From DBpedia to Wikipedia: Filling the Gap by Discovering Wikipedia Conventions

Diego Torres, Pascal Molli, Hala Skaf-Molli, Alicia Diaz

From DBpedia to Wikipedia: Filling the Gap by Discovering Wikipedia Conventions

Diego Torres* and Pascal Molli†
*LIFIA - Facultad de Informática - UNLP,
La Plata, Argentina
Email: {diego.torres, alicia.diaz}@lifia.info.unlp.edu.ar

Hala Skaf-Molli† and Alicia Diaz*
†LINA, Nantes University
Nantes, France
Email: {Pascal.Molli, Hala.Skaf}@univ-nantes.fr

***Abstract*—Many relations existing in DBpedia are missing in Wikipedia yielding up an information gap between the semantic web and the social web. Inserting these missing relations requires to automatically discover Wikipedia conventions. From pairs linked by a property *p* in DBpedia, we find path queries that link the same pairs in Wikipedia. We make the hypothesis that the shortest path query with maximal containment captures the Wikipedia convention for *p*. We computed missing links and conventions for different DBpedia queries. Next, we inserted some missing links according to computed conventions in Wikipedia and evaluated Wikipedians feedback. Nearly all contributions has been accepted. In this paper, we detail the path indexing algorithms, the results of evaluations and give some details about social feedback.**

***Keywords*-Wikipedia Conventions; DBpedia; Wikipedia**

I. Introduction

Semantic Web is growing fast and brings better search and navigability to the web. It is mainly built from metadata extracted from the social web. A good example is DBpedia [1], a knowledge base extracted from Wikipedia[1] infoboxes and wiki markups. DBpedia supports complex semantic queries such as "people born in cities". If we retrieve all people and their birth place in DBPedia, this query retrieves 119,097 $(city, person)$ pairs. However, if we try to navigate from cities to people in Wikipedia, we can only get 65,200 pairs using different paths. Is it normal or an oversight? More generally, is there an information gap between DBPedia and Wikipedia? if yes, how much? can we fix it? In order to answer these questions, we tried to discover how a DBPedia relation is represented in Wikipedia i.e. Wikipedians conventions are used to represent such relations. With this knowledge, we will able to insert missing links according to Wikipedia conventions and evaluate Wikipedians feedback.

Discovering Wikipedia conventions for a DBPedia relation is a challenger. For example, for the "birthPlace" relation, many different kinds of paths are used to link the 65200 pairs. But, which ones best represent the "birthPlace" relation in Wikipedia? For a given relation in DBPedia, we transform paths to navigate between corresponding linked pairs in Wikipedia into path queries. Next, we make the hypothesis that the shortest path query with maximal containment is the best representation of the DBPedia relation in Wikipedia. We evaluated over six representative DBpedia queries. Next, we inserted manually in Wikipedia some missing links following the discovered conventions. Nearly all Wikipedia contributions has been accepted confirming the gap hypothesis. Rejected ones demonstrate some flaws in convention detection.

This paper is structured as follows: section II describes the context and motivating examples. Section III details the path indexing approach and PIA algorithms. Evaluations and validating results are detailed in Section IV. Section V presents related works. Finally, conclusion and further work are presented in section VI.

II. Motivating Example

```
PREFIX o:<http://dbpedia.org/ontology/>
PREFIX p:<http://dbpedia.org/property/>

SELECT ?city, ?person WHERE{
?person a o:Person.
?city a o:City.
?person p:birthPlace ?city}
```

Listing 1. retrieves all people and their birth place city

DBPedia has properties that are missing from Wikipedia. For instance, there is a link relating Boston entity with the Robin Moore entity by the *is_birthPlace_of* property. However, in Wikipedia, it is not possible to navigate from the Boston page to the Robin Moore page[2] through wikilinks or categories. If we want to add this link, we have first to learn how Wikipedians represent the relation *is_birthPlace_of* among cities and people i.e. Wikipedians conventions.

One-to-many relationships are often represented using "list_of" pages or the category tree. Choosing between these two options is already part of conventions. Conventions can also be quite difficult to understand for end-users. For example, Conventions for updating the category tree is described in "Categorization"[3]:

> ... each categorized page should be placed in all of the most specific categories to which it logically belongs. This means that if a page belongs to a subcategory of C (or a subcategory of a subcategory of C, and so on) then it is not normally placed directly into C. For exceptions to this rule, see Non-diffusing subcategories below ...

[1] http://www.wikipedia.org

[2] The complete name is Boston,_Massachusetts. For simplicity, it is referred as Boston.

[3] http://en.wikipedia.org/wiki/Wikipedia:Categorization

Is "bithPlaceOf" a non-diffusing subcategory? After reading conventions and analyzing some pages, People born in a city is denoted by the subcategory called `Cat: People_from_<CityName>`. Thus, for the Paris page, the `Cat: People_from_Paris` is a subcategory of the category `Cat:Paris`. This common practice are generalized by a path like `Cat:<from>/Cat:People_from_<from>` which represents a path to navigate to a category and then continue to its subcategory `Cat:People_from`.

Category tree must be used carefully:

- Although categories in Wikipedia are organized in a hierarchy called the category tree, according to Suchanek et al. [2], "Wikipedia category hierarchy is barely useful for ontological purposes. for example, Zidane is in the category `Cat: Football in France`[4], but Zidane is a football player and not a football". Consequently, relations category/subcategory cannot be used as class/subclass relations.
- However, there are many paths connecting Cities with People, for example `Cat:Capitals_in_Europe/ Cat:Paris/Cat:People_from_Paris`. Having several alternatives, which one is the best representation for the *is_birthPlace_of* relation?

The main concern of this paper is to automatically compute the path in the category tree of Wikipedia that best represents a relation of DBpedia. The approach is to learn from Wikipedia usage how wikipedians usually express these relations based on similar cases.

III. Wikipedia Path Indexing

If we consider Wikipedia as a database the question is: *What is the query in Wikipedia that best approximates to the results obtained by a DBpedia query?* Wikipedia can be seen as a semi-structured database, therefore, it can be queried by path queries [3], [4], [5]. The question now is: *What is the path query in Wikipedia that best approximates to the result of a DBpedia query?* There could be many alternatives to represent a missing link because there could be many possible navigational paths; which one best fits the semantic property? Before developing our algorithms to choose the best one, we start by some preliminary definitions.

A. Preliminary Definitions

Let $\mathcal{Q}$ be a SPARQL query that has a triple pattern with a predicate p and the dataset $\mathcal{D}$. $\mathcal{Q}(\mathcal{D})$ [5] refers to the result of evaluating the query $\mathcal{Q}$ over $\mathcal{D}$

Definition 1. $\mathcal{Q}(\mathcal{D}) = \{(s_i, t_i) : s_i, t_i \in \mathcal{D} \wedge p(s_i, t_i)\}$

For each $\mathcal{Q}(\mathcal{D})$, there is a corresponding result set in Wikipedia $\mathcal{Q}(\mathcal{W})$. These result sets are directly calculated by mapping an entity name[6] in $\mathcal{Q}(\mathcal{D})$ to its equivalent page name in Wikipedia.

[4]We use the prefix `Cat:` to indicate a category. Wikipedia uses the word `Category` for that.

[5]For simplicity, we do not add all patterns matching of the SPARQL query, we keep only the pattern related to the semantic relation.

[6]Entity names in DBpedia have the prefix `http://dbpedia.org/resource/` followed by the Wikipedia page title.

Wikipedia can be formalized as a labeled graph $\mathcal{G}(\mathcal{W}, \mathcal{E})$, where $\mathcal{W}$ are nodes and $\mathcal{E}$ are edges. $\mathcal{W}$ is the set of pages of Wikipedia, pages can be both articles and categories. $\mathcal{E}$ are links between pages. A navigational path is a sequence of pages connected by links. Formally, a path between two pages can be defined by:

Definition 2. *A path $\mathcal{P}(p_1, p_n)$ between two Wikipedia pages p_1, $p_n \in \mathcal{W}$ is a sequence of pages $p_1/\ldots/p_n$, s.t. $\forall i\, p_i \in \mathcal{W} \wedge \forall i, j : 1 \leqslant i < j \leqslant n$, $p_i \neq p_j$, $\forall i : 1 \leqslant i \leqslant n-1$ where (p_i, p_{i+1}) is a link between p_i and p_{i+1}. p_1 and p_n are called the* source *page and the* target *page respectively.*

A path query is a generalization of similar paths. We adapted the path query definition given in [3] to the Wikipedia context.

Definition 3. *A Wikipedia path query (in short path query) $\mathcal{PQ}(s)$ is a regular expression over a finite alphabet Σ.*

$$\Sigma = \mathcal{W} \cup \{\#from, \#to\}$$

The words #from and #to enables us to define path variables. We have adapted the *general path query* definition given in [3] with path expression of the Lorel language [6] in order to generate the suitable regular expressions for the case of Wikipedia. The details of the regular expression generation is out of the scope of this paper.

Table I shows examples of the result set of query $\mathcal{Q}_1$ in the Listing 1, path and path queries.

Definition 4. *The answer of a path query $\mathcal{PQ}$ over a source page $s \in \mathcal{W}$ is the set of pages $t \in \mathcal{W}$ reachable from s by some paths labeled with a word from Σ; its result set can be defined as:*

$$\mathcal{R}(\mathcal{PQ}(s), \mathcal{W}) = \{(s, t_i) : s, t_i \in \mathcal{W} \wedge t_i \in \mathcal{PQ}(s)\}$$

The issue now is: given $\mathcal{Q}(\mathcal{D})$, for each $(s, t) \in \mathcal{Q}(\mathcal{D})$ how can we discover the $\mathcal{PQ}(s)$ that best represent $p(s, t)$, where p is the predicate of $\mathcal{Q}$ according to the Definition 1.

We make the hypothesis that the shortest Wikipedia path query that maximally containment $\mathcal{Q}(\mathcal{W})$ is the best navigational path to represent in Wikipedia the predicate p of the query $\mathcal{Q}$.

Definition 5. *$\mathcal{MC}(\mathcal{Q})$ is the shortest maximally containment path query if and only if $\nexists$ a path query $\mathcal{MC}'(\mathcal{Q})$ such that $\mathcal{R}(\mathcal{MC}(\mathcal{Q})(s),\mathcal{W}) \subset \mathcal{R}(\mathcal{MC}'(\mathcal{Q})(s),\mathcal{W}) \subseteq \mathcal{Q}(\mathcal{W})$.*

In order to calculate $\mathcal{MC}(\mathcal{Q})$, we developed an indexing algorithm detailed in the next section.

B. Path Indexing Algorithm

The *path indexing algorithm PIA* computes the $\mathcal{MC}(\mathcal{Q})$ (Algorithm 1). Given a SPARQL query $\mathcal{Q}$, it takes as inputs the $\mathcal{Q}(\mathcal{W})$ and L. PIA calls the *generateAllPaths* function to perform a depth-first search up to L starting from the source page s of each pair in $\mathcal{Q}(\mathcal{W})$. For each path reaching the target page *t*, it transforms the path into a path query (Algorithm 2) and builds an index as a bipartite

Table I
PATH QUERIES GENERATED BY PIA

$\mathcal{Q}_1(\mathcal{W})$	Path	Path Query
(Paris, Pierre_Curie)	Paris/Cat:Paris/Cat:People_from_Paris/Pierre_Curie	#from/Cat:#from/Cat:People_from_#from/#to
(Rosario, Lionel_Messi)	Rosario/Cat:Rosario/Cat:People_from_Rosario/Lionel_Messi	#from/Cat:#from/Cat:People_from_#from/#to
	Rosario/ Lionel_Messi	#from / #to

graph linking path queries with pairs in $\mathcal{Q}(\mathcal{W})$. The $index$ is defined by three sets: (1) the ordered set of path queries $\mathcal{PQ}(s)$ (ordered by element degree), (2) V is the set of pairs (s,t) where exists a path in Wikipedia up to L and (3) E is the set of edges relating elements from $\mathcal{PQ}(s)$ with elements from $\mathcal{Q}(\mathcal{W})$. The $\mathcal{MC}(\mathcal{Q})$ is the first ranked path query in the index.

Algorithm 1 PIA algorithm

Input: $\mathcal{Q}(\mathcal{W}), L$
Output: Path query index
  $index = (\varnothing, \varnothing, \varnothing)$
  for all $(s,t) \in \mathcal{Q}(\mathcal{W})$ **do**
    $paths \leftarrow \varnothing$, $curL \leftarrow 0$, $p \leftarrow \varnothing$
    $generateAllPaths(s,t,L,curL,paths,p)$
    for all $p' \in paths$ **do**
      $pathQuery \leftarrow pathQueryGenerate(p',s,t)$
      $index \leftarrow insertInIndex(index, pathQuery, (s,t))$
    end for
  end for
  return $index$

Algorithm 2 pathQueryGenerate

Input: $path : p_1/.../p_n; s,t \in \mathcal{W}$
Output: A path query
  for all p in $path$ **do**
    $p \leftarrow stringReplace(p,s,\#from)$
    $p \leftarrow stringReplace(p,t,\#to)$
  end for
  return $path$

```
PREFIX r:<http://dbpedia.org/resource/>
PREFIX o:<http://dbpedia.org/ontology/>
PREFIX p:<http://dbpedia.org/property/>

#Q1: Cities and People born there.
SELECT ?city, ?person WHERE{
?person a o:Person.
?city a o:City.
?person p:birthPlace ?city}

#Q2: Cities and Philosophers born there.
SELECT ?city, ?philosopher WHERE{
?philosopher a o:Philosopher.
?city a o:City.
?philosopher p:birthPlace ?city}

#Q3: France and Philosophers born there.
SELECT r:France, ?philosopher WHERE{
?philosopher a o:Philosopher.
?philosopher p:birthPlace r:France}

#Q4: Books and its authors.
SELECT ?book, ?author WHERE{
?book a o:Book.
?book p:author ?author }

#Q5: Composer and its works.
SELECT ?musician, ?work WHERE{
?work a o:Work.
?work p:musicBy ?musician }

#Q6: Cities and its universities.
SELECT ?city, ?university WHERE{
?university a o:University.
?city a o:City.
?university p:city ?city}
```

Listing 2. Semantic Queries of the Evaluation

IV. IMPLEMENTATION AND EVALUATION

We have implemented the above algorithms in Java and executed them on a local copy of Wikipedia obtained by the English Wikipedia dumps from October 2011. We also implemented an algorithm to extract all results of a given SPARQL query from DBpedia.

Table II
INDEX OF THE QUERIES.

Query	Rank	Path Query	#	∩%
Q_1	1	**#from/ Cat:from/Cat:People_from_-#from/ #to**	34008	100
	2	#from/#to	16312	82
Q_2	1	**#from /Cat:from/Cat:People_-from_#from/ #to**	60	100
	2	#from /Cat:Capitals_in_-Europe/ Cat:#from/People_-from_#from/#to	15	100
Q_3	1	**#from/ Cat:#from/Cat:French people/ Cat:French_-people_by_occupation/ French_philosophers/ #to**	21	100
	2	#from/ Cat:#from/ Cat:French people/Cat:French_-people_by_occupation/ French_sociologists/ #to	3	100
Q_4	1	**#from/#to**	19863	100
	2	#from/ Cat:#to/ #to	119	100
Q_5	1	**#from/ #to**	811	100
	2	#from/Cat:Tony_Award_-winners/Cat:Tony_Award_-winning_musicals/#to	26	100
Q_6	1	**#from/#to**	6031	100
	2	#from/Cat:#from/ #to	1314	95

We conducted an evaluation to answer three main questions: *Is the $\mathcal{MC}$ a good representation for a given semantic property?*, *What is the relevance level of the $\mathcal{MC}$ in Wikipedia?* and *Does Wikipedia community accept the links added thank to the $\mathcal{MC}$?*.

To answer the above questions, we run the PIA algorithm on the local copy of Wikipedia and on the results of six representative SPARQL queries. Listing 2 details the queries, their identification and their intents. We executed PIA algorithm during January 2012 with L=5.

A. Shortest Maximally Containment Path Query representativeness

To calculate the *representativeness* of the $\mathcal{MC}$s, we compare each $\mathcal{MC}(Q_i)$ with other path queries in the index resulting of the execution of the PIA algorithm. The comparison is based on the cardinality and the coverage. Table II presents the obtained path index after the execution of the PIA algorithm for the six queries.

We notice that the $\mathcal{MC}$ for each query has the highest cardinality in the index and it has a very high coverage. It is 95% or more compared with the second ranked path query, except for the $\mathcal{MC}$ of Q_1. However for the Q_2 and

Table III
RECALL AND PRECISION FOR THE $\mathcal{MC}(\mathcal{Q}_i)$ FROM THE EVALUATION OF BINARY QUERIES.

Query	Query property	Precision	Recall
Q_1	is birthPlace of	0.415	**0.52**
Q_2	is birthPlace of	0.003	**0.58**
Q_3	is birthPlace of	0.099	**1**
Q_4	author	0.22	**0.97**
Q_5	is musicBy of	0.027	**0.97**
Q_6	is city of	0.014	**0.63**

Q_3, which are more specific, the coverage is total. This leads us to believe that the PIA algorithm is well adopted for specific queries.

B. Shortest Maximally Containment Path Query relevance

We computed precision and recall for each $\mathcal{MC}(\mathcal{Q}_i)$. We considered relevant documents those which are in DBpedia and Wikipedia as well. Table III shows the results. The precision values are rather low, the $\mathcal{MC}(\mathcal{Q}_i)$ has more results than expected. For queries Q_1, Q_2 and Q_3, this is because the social categorization in Wikipedia has different criteria to relate articles within a category; for example the category `Cat:People_from_Rosario` includes either people who was born in Rosario (Argentina) and also people related to Rosario. On the other hand, the $\mathcal{MC}$ for queries Q_4 to Q_6 is a direct link, the $\mathcal{MC}$ has a large number of pages which are not related with those retrieved by semantic queries. Therefore, we may consider that precision is not enough relevant for our study because of the flexibility of the Wikipedia categorization model.

Regarding recalls, the values are better, even they significantly increased for more specific queries. This leads us to think that the correctness level of the $\mathcal{MC}$ can be improved if the domain and range of the property of the queries are more specific types.

C. Wikipedia Community Evaluation

We also carried out a Wikipedia community evaluation. We started by enriching Wikipedia by adding new contributions according to the results obtained of the $\mathcal{MC}$ of each query. After that, we observed whether the Wikipedia community accepted or rejected such contributions.

Table IV details these results only for the Q_1, Q_5, Q_6 queries. The first column identifies the query, the second column details the number of contributions that we made in Wikipedia. The *Last edition* column indicates the number of pages that were not edited by any Wikipedia user after our edition; then, the *Other editions* column details the number of pages that were edited by another Wikipedia user but preserves our contribution; and the last column details the contributions that were rejected by the community[7].

According to the Table IV, the rate of rejection is 16.6% for Q_1, 10% for Q_6 and 1.5% for Q_5. Q_1 had

[7]All the contributions could be checked in the contribution page of magic.towers user in Wikipedia

Table IV
COMMUNITY EVALUATION RESULTS. [a] 13 OF THESE CONTRIBUTIONS WERE MADE BY WIKIPEDIA USERS WHO RESPECTS THE $\mathcal{MC}(Q_5)$

Query	Contributions	Last edition	Other editions	Rejected
Q_1	78	35.9%	47.4%	16.6%
Q_5	63	25.4%	73%[a]	1.6%
Q_6	70	25.7%	64%	10%

13 rejections. Contribution of this query consisted in categorizing articles. Two causes of rejections were identified: a more general categorization was preferred to a specific one and a specific category was unnecessarily created. For instance, the category People from Edinburgh was changed by the more specific *Sportspeople from Edinburgh* and the *People from Dayton, Kentucky* category was deleted because *"Dayton, Kentucky is a small community"* and *"this category contain one article and have little possibility for growth"* [8]. The solution was *upmerge* the categorization. On the other hand, the rejections related to the query Q_6 were produced in city articles that include a list of educational institutes in a Wikipedia special page.

The social evaluation demonstrated that contributions derived from the $\mathcal{MC}$ were generally accepted by the Wikipedia community.

However, some editions were rejected. Why? Is there another possible interpretation of the path index? Although the levels of representation of $\mathcal{MC}$ are sustainable, the Wikipedia community feedback shows us that there are different conventions for the same property of DBpedia. These conventions depend on the articles development and on the community that sustains them. Therefore, we can conclude that adding missing link in Wikipedia is socially accepted but convention detection must be improved. In this direction, we need to analyze not only the general conventions but also the convention of a specific community that maintains those resources.

V. RELATED WORK

Wigipedia [7] allows regular Wikipedia's users to introduce missing semantic relations from DBpedia into the current Wikipedia article. What we propose in this paper is more general, it is not related to only one article but it uses the result of a semantic query. In addition, Wigipedia does not take into consideration the Wikipedia community convention as a whole, like we do in our approach.

Hoffman et al. [8] developed a human-machine collaborative system to complete infobox information by taking sentences from the Wikipage using Kylin. Like in our approach, the machine effort is used to detect missing relations in Wikipedia. However, we are able to fix relations among articles which do not have necessary related sentences in their text body.

Other approaches propose to detect missing links in Wikipedia compared in [9], [10]. These approaches can

[8]http://en.wikipedia.org/wiki/Wikipedia:Categories_for_discussion/Log/2012_March_25

be used only to fix missing direct links. In our approach, PIA allows to detect relations among Wikipedia articles and categories, not only with direct links.

Recently, new approaches propose to combine Wikipedia and DBpedia, Gerber et al. [11] introduced the framework BOA. The goal of the BOA framework is to allow extracting structured data from unstructured data. This work is similar with our approach since both study the synergy between Web of Data and Web of Document. However, BOA follows the traditional approach, it extracts information from the Web of Document to enrich the Web of Data. In this work, we propose an original approach where the information from the Web of Data is inserted in the Web of Document.

In a previous work [12], we proposed an approach to inject DBpedia information that is missing in Wikipedia by learning from existing relations in Wikipedia. This paper extends the previous work; it develops the Path Index Algorithm (PIA) and evaluates the PIA approach on representative DBPedia queries.

VI. Conclusion and Further Work

In this article, we raised issues concerning the information gap between DBPedia and Wikipedia. We proposed an approach that detect missing links and discover from established links the conventions to be applied for the missing links.

We developed the Path Index Algorithm algorithm, which indexes path queries in Wikipedia based on result sets generated by queries on DBpedia. Next, we run an evaluation using six DBpedia semantic queries. The evaluation results demonstrate that adding missing link in Wikipedia is socially accepted. Although, PIA algorithm can discover Wikipedia conventions, it must be improved for specific cases e.g. the $\mathcal{MC}$ may be slightly more general or more specific depending on the particular case.

Therefore, the work reported in this paper is a first approximation to our goals. At the moment we are working on improving the PIA algorithm in order to better approximate the $\mathcal{MC}$. We also are working on a better generalization strategy to reduce the number of path queries and thus, to reduce the indexes. Regarding the evaluation, we are designing a new evaluation that tends to cover all DBpedia properties and would allow us to be actually aware of the dimension of the information gap between Wikipedia and DBpedia.

Acknowledgements.

This work is supported by the French National Research agency (ANR) through the KolFlow project (code: ANR-10-CONTINT-025), part of the CONTINT research program. This work was also funded by the PAE 37279-PICT 02203 which is sponsored by the ANPCyT, Argentina.

References

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia - a crystallization point for the web of data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154 – 165, 2009.

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 697–706.

[3] S. Abiteboul and V. Vianu, "Regular path queries with constraints," in *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, ser. PODS '97. New York, NY, USA: ACM, 1997, pp. 122–133.

[4] F. Alkhateeb, J.-F. Baget, and J. Euzenat, "Extending sparql with regular expression patterns (for querying rdf)," *Web Semantic Journal*, vol. 7, no. 2, pp. 57–73, April 2009.

[5] M. Arenas, S. Conca, and J. Pérez, "Counting beyond a yottabyte, or how sparql 1.1 property paths will prevent adoption of the standard," in *Proceedings of the 21st international conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 629–638.

[6] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener, "The lorel query language for semistructured data," *International Journal on Digital Libraries*, vol. 1, pp. 68–88, 1997.

[7] S. Bostandjiev, J. O'Donovan, C. Hall, B. Gretarsson, and T. Höllerer, "Wigipedia: A tool for improving structured data in wikipedia," in *ICSC*. IEEE, 2011, pp. 328–335.

[8] R. Hoffmann, S. Amershi, K. Patel, F. Wu, J. Fogarty, and D. S. Weld, "Amplifying community content creation with mixed initiative information extraction," in *Proceedings of the 27th international conference on Human factors in computing systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1849–1858.

[9] S. F. Adafre and M. de Rijke, "Discovering missing links in wikipedia," in *Proceedings of the 3rd international workshop on Link discovery*, ser. LinkKDD '05. New York, NY, USA: ACM, 2005, pp. 90–97.

[10] O. Sunercan and A. Birturk, "Wikipedia missing link discovery: A comparative study," in *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*. AAAI, 2010.

[11] D. Gerber and A.-C. Ngonga Ngomo, "Bootstrapping the linked data web," in *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*, 2011.

[12] D. Torres, P. Molli, H. Skaf-Molli, and A. Diaz, "Improving wikipedia with DBpedia," in *SWCS - Semantic Web Collaborative Spaces Workshop 2012 in 21st WWW Conference 2012*. Lyon, France: ACM, 2012.