Recommending non-English Wikipedia Links Using DBpedia Properties

Diego Torres[1], Alicia Díaz[1], Hala Skaf-Molli[2], and Pascal Molli[2]

[1] LIFIA,Fac. Informática, Universidad Nacional de La Plata,
1900 La Plata, Argentina,
{diego.torres, alicia.diaz}@lifia.info.unlp.edu.ar,
[2] Université de Nantes, LINA, 2, rue de la Houssiniere, 44322 Nantes, France,
{Hala.Skaf, Pascal.Molli}@univ-nantes.fr

Abstract. DBpedia knowledge is mainly obtained by extracting information from Wikipedia. DBpedia can easily infer new facts that are not present or partially present in Wikipedia. How can these new facts be automatically inserted into Wikipedia? This requires discovering conventions used to express relations in Wikipedia. In this paper, we extend previous work to see how given a relation in English DBpedia, it is possible to discover convention in English, French and Spanish Wikipedia. We compare the conventions of different Wikipedia languages and the effectiveness of BlueFinder recommender according to the languages.

Keywords: Semantic Web, Social Web, DBpedia, Wikipedia, Recommendations

1 Introduction

The Semantic Web brings the ability to have better search and navigability on the Web. It is mainly build from meta-data extracted from the Social Web. DBpedia [1] knowledge base is built from data extracted from Wikipedia infoboxes and categories. The semantic capacities of DBpedia enable SPARQL queries to retrieve information that is originally not in Wikipedia [2]. Therefore, it is possible to enhance Wikipedia with information inferred from DBpedia.

Pushing back information from DBpedia to Wikipedia enables to enrich the Social Web with information extracted from the Semantic Web. In the case of Wikipedia, this requires to discover Wikipedia conventions[3] to express this new information.

In a previous work, we introduced an item-based collaborative filtering algorithm called BlueFinder [3] to recommend the best navigational path in Wikipedia that expresses semantic property of DBpedia. The algorithm receives unconnected pages (p_1, p_2) of the English Wikipedia i.e. pages that do not have a navigational path from p_1 to p_2. If their corresponding pages in the English DBpedia are related by a semantic property s, then BlueFinder recommends navigational

[3] http://en.wikipedia.org/convention

path patterns that best express the semantic property s and respect the English Wikipedia conventions.

However, Wikipedia conventions can differ according to Wikipedia language. In this paper, we want to discover how a given semantic property in the English DBpedia is expressed in the French and the Spanish Wikipedia. We want to compare the different conventions. The contributions of the paper are:

- We adapted the BlueFinder recommender for handling non-English Wikipedia using inter-language links.
- We adapted the BlueFinder navigational pattern detection to handle non-English Wikipedia conventions.
- We evaluated the performance of BlueFinder recommender in terms of precision and recall for Spanish and French Wikipedia. Predictions are better in French Wikipedia because Spanish Wikipedia has fewer articles, therefore, sparsity problems increase.

This paper is organized as follows. Section 2 provides background about BlueFinder algorithm. Section 3 details the adaptation of the BlueFinder algorithm to French and Spanish Wikipedia. Section 4 presents the results of the experimentation of the adapted algorithm to French and Spanish Wikipedia. Results are better in French Wikipedia than in Spanish Wikipedia because of sparsity problems. Finally, the last section concludes the paper and presents further work.

2 Background

We represent a Wikipedia convention as path query [3]. A path query is a generalization of similar paths, for example, the path query `#from/ Cat:#from/ Cat:People_from_#from/ #to` is the convention to represents (or cover) the path `Paris/ Cat:Paris/ Cat:People_from_Paris/ Julie_Andrieu` and `Rosario/ Cat:Rosario/ Cat:People_from_Rosario/ Lionel_Messi`. The symbol `#from` replace all the occurrences of the origin page and the symbol `#to` replace all the occurrences of target page, in the examples Paris and Julie_Andrieu and Rosario and Lionel_Messi accordingly.

According to Adomavicioius and Tuzhilin [4], *"collaborative recommender systems try to predict the utility of items for a particular user based on the items previously rated by other users"*. More formally, the utility $u(c, s)$ of item s for user c is estimated based on the utilities $u(c_j, s)$ assigned to item s by those users $c_j \in C$ who are "similar" to user c. In the context of Wikipedia, we do not directly transpose recommenders to suggest Wikipedia articles to users but to suggest links between articles. We want to predict the utility of path queries for a particular pair of Wikipedia articles based on those rated by Wikipedia community. In other words, the pairs of articles (from,to) will play the role of users ($Q_p(D)$) and the path queries will be the items (PQ). Then, the utility $u(c, pq)$ of a path query pq for a pair c related by a semantic property p is estimated based on the utilities $u(c_j, pq)$ assigned to pair c by those pairs $c_j \in C_p(l)$, $u : Q_p(D) \times PQ \to R$, where $C_p(l)$ is the set of pairs $(f, t) \in Q_p(D)$

that are connected in Wikipedia by a path with length up to l[4] and R is a totally ordered set.

Given a property p in DBpedia, $C_p(l)$ and PQ path queries covered by the elements of $C_p(l)$. Then, for a given pair of Wikipedia articles $(from, to)$, BlueFinder recommends the path query that maximise the utility function.

BlueFinder is a collaborative filtering recommender system. It uses a memory-based algorithm [5] to make rating predictions based on the entire collection of previously rated path queries. The value of the unknown rating $r_{c,s}$ for a pair c and path query s will be computed as an aggregate rating of other k similar pairs for the same path query s. The recommender returns a set of recommended path queries that can be used to represent the semantic property. The recommendations have to include at least one path query that can represent the semantic relation following the conventions of Wikipedia community. BlueFinder is based on the popular k-Nearest Neighbors(kNN) and Multi label kNN algorithm [6] adapted to the context of DBpedia and PIA index. The PIA index is a bipartite graph $(PQ, C_p(l), I)$ that represents the coverage of path queries for a set of pairs of Wikipedia articles that are related by a DBpedia property p and, additionally, the path query with more coverage in PQ is the general representation of the semantic property p in Wikipedia [2, 7]. The BlueFinder algorithm first identifies the k neighbors of the unconnected pair (from, to), and then applies PIA algorithm only for the k nearest neighbors to generate the PIA index. PIA results will be the prediction set.

The k-Nearest Neighbors algorithm uses a similarity measure function to select the nearest neighbors. In this work we use the well-known Jaccard distance [8] to measure similarity. The Jaccard distance measures the degree of overlap of two sets and ranges from 0 (identical sets) to 1 (disjoint sets). In this work, we apply Jaccard distance to types of DBpedia resources.

Definition 1. *Given two pairs of pages $c_1 = (a, b)$ and $c_2 = (a', b')$ and the data type set for b and b' in DBpedia defined as $B = \{t/\ b\ rdf : type\ t\ \} \in DBpedia$ and $B' = \{t'/\ b'\ rdf{:}type\ t'\} \in DBpedia\ \}$. The similarity measure $jccD(c_1, c_2)$ is defined by*

$$jccD(c_1, c_2) = Jaccard\ distance(B, B') = \frac{|B \cup B'| - |B \cap B'|}{|B \cup B'|}$$

Now, we can define the kNN [9] in our context as:

Definition 2. *(KNN) Given a pair $r \in Q_p(D)$ and an integer k, the k nearest neighbors of r denotes $KNN(r, Q_p(D))$ is a set of k pairs from $Q_p(D)$ where $\forall o \in KNN(r, Q_p(D))$ and $\forall s \in Q_p(D) - KNN(r, Q_p(D))$ then $jccD(i, r) \leq jccD(s, r)$.*

The value for an unknown rating $r_{c,s}$ for a unconnected pair in Wikipedia c and a path query $s \in C_p(l)$, can be computed as:

$r_{c,s}$ =degree of s in PQ,
where $(PQ, V, I) = PIA(KNN(c, Q_p(D)))$

[4] We restrict the length for practical reasons.

BlueFinder algorithm citeBlueFinder computes predictions for $r_{c,s}$. It receives four inputs: (1) k: the number of neighbors, (2) $maxRecom$: the maximum number of recommendations, (3) $Q_p(D)$: PIA index and (4) x : the unconnected pair of Wikipedia. It produces a set of star path queries as recommendations.

3 Adapting BlueFinder to French and Spanish Wikipedia

In this section, we explain the process to adapt and apply BlueFinder algorithm to French Wikipedia (*W@fr*) and Spanish Wikipedia (*W@sp*).

In order to apply BlueFinder to a given English DBpedia property, we have to provide the set of pairs of pages that are related by the property $Q_p(D)$ to generate the PIA index. For example, to obtain the pairs of pages for the semantic property *birthplace* that relates Cities and People we execute the following SPARQL query on DBpedia:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
PREFIX dbpedia-owl:<<http://dbpedia.org/ontology/>.
SELECT DISTINCT ?wpCity ?wpPerson WHERE{
  ?enCity a dbpedia-owl:City.
  ?enPerson a foaf:Person.
  ?enPerson dbpedia-owl:birthPlace ?enCity.
  ?enCity foaf:isPrimaryTopicOf ?wpCity.
  ?enPerson foaf:isPrimaryTopicOf ?wpPerson.
}
```

Listing 1: SPARQL query to obtain the English Wikipedia pages that are related by the birthplace property.

If we use the pairs retrieved by the above query with PIA in the non-English Wikipedia, PIA will generate neither $C_p(l)$ set nor the PIA index. Most of the pages retrieved by the query in Listing 1 do not exist in the non-English Wikipedia because the pages retrieved by this query are only in English. For example, the "Edinburgh" page in the English Wikipedia (*W@en*) does not exist neither in the *W@sp* nor in the *W@fr*: the Spanish name is "Edinburgo" and the French is "Édimbourg". In order to compute PIA, the name of the retrieved pages must to be translated.

As a first adaptation of BlueFinder, we made page name translation by using the `owl:sameAs`[5] property. In DBpedia, the different language versions of a resource are associated by the `owl:sameAs` property. For example, Listing 2 shows the SPARQL query to obtain Spanish resources that are related with *birthplace* property. The `filter` in the SPARQL query helps to select resources in one specific language. With this adaptation we can apply PIA without problems.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
PREFIX dbpedia-owl:<<http://dbpedia.org/ontology/>.
```

[5] http://www.w3.org/TR/owlref/#sameAs-def

```
PREFIX owl:http://www.w3.org/2002/07/owl#.
SELECT DISTINCT ?from ?to WHERE{
  ?enCity a dbpedia-owl:City.
  ?enPerson a foaf:Person.
  ?enPerson dbpedia-owl:birthPlace ?enCity.
  ?enCity owl:sameAs ?from.
  ?enPerson owl:sameAs ?to.
  FILTER (regex(?from, "http://es.dbpedia.org") &&
       regex(?to, "http://es.dbpedia.org")). }
```

Listing 2: SPARQL query to obtain the Spanish DBpedia resources that are related by the birthplace property.

The second adaptation of BlueFinder is to apply the *jccD* distance function in the English DBpedia instead of using French or Spanish DBpedia. Because we observed that English DBpedia has a richer description of resources than non-English Wikipedias. Non-English Wikipedias can derive a wrong selection of similar items producing incorrect prediction. For example, if we compare the types that describe the famous singer *John Lennon* and famous actor *Al Pacino* pages in English DBpedia (*DB@en*) and Spanish DBpedia (*DB@sp*), we can notice that the detailed descriptions for both pages in *DB@en* have 42 and 22 classes respectively and 8 and 7 classes in the *DB@sp*. This difference in the use of types among the different language versions of DBpedia can generate contradictions in the distance measurement. In the English version *John Lennon* and *Al Pacino* are measured as distant (0.857). Meanwhile in the Spanish version, they are measured as close (0.334). We obtain the English DBpedia types of a non-English resource using the `owl:sameAs` property. Listing 3 shows an example for the Spanish version of Al Pacino.

```
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
  PREFIX dbpedia-owl:<<http://dbpedia.org/ontology/>.
  PREFIX owl:http://www.w3.org/2002/07/owl#.
  SELECT DISTINCT ?type WHERE{
?enFrom <owl:sameAs> <es.dbpedia.org/resource/Al_Pacino>.
?enFrom a ?type.}
```

Listing 3: SPARQL query to obtain English type description for Spanish Al Pacino resource.

The last adaptation is the translation of path queries to English by means of cross language links. The translation is from the non-English Wikipedia pages to the English version. This enables us to use the same normalization strategy as in the English Wikipedia and also to compare the conventions between the non-English Wikipedia.

4 Experimentation

We applied the adapted version of the BlueFinder algorithm to Spanish and French Wikipedia and English DBpedia, all of them are retrieved on January

2013. We want to discover French and Spanish Wikipedia conventions for English DBpedia property *birthplace*. We evaluate precision and recall for both languages.

4.1 Dataset

Data set	$\|PQ\|$	$\|C_p(l)\|$	$\|I\|$	$\|Q_{birthplace}(D)\|$
$Q_{birthplace@sp}(D)$	1,502	7,486	14,998	22,281
$Q_{birthplace@fr}(D)$	3,721	30,407	114,156	36,952

Table 1. PIA index data sets used in the experimentation. Columns shows the number of path queries, the number of connected pairs, the number of edges and the number of elements in the Data set

For this experimentation, we used the semantic property *birthplace* that relates Cities with People. We run $Q_{birthplace@sp}$ and $Q_{birthplace@fr}$ given in Listing 2 and 4 respectively. We used a PIA index for $Q_{birthplace@sp}(D)$ and another for $Q_{birthplace@fr}(D)$. Table 1 details these indexes. Then we run BlueFinder algorithm on the same datasets as the PIA index.

```
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>.
  PREFIX dbpedia-owl:<<http://dbpedia.org/ontology/>.
  PREFIX owl:http://www.w3.org/2002/07/owl#.
  SELECT DISTINCT ?from ?to WHERE{
    ?enCity a dbpedia-owl:City.
    ?enPerson a foaf:Person.
    ?enPerson dbpedia-owl:birthPlace ?enCity.
    ?enCity owl:sameAs ?from.
    ?enPerson owl:sameAs ?to.
FILTER (regex(?from,"http://fr.dbpedia.org") &&
regex(?to,"http://fr.dbpedia.org")). }
```

Listing 4: SPARQL query to obtain the French Wikipedia resources that are related by the birthplace property.

4.2 Metrics

We use *precision*, *recall* [10], *pot* [3] and F_1 *score* [11] as metrics. These metrics compute the proportion between the BlueFinder retrieved path queries (BFPQs) and the relevant ones i.e. correct path queries (CPQs). The *precision* is the proportion of retrieved path queries that are correct with respect to connected Wikipedia articles (1), *recall* is the proportion of correct path queries that are retrieved (2). *pot* measures the number of cases where BlueFinder recommends at least one correct path query that can fix the missing links (potentially fixed) (3). Finally, F_1 *score* is the combination of precision and recall (4)

$$(1) precision(CPQs, BFPQs) = \frac{|CPQs \bigcap BFPQs|}{|BFPQs|}$$
$$(2) recall(CPQs, BFPQs) = \frac{|CPQs \bigcap BFPQs|}{|CPQs|}$$
$$(3) pot(CPQs, BFPQs) = \begin{cases} 0 & if CPQs \bigcap BFPQs = \emptyset \\ 1 & otherwise \end{cases}$$
$$(4) F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

4.3 Methodology

In order to discover the Wikipedia convention for birthplace property in Spanish and French Wikipedia we follow the next steps:

1. **Evaluation setup.** For all the executions the maximum length of the navigational path was 5. The values of k neighbors was from 1 to 10 and the limit of recommendations was $maxRecom$=5.
2. **Exercise BlueFinder in non-English Wikipedia** We eliminate links that already exist in the PIA index and then we observe if BlueFinder is able to recreate them. We generate a mock PIA index without the path queries of the pair i.e. we disconnect the pair. After that, we exercise BlueFinder to fix the pair by using the mock PIA index. Then, we compare the recommendations with the path queries of PIA.

We exercised BlueFinder with 5000 pairs in each PIA index.

4.4 Results

K	1	2	3	4	5	6	7	8	9	10
$precision_k$	0.426	0.400	0.347	0.313	0.283	0.264	0.245	0.232	0.224	0.214
$recall_k$	0.421	0.547	0.602	0.634	0.652	0.664	0.669	0.674	0.674	0.670
pot_k	0.485	0.614	0.667	0.697	0.714	0.727	0.732	0.735	0.734	0.731
F_1	0.424	0.462	0.44	0.419	0.395	0.377	0.358	0.345	0.336	0.324

Table 2. Detail of BlueFinder evaluation in the Spanish Wikipedia.

The PIA index for the Spanish Wikipedia had 7,486 connected pairs out of 22,281 retrieved by the SPARQL query (Table 1). This means that the set of pairs to learn is smaller than the set of pairs to fix. This can generate a low rate of fixed values.

Surprisingly, the execution over the Spanish Wikipedia demonstrated that BlueFinder can fix 70 % of the cases. BlueFinder performs well for Spanish Wikipedia. This is evidenced with the curves of *precision*, *recall* and *pot* shown in Figure 1. For $K = 5$ the *pot* and *recall* rate begins to be stabilised and the *precision* begins to decrease.

In Table 2, the best *pot* value was 73,5% with K=10, the best *recall* was 0.674 for K=8 and 9 and the best *precision* was 0.426 with K=1. The worst

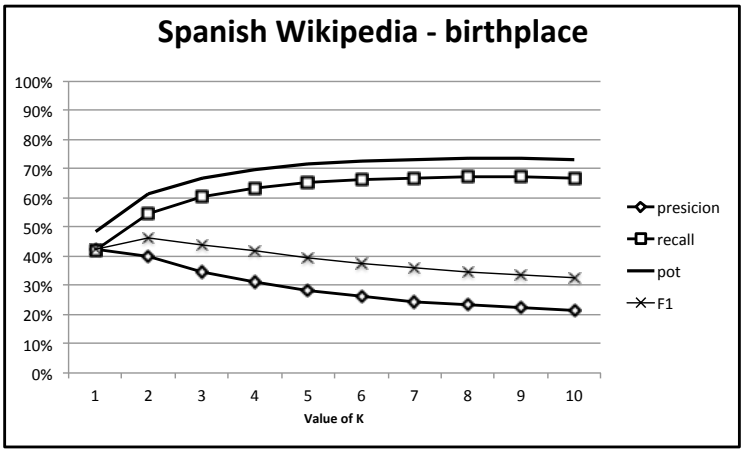

Fig. 1. BlueFinder evaluation in the Spanish Wikipedia.

value of *pot* was 48,5 % with K=1, the worst value of *precision* was 0.214 with K=10 and the worst value of *recall* was 0.421.

The SPARQL query for *birthplace* property in French language retrieves a set of 36,952 pairs and 30,407 are connected in Wikipedia by a navigation path (Table 1). That is equivalent to more than 80% of connected pairs. This means that the French Wikipedia is well developed for the birthplace property. Additionally, this is a good context to BlueFinder because it has a big number of cases to learn and find the proper neighbors.

The execution of BlueFinder over French Wikipedia confirms the effectiveness of the algorithm. In this case, BlueFinder can fix 90% of the cases with a balanced value of *precision* and *recall*. The curve of *precision*, *recall* and *pot* in Figure 2 showed that for K=8 the *pot*, *recall* and *precision* begins to stabilised with the 90% of fixed ratio. In Table 3 the best value for *pot* was 91% for K=10, the best value for *recall* was 0.724 for K=10 and the best value for *precision* was 0.559 for K=1. The worst value for *pot* was 0.72 for K=1, the worst value for *recall* was 0.552 for K=1 and the worst value for *precision* was 0.407 for K=10.

K	1	2	3	4	5	6	7	8	9	10
precision	0.559	0.499	0.460	0.438	0.424	0.418	0.413	0.412	0.408	0.407
recall	0.552	0.642	0.670	0.685	0.693	0.706	0.710	0.719	0.721	0.724
pot	0.720	0.807	0.837	0.862	0.870	0.887	0.891	0.902	0.906	0.910
F_1	0.555	0.561	0.545	0.534	0.526	0.524	0.521	0.523	0.520	0.521

Table 3. Detail of BlueFinder evaluation in the French Wikipedia.

Finally, the path query `#from/Cat:#from/Cat:People_from_#from/#to` was the best ranked in PIA index for the Spanish Wikipedia with 3,190 pairs cov-

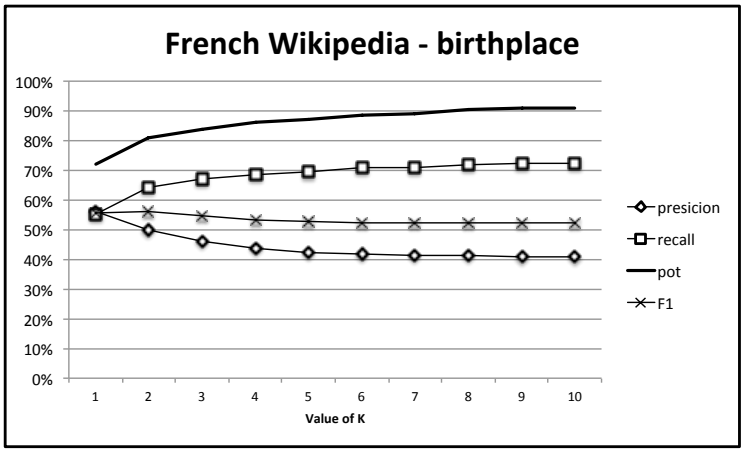

Fig. 2. BlueFinder evaluation in the French Wikipedia.

ered (out of 7,486), and the second best ranked in the French Wikipedia PIA index with 11,335 pairs covered (out of 30,407). This demonstrates that both Wikipedias share the same convention for *birthplace* with some minors differences.

5 Conclusions and Further Work

DBpedia can easily infer new facts that are not in Wikipedia. We addressed the problem of inserting these new facts back to different Wikipedia languages. In previous work, we proposed BlueFinder algorithm to discover conventions and enrich English Wikipedia. This work adapts BlueFinder to handle Spanish and French Wikipedia. From a given relation in English DBpedia, BlueFinder discovers conventions in French and Spanish Wikipedia. Experiments demonstrate that both Wikipedia share the same conventions with some minors differences and that BlueFinder delivers good recommendations in 70% of cases in Spanish and 91% of cases in French. Recommendation is better in French Wikipedia because it has greater number of pairs and paths than Spanish Wikipedia. Future works will explore more relations and try to improve *precision* of recommendation by using more adequate similarity metrics.

Acknowledgments. 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. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web **7**(3) (2009) 154 – 165
2. Torres, D., Molli, P., Skaf-Molli, H., Díaz, A.: Improving Wikipedia with DBpedia. In Mille, A., Gandon, F.L., Misselis, J., Rabinovich, M., Staab, S., eds.: WWW (Companion Volume), ACM (2012) 1107–1112
3. Torres, D., Skaf-Molli, H., Molli, P., Diaz, A.: BlueFinder: Recommending Wikipedia Links Using DBpedia Properties. In: ACM Web Science Conference 2013 (WebSci 13), Paris, France (May 2013)
4. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6) (2005) 734–749
5. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. (1998) 43–52
6. Zhang, M., Zhou, Z.: A k-nearest neighbor based algorithm for multi-label classification. In: Granular Computing, 2005 IEEE International Conference on. Volume 2., IEEE (2005) 718–721
7. Torres, D., Molli, P., Skaf-Molli, H., Diaz, A.: From dbpedia to wikipedia: Filling the gap by discovering wikipedia conventions. In: 2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI'12). (2012)
8. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. Computer Vision–ECCV 2008 (2008) 537–547
9. Lu, W., Shen, Y., Chen, S., Ooi, B.: Efficient processing of k nearest neighbor joins using mapreduce. Proceedings of the VLDB Endowment **5**(10) (2012) 1016–1027
10. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In Shavlik, J.W., ed.: ICML, Morgan Kaufmann (1998) 46–54
11. Chan, K., Chen, T., Towey, D.: Restricted random testing. Software QualityECSQ 2002 (2006) 321–330