An Ontology Matching Approach Based on Affinity-Preserving Random Walks

Chuncheng Xiang, Baobao Chang, Zhifang Sui
Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
Collaborative Innovation Center for Language Ability, Xuzhou 221009 China
{ccxiang,chbb,szf}@pku.edu.cn

Abstract

Ontology matching is the process of finding semantic correspondences between entities from different ontologies. As an effective solution to linking different heterogeneous ontologies, ontology matching has attracted considerable attentions in recent years. In this paper, we propose a novel graph-based approach to ontology matching problem. Different from previous work, we formulate ontology matching as a random walk process on the association graph constructed from the to-be-matched ontologies. In particular, two variants of the conventional random walk process, namely, Affinity-Preserving Random Walk (APRW) and Mapping-Oriented Random Walk (MORW), have been proposed to alleviate the adverse effect of the false-mapping nodes in the association graph and to incorporate the 1-to-1 matching constraints presumed in ontology matching, respectively. Experiments on the Ontology Alignment Evaluation Initiative (OAEI[1]) datasets show that our approach achieves a competitive performance when compared with state-of-the-art systems, even though our approach does not utilize any external resources.

1 Introduction

Ontologies have recently gained popularity because they help with inter-operability, information sharing and knowledge reuse. However, there are many different ontologies that describe the same domain, they are often different from each other because they have been designed and constructed independently by different ontology developers. As a result, how to address the heterogeneity of ontologies has attracted considerable attention in recent years.

Ontology matching is the process of finding semantic correspondences between entities from different ontologies. It is one of the most effective solutions to solving the semantic heterogeneity problem [Shvaiko and Euzenat, 2013].

[1]The OAEI is an international initiative organizing annual campaigns for evaluating ontology matching systems. All of the ontologies provided by OAEI are described in OWL-DL language, and like most of the other participants our approach manages the OWL ontology. OAEI: http://oaei.ontologymatching.org/

A myriad of approaches have been proposed for ontology matching. In graph-based ontology matching approaches, graphs are used to represent the two input ontologies and compute structural similarities of graphs. Examples of these approaches include Anchor-Prompt [Noy and Musen, 2001], GMO [Hu *et al.*, 2005] and Similarity Propagation [Li *et al.*, 2009; Ngo *et al.*, 2012b]. Anchor-Prompt is an ontology merging and mapping tool, which treats ontologies as directed labeled graphs. The basic idea is that if two pairs of entities are similar and there are paths connecting them, the entities in these paths are often similar as well. GMO is an iterative structural matcher, that uses RDF bipartite graphs to represent ontologies and computes structural similarities between entities by recursively propagating their similarities in the bipartite graphs. Similarity Propagation (SP) is a graph matcher inspired from the work in [Melnik *et al.*, 2002], which aims at database schema matching. SP uses fixed-point computation to determine corresponding nodes in the graphs. The basic idea is that the similarity between two nodes depends on the similarity between their adjacent nodes, or that similarities of nodes can propagate to their respective neighbors. SP is effective and has been embedded in some outstanding ontology matching systems such as RiMOM[Li *et al.*, 2009] and YAM++[Ngo *et al.*, 2012b].

In this paper, we propose a new graph-based approach based on random walks, which is inspired by the work in [Cho *et al.*, 2010] that aims at a graph matching problem in the image processing field. We introduce an association graph constructed with nodes as candidate mappings and edges as affinities between candidate mappings, and we show that the search for mappings between two to-be-matched ontologies can be cast as a node ranking and selection problem in the association graph. We summarize our contributions as follows. (1) We establish a novel random walk view for ontology matching and provide a random walk interpretations for ontology matching. (2) We preserve the original affinity relations between ontology entity pairs by adding an absorbing node into the traditional Markov chains to differentiate the potential true and false mappings. (3) We dynamically add the 1:1 matching constraint to the process of random walk to further improve performance.

To evaluate the effectiveness of our approach, we conduct experiments on the public datasets published in the OAEI campaign. The experimental results show that our match-

ing approach achieves a competitive performance when compared with the state-of-the-art systems.

2 Problem Statement

Ontology is a formal, explicit specification of a shared conceptualization in terms of classes, properties and relations [Euzenat and Shvaiko, 2013]. The process of ontology matching is to find correspondences between entities (classes, properties or individuals) from two ontologies. A mapping or a correspondence is defined as a four-tuple (as written in Eq. (1)), where e^1 and e^2 represent the entity in ontology O^1 and O^2, respectively, r is a type of matching relation (e.g., equivalent, subsume) and $k \rightarrow [0, 1]$ is the degree of confidence of matching relation between e^1 and e^2 [Mao *et al.*, 2010].

$$m =< e^1, e^2, r, k > \quad (1)$$

Similar to the work in [Mao *et al.*, 2011; Shvaiko and Euzenat, 2013; Ngo *et al.*, 2012a], we limit the type of entity to class and property, and focus on discovering only equivalent mappings between entities with cardinality 1:1. That is, one class (property) in ontology O^1 can be matched to at most one class (property) in ontology O^2 and vice versa.

3 Random Walks for Ontology matching

Ontology (more accurately, OWL ontology) can be modeled as a graph[Hu *et al.*, 2005; Euzenat and Shvaiko, 2013]. That is, given an ontology, we can define a graph $G =< V, E >$, where V is the set of labeled nodes representing the entities in the ontology and E is the set of edges representing the relations between two entities. Thus, ontology matching can be viewed as a graph matching problem that aims at finding the semantically similar nodes between two ontology graphs.

In this paper, we treat the problem of ontology matching between two given ontology graphs as a random walk process on an association graph $G^{rw} = (V^{rw}, E^{rw})$ constructed from the input ontology graphs, which will be described in detail in Sec.3.1. Then, the original ontology matching problem between ontologies is equivalent to selecting suitable nodes in the association graph G^{rw} because the selected nodes correspond to mappings between entities. To select nodes in G^{rw}, we adopt the statistic of the Markov random walks, which have been used to compute the ranking or relevance of web pages on the Internet [Kleinberg, 1999; Page *et al.*, 1999]. Thus, finding mappings between ontologies O^1 and O^2 can finally be viewed as node ranking and selection problems by random walks on the association graph G^{rw}.

3.1 Association Graph from Ontologies

Nodes in the Association Graph

We present here the composition of nodes in the association graph. As shown in Figure 1, ellipses and squares in the ontology graph G^i, where $i \in \{1, 2\}$, represent classes and properties, respectively. A node $v_{ia} \in V^{rw}$ in the association graph G^{rw} denotes a candidate mapping $< c_i^1, c_a^2 >$ (c denotes the class) or $< p_i^1, p_a^2 >$ (p denotes the property) because classes (properties) can only be matched to classes (properties). Here, the subscripts i and a denote the ith and ath node in the ontology graphs G^1 and G^2, respectively.

Edges in the Association Graph

To set edges between nodes in the association graph, we first extract the existing four main types of relations between classes and properties in the ontology. These relations are: (1) the *is-a* relation between two classes or two properties; (2) the *hasProperty* relation between a class and its properties; (3) the *hasDomain* relation between a property and its domain classes; and (4) the *hasRange* relation between a property and its range classes. Then, we set an edge between two nodes $v_{ia} \in V^{rw}$ and $v_{jb} \in V^{rw}$ if and only if the relations of entity pairs (e_i^1, e_j^1) and (e_a^2, e_b^2) are exactly the same. For example, in Figure 1 (G^{rw}), there is an edge between nodes v_{11} and v_{32} because the entities e_3^1 and e_2^2 are subclasses of entities e_1^1 and e_1^2, respectively.

Edge Weights Assignment

We compute the weight $w_{ia;jb}$ of edge $E_{ia;jb}^{rw}$ with the method described in Algorithm 1 for which the returning value is an $n \times n$ real diagonal matrix $\mathbf{W}$, where n is the number of nodes in graph G^{rw}. In Algorithm 1, the array **ES** contains similarity scores between classes and properties, which is computed with the cosine formula based on the classes' and properties' TFIDF vectors created from their descriptions. The descriptions of an entity include: *local name*, *label*, *comment*, *subterms*, *superterms* (both for classes and properties) and *properties* and *instances* (for classes) or *domain* and *range* (for properties). Pre-processing steps for the description *label* and *comment* consist of tokenization, lemmatization, stop word removal and translations as in [Cheatham and Hitzler, 2013].

Algorithm 1: Computing the weights of E^{rw}

Input: The number of entities in O^1 and O^2, m_1 and m_2; The set of nodes in the association graph, V; The array of entity similarity, ES;
Output: The weight matrix, W;

```
for all v^h ∈ V do
    e_i ← the entity v_1^h from O^1;
    e_a ← the entity v_2^h from O^2;
    for all v^k ∈ V do
        e_j ← the entity v_1^k from O^1;
        e_b ← the entity v_2^k from O^2;
        if hasSameRelation(e_i, e_j, e_a, e_b) then
            sim ← (ES_h + ES_k) / 2;
            if sim ≠ 0 then
                w_{ia;jb} ← exp(sim);
            else
                w_{ia;jb} ← 1.0/(m_1 + m_2);
        else
            w_{ia;jb} ← 1.0/|V|;
    w_{h;h} ← exp(ES_h);
return W;
```

In Algorithm 1, $exp()$ means the exponential function, and *hasSameRelation* is used for checking the equality of relation between e_i and e_j and between e_a and e_b in the orig-

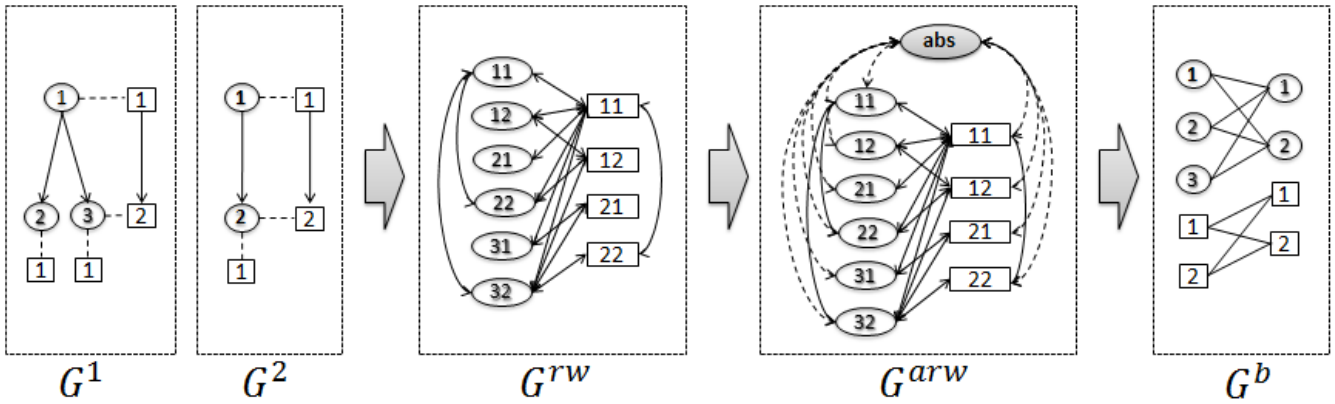

Figure 1: Association Graphs for Ontology Matching by Random Walks. $\mathbf{G}^{rw}$ is the association graph constructed from the two to-be-matched ontology graphs G^1 and G^2. $\mathbf{G}^{arw}$ is the graph augmented with an absorbing node based on the graph $\mathbf{G}^{rw}$. G^b is a bipartite graph used for extracting final mappings.

inal graphs. We can see that the larger the $w_{ia;jb}$ value is, the more likely the nodes v_h and v_k in G^{rw} represent mappings. It is different from the methods of assigning weights to edges in Similarity Propagation[Li *et al.*, 2009; Ngo *et al.*, 2012b], in which edge weights are empirically determined by the number of outgoing edges of source nodes. However, the weights of edges in our association graph are calculated based on not only the relations but also the similarities between entities. We call $w_{ia;jb}$ "affinity" between these two nodes in the association graph, and the affinity matrix **W** will be used to compute the transition probability matrix of the Markov chain in Sec.3.2.

3.2 Affinity-Preserving Random Walks

Imagine a walker who takes off from an arbitrary node in a graph and then successively visits new nodes by randomly selecting one of the outgoing edges according to a Markov transition probability and then repeats this process. In this way, we define a random walk on a graph.

Problem of Internet Democracy

Generally, to define the transition probability matrix for a Markov chain on a weighted graph, traditional approaches convert the affinity or weight matrix **W** to a row stochastic matrix by $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, where **D** is a diagonal matrix with entries $\mathbf{D}_{ii} = d_i = \sum_j \mathbf{W}_{ij}$. This practice can be found in PageRank [Page *et al.*, 1999], in which each outgoing hyperlink from a node i is row-normalized by $1/d_i$. This also can be interpreted as Internet Democracy [Langville and Meyer, 2004]. However, in our ontology matching task, there exists lots of pseudo candidate mappings (i.e., outlier nodes); this row normalization can strengthen the adverse effect of the pseudo mappings and weaken the true mappings (i.e., inlier nodes). For example, in Figure 1 (G^1, G^2), suppose the class c_1^1 and the property p_1^1 correspond to the class c_1^2 and the property p_1^2, respectively. The democratic normalization on G^{rw} scales up the affinities of outgoing edges of outlier nodes such as $<c_1^1, c_2^2>$ and $<p_1^1, p_2^2>$ when compared with the affinities of two inlier nodes $<c_1^1, c_1^2>$ and $<p_1^1, p_1^2>$ because the affinity sum of an outlier node is usually smaller than that of an inlier node. A similar problem also exists in Similarity Propagation [Li *et al.*, 2009; Ngo *et al.*, 2012b], in which the confidences from the parent node are evenly distributed to subnodes even if some of these subnodes are outliers.

Affinity Preserving in Random Walks

To preserve the original affinity relations while transforming the affinity matrix into the stochastic transition matrix for random walks, some other normalization methods should be used. We define a maximum degree $d_{max} = max_i d_i$ and construct an augmented graph G^{arw} (as shown in Figure 1 (G^{arw})) with an absorbing node v_{abs}, which soaks affinity $d_{max} - d_i$ out of all the unabsorbed nodes. In this special Markov chain, once the random walker reaches the absorbing node, he cannot walk out of there. Because each node in the graph G^{arw} has the same degree of d_{max}, its affinity matrix normalized by $1/d_{max}$ results in a stochastic matrix and corresponds to an *absorbing Markov chain* [Seneta, 2006], which preserves the original affinity relations of the association graph G^{rw}. We call this approach an "affinity-preserving random walk" as it is used in [Cho *et al.*, 2010], which aims at a graph matching problem in the image processing field, and the word "affinity" there means the similar degree between two nodes of an image. The transition matrix **P** and absorbing Markov chain of this *affinity-preserving random walk* are formulated as follows.

$$\mathbf{P} = \begin{pmatrix} \mathbf{W}/d_{max} & (\mathbf{1\text{-}d})/d_{max} \\ \mathbf{0}^T & 1 \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{abs}^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(n)T} & x_{abs}^{(n)} \end{pmatrix} \mathbf{P}, \qquad (2)$$

where $\mathbf{W}/d_{max}$ is a $|V^{rw}| \times |V^{rw}|$ substochastic matrix, and **1** is a $|V^{rw}| \times 1$ vector with all elements 1 and **0** with all elements 0. Because the steady state distribution of the absorbing Markov chain is always $(\mathbf{0}^T \quad 1)$, it cannot be used for node ranking in the same way as PageRank [Page *et al.*, 1999]. For the purpose of ranking nodes on the absorbing Markov chain, we denote $X^{(n)}$ as the node where a random

walker in the absorbing Markov chain of Eq.(2) stays at time n and define the conditional distribution $\bar{\mathbf{x}}^{(n)}$ as

$$\bar{\mathbf{x}}_{ia}^{(n)} = P(X^{(n)} = v_{ia} | X^{(n)} \neq v_{abs}) = \frac{\mathbf{x}_{ia}^{(n)}}{1 - x_{abs}^{(n)}}, \quad (3)$$

which refers to the distribution of unabsorbed node $v_{ia} \in G^{arw}$ at time n. If $\bar{\mathbf{x}}^{(n+1)} = \bar{\mathbf{x}}^{(n)} = \bar{\mathbf{x}}$, we call $\bar{\mathbf{x}}$ a *quasi-stationary distribution* of the absorbing Markov chain. This corresponds to a steady-state distribution in the Markov chain without absorbing nodes. $\bar{\mathbf{x}}$ is a $|V^{rw}| \times 1$ real vector, and the element $\bar{x}_h \in \bar{\mathbf{x}}$ represents the final probability that the random walker visits the unabsorbed node $V_{ia} \in V^{arw}$.

3.3 Finding mappings from walking results

Because each unabsorbed node in graph G^{arw} consists of entities $e_i^1 \in O^1$ and $e_a^2 \in O^2$, we say that the larger the visit probability of the unabsorbed node $V_{ia} \in V^{arw}$ is, the more likely is the mapping that it represents. To extract mappings from the steady-state probabilities of nodes in graph G^{arw}, we first construct a bipartite graph $G^b = (V^1, V^2; E^b)$, and the weight of $E_{ia} \in E^b$ is set to the visit probability of the unabsorbed node $V_{ia} \in V^{arw}$. This bipartite graph is displayed in Figure 1 (G^b). Suppose the number of entities in ontologies O^1 and O^2 is N_1 and N_2, respectively; then, an $N_1 \times N_2$ matrix S can be derived from the weighted bigraph G^b. In the matrix S, the value of element S_{ia} will be set to 0 if the edge $E_{ia} \notin E^b$ or the visit probability of the unabsorbed node $V_{ia} \in G^{arw}$ otherwise. We then use the Stable Marriage (SM) algorithm [Hopcroft and Karp, 1973; He, 2006], which has been widely used for finding 1:1 mappings in bigraphs, to extract the final mappings from the matrix S.

3.4 Mapping-Oriented Random Walks

In the affinity-preserving random walks presented in Sec.3.2, the 1:1 matching constraint has not been considered in the random walk process and has only been used in the last step of extracting final mappings. Because the calculation process of affinity-preserving random walks is iterative, taking the 1:1 matching constraint as a post-processing separate step may gradually submerge some true mappings and highlight some pseudo mappings. For example, in Figure 1, suppose the class entities c_1^1 and c_2^1 correspond to class entities c_1^2 and c_2^2, respectively, and the similarities of class entity pairs (c_1^1, c_1^2) and (c_2^1, c_2^2) are much larger than those of class entity pair (c_2^1, c_2^2). Then, during the affinity-preserving random walks, the true mapping $< c_2^1, c_2^2 >$ will be gradually submerged and the pseudo mapping $< c_1^1, c_1^2 >$ will be highlighted.

How, then, can we consider the 1:1 matching constraint in the affinity-preserving random walk? Inspired by the personalization approach widely used in webpage ranking methods [Haveliwala, 2002; Langville and Meyer, 2004], which strengthens the effects of inlier nodes in random walks, we achieve this goal by adopting a jump in the random walk: The random walker moves by traversing an edge with probability α or by performing a jump to some other nodes with probability $1 - \alpha$. The variable α represents the bias between the two possible actions, i.e., following an edge or jumping.

Algorithm 2: MORW for Ontology Matching

Input: The weight matrix, $\mathbf{W}$; The starting and maximum number of iteration, B and M; The mapping-oriented factor, α;
Output: The steady-state distribution, $\bar{\mathbf{x}}$;

```
d_max ← max_i d_i;
P ← W/d_max;
Initialize the starting probability x as uniform;
for i ← 1, 2, ..., M do
    if i < B then
        x̄ ← Px;
    else
        t ← selectingBySM( x̄ );
        r ← amplifyingElements( x̄, t );
        r ← r/∑ r_ai;
        x̄ ← αPx + (1 − α)r;
    x̄ ← x̄/∑ x̄_ai;
    if ||x̄ − x||^2 < ε then
        return x̄;
    x ← x̄;
return x̄;
```

The probability distribution when adopting the personalized jump is updated as follows.

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{abs}^{(n+1)} \end{pmatrix} = \alpha \begin{pmatrix} \mathbf{x}^{(n)T} & x_{abs}^{(n)} \end{pmatrix} \mathbf{P} + (1-\alpha)\mathbf{r}^T, \quad (4)$$

where a mapping-oriented jump vector $\mathbf{r}$ is added to the affinity-preserving random walk of Eq.(2). That is, we use the jumps for generating a biased random walk to the 1:1 matching constraint. Particularly, we first apply the Stable Marriage (SM) algorithm to select potential mappings (i.e., inlier nodes), then amplify the distribution of these inlier nodes by $e^{\mathbf{x} \times max\mathbf{x}}$ and normalize $\mathbf{x}$. This procedure is equivalent to attenuating small values of $\mathbf{x}$ and amplifying large values of $\mathbf{x}$ and made the unreliable mappings contribute insignificantly, which has been denoted by the *selectingBySM* function and *amplifyingElements* function in Algorithm 2. Accordingly, our *mapping-oriented random walk* is formulated by

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{abs}^{(n+1)} \end{pmatrix} = \alpha \begin{pmatrix} \mathbf{x}^{(n)T} & x_{abs}^{(n)} \end{pmatrix} \mathbf{P} + (1-\alpha) \begin{pmatrix} f(\mathbf{x}^{(n)T}\mathbf{W})^T & 0 \end{pmatrix}, \quad (5)$$

where $f(\cdot)$ denotes the mapping-oriented function incorporating 1:1 constraints. Note that this is a dynamic Markov chain for which the jump distribution is dynamically varying and dependent on the present distribution of $\mathbf{x}$, unlike conventional jumps in random walks [Haveliwala, 2002; Langville and Meyer, 2004]. As the $f(\cdot)$ generates a jump distribution close to a good solution, the subsequent random walks strengthen the distribution and move toward the final solution. The quasi-stationary distribution of the mapping-oriented random walk is computed using the iteration method as summarized in Algorithm 2, in which the parameter B is used to reduce the influence of disturbance from the initial uniform distribution $\mathbf{x}$.

4 Experiments

4.1 Data sets and evaluation criteria

Because the annual OAEI (Ontology Alignment Evaluation Initiative) campaign has become an authoritative contest in the area of ontology matching, we use the datasets from OAEI to evaluate our proposed approach. All of the ontologies in the dataset are described in OWL-DL language.

Development dataset: the Standard Benchmark 2012 dataset, which OAEI provides for participants to develop their systems before entering the competition, is used as the development dataset in our experiments. This dataset contains one reference ontology and 109 target ontologies. We use this dataset to set the parameters in our approaches.

Testing dataset: (1) the Benchmark-Biblio 2012 dataset, which contains one reference ontology and 94 target ontologies; and (2) the Benchmark-Biblioc 2013 dataset, which has five sub-datasets with one reference ontology and 93 target ontologies in each sub-dataset. We use these two datasets to evaluate the effectiveness of our approach.

In the matching scenario, each target ontology should be mapped to the reference ontology. We followed the standard evaluation criteria from the OAEI campaign, calculating the precision, recall and F-measure for each test. The version computed here is the harmonic mean of precision and recall.

4.2 Experiments on the Development dataset

We conduct experiments on the development dataset to check the correctness and rationality of our proposed approaches and select the final values of parameters in the approaches. The compared approaches are: (1) VSM (vector space model), which uses TFIDF vectors (as described in Sec.3.1.) to measure the similarities between classes and properties by cosine similarity and then applies Stable Marriage (SM) algorithm to extract the final mappings without random walk. This approach can be viewed as a baseline. (2) RNRW (Row-Normalized Random Walk), for which the transition probability matrix is a row-normalized affinity matrix. (3) APRW (Affinity-Preserving Random Walk), which is described in Sec.3.2. (4) MORW (Mapping-Oriented Random Walk), which is presented in Sec.3.4. The final settings of our approach are as follows: The maximum number of iteration M is 300, the number of starting iteration B is 50, the mapping-oriented factor α is 0.7, and the minimum error ε is 1E-60. Table 1 reports the comparison results.

Approach	***Prec.***	***Rec.***	***F-m.***
VSM	0.866	0.738	0.773
RNRW	0.764	0.722	0.730
APRW	0.856	0.815	0.823
MORW	0.864	0.823	0.830

Table 1: Experiments on the Development dataset.

As shown in Table 1, the F-measure reached 77.3% even in the case of the vector space model (VSM) for matching. In our random walk-based approaches, the similarities, which have been employed by VSM, are used to create the transition probability matrix. When the 1:1 matching constraint has not been considered in the process of random walk (i.e., in the APRW approach), the matching F-measure is 82.3%. When we take this constraint into account during the random walk (i.e., in the MORW approach), the F-measure improved by approximately 1% on the whole dataset. The F-measure of RNRW is even lower than that of VSM, and this shows that the settlement of the Internet democracy problem is very important in our approach.

	Benchmark-Biblio 2012			Benchmark-Biblioc 2013		
	Prec.	***Rec.***	***F-m.***	***Prec.***	***Rec.***	***F-m.***
VSM	0.875	0.724	0.766	0.872	0.722	0.764
RNRW	0.647	0.647	0.647	0.670	0.670	0.670
APRW	0.858	0.859	0.858	0.856	0.856	0.856
MORW	0.866	0.866	0.866	0.864	0.864	0.864

Table 2: Experiments on the test datasets.

4.3 Experiments on the test dataset

We conduct experiments on the test datasets to evaluate the effectiveness of our approach. We use the same model settings as in the experiment conducted on the development dataset. The experimental results are presented in Table 2.

As we can see from Table 2, due to the improper row normalization, RNRW perform even worse than the VSM baseline by about 10 percent in F-measure on both test datasets. However, by introducing the absorbing node in the association graph and incorporating the 1:1 matching constraints, both APRW and MORW outperform the VSM baseline by large margins. MORW perform slightly better than APRW consistently on both datasets. To determine whether the difference is significant, we conduct significance test using the sample evaluation method, which was proposed by Van Hage *et al.* [2007] to assess ontology-matching performances. The test results on both datasets show that the performance difference between APRW and MORW is significant at a confidence level of 95% and justify the necessity of the additional jumping technique used in MORW.

4.4 Comparison with OAEI Participants

We compare our final ontology matching approach MORW with other multi-strategy matching systems on the two testing datasets. Figure 2 shows the results of the top five matching systems according to their F-measure on the Benchmark-Biblio 2012 dataset and Benchmark-Biblioc 2013 dataset.

As shown in Figure 2, our MORW outperforms most of the participants except for the MapSSS system, for which the F-measure is 0.87 in 2012, and the YAM++ and CroMatcher systems, for which the F-measures are 0.89 and 0.88 in 2013, respectively. Unlike MapSSS, our approach does not use any external resources such as Google queries in its current version. In the YAM++ approach, the gold standard datasets that are taken from the Benchmark dataset published in OAEI 2009 are used to generate training data to train a decision tree classifier. And in the classifying phase, each pair of elements from two ontologies is predicted as being matched or not according to its attributes. However, our approach is entirely an unsupervised ontology matching approach, but it

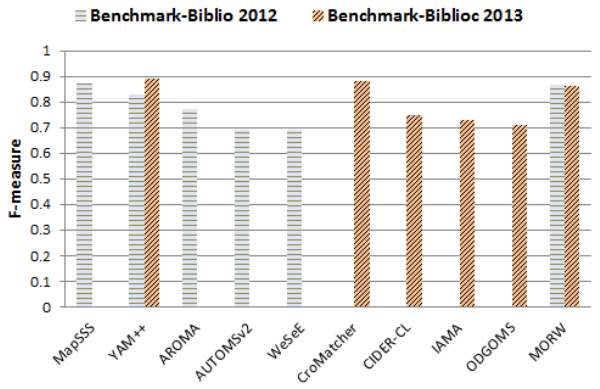

Figure 2: Comparison with other OAEI systems.

does not exclude using training data to help initialize the transition probability of the Markov chain to further improve the performance. Unfortunately, despite consulting a great deal of materials, we do not know any of the details for the CroMatcher approach, at least at the present time.

5 Related work

In addition to the graph-based approaches described in the introductions, there are also other approaches such as linguistic matching, hybrid matching, machine learning-based matching and probabilistic matching.

Linguistic matching is the construction of virtual documents. V-Doc [Qu *et al.*, 2006] is an example of a linguistic matcher. It exploits the RDF graph to extract the description information from three types of neighboring entities, including subject neighbors, predicate neighbors and object neighbors. Hybrid matching uses linguistic information (e.g., name, label, and description) and structural information (e.g., key properties and taxonomic structure) to find correspondences between entities. For example, Falcon-AO++ [Jauro *et al.*, 2014] is a hybrid matching system that supports the interactive contribution of a domain expert in the matching process. There are several works that exploit the machine learning techniques for ontology matching. Doan *et al.* [2003] created a well-known machine learning-based ontology mapping system, in which the joint probability distribution of the concepts is measured to find similar concepts. In [Eckert *et al.*, 2009], string-based, linguistic and structural measures (in total 23 features) were used as inputs to train an SVM classifier to align ontologies. Mao *et al.* [2010] proposed a neural network-based approach to search for a global optimal solution that can satisfy ontology constraints to the greatest extent possible to find mappings. CSR (Classification-based learning of Subsumption Relations) is a generic method for automatic ontology matching between concepts based on supervised machine learning [Spiliopoulos *et al.*, 2010]. It specifically focusses on discovering subsumption correspondences. SMB (Schema Matcher Boosting) is an approach to combining matchers into ensembles [Gal, 2011]. It is based on a machine learning technique called boosting, which is able to select (presumably the most appropriate) matchers that participate in an ensemble. Some researchers have explored using a probabilistic scheme for ontology matching. iMatch [Albagli *et al.*, 2009] is a probabilistic interactive ontology matching system based on Markov networks. It first builds a specific Markov network for a given pair of ontologies, and the topology of the network is defined based on constraints and rules; then, it uses initial match distributions to initialize the evidence potentials of the network, and ultimately, it exploits probabilistic reasoning in the Markov network to compute the final alignment. CODI is a probabilistic-logical ontology matching system [Niepert *et al.*, 2010; Huber *et al.*, 2011]. It is based on Markov logic networks and provides a declarative framework for matching classes, properties and individuals. The matching problem is reduced to a maximum a posteriori inference in the Markov logic network, which is in turn solved by using integer linear programming.

Our approach belongs to the graph-based ontology matching approaches. The main characteristics of our approach are the following: (1) edge weights in our association graph are measured by the affinity between entities rather than being assigned based on experience as in other graph-based approaches such as Similarity Propagation[Li *et al.*, 2009; Ngo *et al.*, 2012b]; (2) our *affinity-preserving random walk* approach differentiates the potential true and false mappings in order to weaken the adverse effect of pseudo-mapping nodes, while in previous graph-based approaches, these mappings have been treated equivalently; (3) in the random walk view, our *mapping-oriented random walk* approach adopts the personalization strategy of PageRank algorithms [Haveliwala, 2002] by dynamically adding jumps for the matching constraints, which has only been used in the final step in most of the previous approaches. This can simultaneously update and exploit the confidences of candidate mappings to further improve the performance.

6 Conclusions

In this paper, we propose a novel ontology matching approach based on random walks. In particular, inspired by the personalization strategy of PageRank algorithms, we propose an affinity-preserving random walk to avoid the Internet democracy problem in our task. Furthermore, to incorporate the matching constraints into the random walks dynamically, a mapping-oriented random walk approach is proposed. We evaluate our approach on the public datasets from OAEI campaigns, and the experimental results show that the matching quality of our approach nears the highest levels. In future work, we plan to integrate external resources such as search engines and labeled data into our approach to help with the calculations of the transition probability matrix.

Acknowledgments

This research is supported by National Key Basic Research Program of China (No.2014CB340504) and National Natural Science Foundation of China (No.61375074,61273318). The corresponding authors of this paper are Baobao Chang and Zhifang Sui.

References

Sivan Albagli, Rachel Ben-Eliyahu-Zohary, and Solomon E Shimony. Markov network based ontology matching. In *Proceedings of the 21st international jont conference on Artifical intelligence (IJCAI)*, pages 1884–1889. Morgan Kaufmann Publishers Inc., 2009.

Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In *The Semantic Web–ISWC 2013*, pages 294–309. Springer, 2013.

Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 492–505, Berlin, Heidelberg, 2010. Springer-Verlag.

AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevy. Learning to match ontologies on the semantic web. *The VLDB JournalThe International Journal on Very Large Data Bases*, 12(4):303–319, 2003.

Kai Eckert, Christian Meilicke, and Heiner Stuckenschmidt. Improving ontology matching using meta-level learning. In *The Semantic Web: Research and Applications*, pages 158–172. Springer, 2009.

Jérôme Euzenat and Pavel Shvaiko. Ontology matching. 2013.

Avigdor Gal. Uncertain schema matching. *Synthesis Lectures on Data Management*, 3(1):1–97, 2011.

Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 517–526, New York, NY, USA, 2002. ACM.

Andreas He. An iterative algorithm for ontology mapping capable of using training data, 2006.

John E Hopcroft and Richard M Karp. An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

Wei Hu, Ningsheng Jian, Yuzhong Qu, and Yanbing Wang. Gmo: A graph matching for ontologies. In *Proceedings of K-CAP Workshop on Integrating Ontologies*, pages 41–48, 2005.

Jakob Huber, Timo Sztyler, Jan Noessner, and Christian Meilicke. Codi: Combinatorial optimization for data integration–results for oaei 2011. *Ontology Matching*, page 134, 2011.

Fatsuma Jauro, S B Junaidu, and S E Abdullahi. Falcon-ao++: An improved ontology alignment system. *International Journal of Computer Applications*, 94(2):1–7, 2014.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

Amy N. Langville and Carl D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1:2004, 2004.

Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *Knowledge and Data Engineering, IEEE Transactions on*, 21(8):1218–1232, 2009.

Ming Mao, Yefei Peng, and Michael Spring. An adaptive ontology mapping approach with neural network based constraint satisfaction. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1):14–25, 2010.

Ming Mao, Yefei Peng, and Michael Spring. Ontology mapping: as a binary classification problem. *Concurrency and Computation: Practice and Experience*, 23(9):1010–1025, 2011.

Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.

DuyHoa Ngo, Zohra Bellahsene, and Remi Coletta. A flexible system for ontology matching. In *IS Olympics: Information Systems in a Diverse World*, pages 79–94. Springer, 2012.

DuyHoa Ngo, Zohra Bellahsene, et al. Yam++-a combination of graph matching and machine learning approach to ontology alignment task. *Journal of Web Semantics*, 16, 2012.

Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *AAAI*. Citeseer, 2010.

Natalya F Noy and Mark A Musen. Anchor-prompt: Using non-local context for semantic matching. In *Proceedings of the workshop on ontologies and information sharing at the international joint conference on artificial intelligence (IJCAI)*, pages 63–70, 2001.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

Yuzhong Qu, Wei Hu, and Gong Cheng. Constructing virtual documents for ontology matching. In *Proceedings of the 15th international conference on World Wide Web (WWW)*, pages 23–31. ACM, 2006.

Eugene Seneta. *Non-negative matrices and Markov chains*. Springer, 2006.

Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):158–176, 2013.

Vassilis Spiliopoulos, George A Vouros, and Vangelis Karkaletsis. On the discovery of subsumption relations for the alignment of ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1):69–88, 2010.

Willem Robert Van Hage, Antoine Isaac, and Zharko Aleksovski. Sample evaluation of ontology-matching systems. In *EON*, volume 2007, pages 41–50, 2007.