evidences containing `pAss` and `sameAs` are hidden. The rule set in Section 2.2 provides a template for creating a Markov Network where each node represents a ground atom or an evidence. Intuitively, a possible world is a truth value assignment to every evidence without any hard rules being violated. Whenever an evidence involving a `sameAs` is in a conflict with others, it makes that world less likely and it gets penalized by the weight w associated with it. There can be multiple such possible worlds, but we are interested in the world with maximal sum of the weights of the weighted evidences. This world is called the *most probable* world which contains a subset of the initial `sameAs` hypotheses.

3. EMPIRICAL RESULTS

In the following we report about the experiments we conducted so far, focusing primarily on the `sameAs` hypotheses. In Section 3.1 we analyze our approach described in Section 2.1 in generating good mapping hypotheses among the $top-k$ proposals. These proposals will finally be the input to our Markov Logic based method and serve as a baseline for comparison as well. It is thus important to understand the characteristics of our method for generating linking proposals in details. In Section 3.2 we report on first results for applying the ML based approach in a simple setting. The results of these experiments give a first impression of the benefits of our approach.

3.1 Linking Entities

For the task of linking, we sample a set of Nell properties having atleast 100 instances each. We annotated them to come up with a gold standard for which the data set is publicly available[4]. We ran our baseline algorithm against the gold standard. In Figure 1, we show the *precision*@1 and *recall*@1 values obtained on this set of Nell predicates. 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%.

These results indicate that there is significant room for improvement. Our extended experiments showed a steady

[4]http://web.informatik.uni-mannheim.de/data/nell-dbpedia/NellGoldStandard.tar

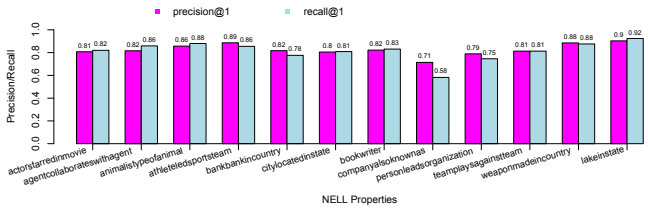

Figure 1: *precision*@1 and *recall*@1 of the baseline method.

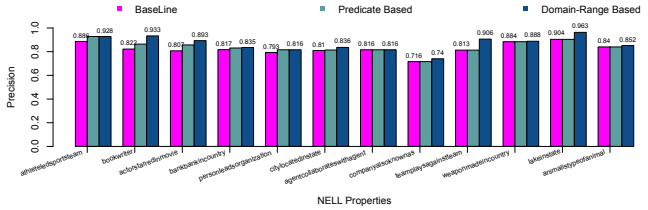

Figure 2: precision values comparison of baseline method and MLN based approach.

rise in recall for $top-2$ and beyond with a saturation after $top-5$ ([7]). This denotes that there are good candidate matches within $top-5$ and with probabilistic reasoning we can influence the precision and recall in such cases where the $top-1$ is not the best candidate choice. Intuitively, in other cases, there is nothing to improve.

3.2 MLN Based Approach

Initially, we adopt a naive way of manually mapping the NELL properties to a DBPEDIA property. The rationale is to link properties with a subsumption axiom. We add hard evidences of the form `psub(bookwriter, author)`. For some properties, like `actorstarredinmovie`, we had to extend our rule set with the addition of `invpsub` to capture their behavior (similar to rule 2b). We observed, there are few properties for which a DBPEDIA counterpart was not found.

Hence, we extended our experiments with a different approach. We manually specified the *domain* and *range* restrictions for the NELL properties in terms of DBPEDIA classes. Table 3 gives the collective details of the manual mappings for the set of NELL properties; analogous DBPEDIA properties it is in subsumption with and the domain/range values used. Figure 2 reports our first results using MLN. It shows a comparison of the two above mentioned approaches with the baseline. Note that properties which were not possible to be mapped initially, had considerable improvements with the domain and range based mapping. We used Rockit [18] for computing the MAP state for our inference task.

In particular, even though manual methods showed some improvements, we need to develop automated techniques to generate such property mappings and domain/range restrictions in order to make our approach applicable to the complete data set from NELL and REVERB as well.

4. RESEARCH ISSUES

Distant Supervision: However, we intend to incorporate semi-supervised techniques to automatically learn such restrictions between NELL and DBPEDIA. For a given NELL property, we can sample a certain number of triples and for each one of them we can find an analogous DBPEDIA triple with the similar subject-object combination. This can give a probability distribution of *domain* and *range* of the NELL property over a set of possible DBPEDIA triples. For instance, NELL property *bookwriter* has domain as DBPEDIA concept *Book* with 56% probability and *Work* with 78% probability. Similar values can also be computed for range.

Nell Property	DBpedia Property	Domain	Range
bookwriter	author	WrittenWork	Person
athleteledsportsteam	formerTeam	Athlete	SportsTeam
bankbankincountry	regionServed	Organisation	Country
citylocatedinstate	location	City	Place
personleads organization	employer	Person	Organisation
actorstarredinmovie	starring*	Person	Film
agentcollaborates withagent	-	Agent	Agent
animalistypeof animal	-	Animal	Animal
teamplaysagainst team	-	SportsTeam	SportsTeam
companyalso knownas	-	Organisation	Organisation
weaponmade incountry	-	Device	Country
lakeinstate	-	BodyOfWater	Place

Table 3: Property Mappings of Nell to DBpedia.(* : inverse property subsumption)

Weight Learning: It is important to weigh the evidences we create between NELL and DBPEDIA. For instance, `psub` discussed in Section 3.2. In an ideal setting, we cannot claim a strict property subsumption restriction between properties across different domains. The choice of the weight is critical otherwise, it can lead to a degraded model. Our initial experiments, lead to deteriorated results while trying to set weights manually to such rules. This motivates us, to adopt an efficient weight learning strategy. There have been works on efficient weight learning strategies [15], which are expected to work better than pseudo likelihood, as was proposed in [19].

Similar Type but Incorrect Mapping: There are two broad types of the problem we aim to address. Firstly, a scenario where a particular assignment of the `sameAs` statement is wrong and it has incorrect type information. For instance, the term *jaguar* in some NELL triple refers to the animal but it should have been the car company. And, secondly there can be a scenario where an assignment is wrong but the type is correct. For instance, a term like *steve* in some triple refers to *SteveNash*, the basketball player and not *SteveJobs*, the ex-CEO of Apple. Both are of type persons in this case and we need a way to deal with such scenarios. Essentially, every entity involved in some relation, has some inherent characteristics. For e.g. entities involved in a *hasSpouse* relationship cannot have age-difference more than 100 years, *bookwriter* cannot have author's birth year later than the publication date of the book, and so on. We had exploited this characteristics for a small set of OIE properties and efficiently employed non-parametric ways of estimating the probability density function of the variables. We used Kernel Density Estimation (KDE) to rank enti-

ties based on their estimation function values. However, the biggest challenge still is to discover such features we intend to use in an automated setting.

5. RELATED WORK

OIE 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 [24, 14]. There has been closely related work in entity linking in the past. Machine learning based approach try to indicate equivalent instances that refer to the same real-world object [20]. Our approach closely relates to the work using linking features such as string matching. Extraction frameworks like NELL, TextRunner [2], are of prime importance in providing massive data sets of web-extracted statements. PARIS [22] takes a probabilistic approach to align ontologies. This work is aimed towards ontology alignment and utilizes the interdependence of instances and schema to compute probabilities for the instance matches. Please note the important distinction between entity resolution in text and integrating triples from information extraction projects. The latter can explicitly take into account the semantics of the relations that links subject and object in each of the triples.

Reasoning in general, on large scale data was applied in enriching Yago [17]. But this used MaxSat-based constraint reasoning and did not compute MAP state. There have been works in disambiguating entities using MLN [6] but they use much of the background knowledge of entities in coming up with their disambiguation formulae.

6. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ive. Dbpedia: A nucleus for a web of open data. In *Proceedings of ISWC+ASWC 2007*, pages 722–735, 2007.

[2] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia – A crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[4] R. Bunescu and M. Paşca. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL-06*, pages 9–16, 2006.

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proc. of AAAI-10*, pages 1306–1313, 2010.

[6] H.-J. Dai, R. T.-H. Tsai, and W.-L. Hsu. Entity disambiguation using a markov-logic network. In *Proceedings of 5th IJCNLP*, pages 846–855, 2011.

[7] A. Dutta, M. Niepert, C. Meilicke, and S. P. Ponzetto. Integrating Open and Closed Information Extraction : Challenges and First Steps. *Proceedings of the NLP and DBpedia workshop ISWC*, 1064, 2013.

[8] O. Etzioni. Search needs a shake-up. *Nature*, 476(7358):25–26, 2011.

[9] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proc. of EMNLP-11*, pages 1535–1545, 2011.

[10] D. Fleischhacker, C. Meilicke, J. Völker, and M. Niepert. Computing incoherence explanations for learned ontologies. In *RR*, pages 80–94. Springer, 2013.

[11] D. Fleischhacker and J. Völker. Inductive learning of disjointness axioms. In *On the Move to Meaningful Internet Systems: OTM 2011*, volume 7045, pages 680–697. Springer, 2011.

[12] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proc. of AAAI-06*, pages 1301–1306, 2006.

[13] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI-07*, pages 1606–1611, 2007.

[14] R. Hoffmann, C. Zhang, and D. S. Weld. Learning 5000 relational extractors. In *Proc. of ACL-10*, pages 286–295, 2010.

[15] D. Lowd and P. Domingos. Efficient weight learning for markov logic networks. In *Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 200–211, 2007.

[16] P. Mendes, M. Jakob, and C. Bizer. Dbpedia: A multilingual cross-domain knowledge base. In *Proc. of LREC-12*, 2012.

[17] N. Nakashole, M. Theobald, and G. Weikum. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the Fourth ACM WSDM*, WSDM '11, pages 227–236. ACM, 2011.

[18] J. Noessner, M. Niepert, and H. Stuckenschmidt. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *AAAI*, 2013.

[19] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[20] S. Rong, X. Niu, E. Xiang, H. Wang, Q. Yang, and Y. Yu. A machine learning approach for instance matching based on similarity metrics. In *The Semantic Web – ISWC 2012*, volume 7649, pages 460–475. Springer-Verlag, 2012.

[21] V. I. Spitkovsky and A. X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *Proc of LREC-12*, pages 3168–3175, 2012.

[22] F. M. Suchanek, S. Abiteboul, and P. Senellart. Paris: probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168, 2011.

[23] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW 2007*, New York, NY, USA, 2007. ACM Press.

[24] F. Wu and D. S. Weld. Open information extraction using wikipedia. In *Proc. of ACL-10*, pages 118–127, 2010.