Time-aware Link Prediction in RDF Graphs

Jaroslav Kuchař, Milan Dojchinovski and Tomas Vitvar
Web Intelligence Research Group, Faculty of Information Technology,
Czech Technical University in Prague, Prague, Czech Republic

Keywords: Link Prediction, Web APIs, Temporal Information, Semantics, Tensor Factorization.

Abstract: When a link is not explicitly present in an RDF dataset, it does not mean that the link could not exist in reality. Link prediction methods try to overcome this problem by finding new links in the dataset with support of a background knowledge about the already existing links in the dataset. In dynamic environments that change often and evolve over time, link prediction methods should also take into account the temporal aspects of data. In this paper, we present a novel *time-aware link prediction* method. We model RDF data as a tensor and take into account the time when RDF data was created. We use an ageing function to model a retention of the information over the time; lower the significance of the older information and promote more recent. Our evaluation shows that the proposed method improves quality of predictions when compared with methods that do not consider the time information.

1 INTRODUCTION

Over the last few years the number of published RDF datasets in the Linked Data cloud has grown significantly. One of the key Linked Data publishing principles is to use URI references to identify Web resources and links between them[1]. Such link are usually defined at the time of creation of the datasets and they are often not updated. However, over the time the links can get old and loose their significance. Link prediction algorithms, on the other hand, find new links in datasets that are not explicitly present but they implicitly exist due to existing structural patterns.

An increasing amount of datasets and their evolution over time introduce another dimension to link prediction methods. In this paper we develop a novel method that is able to predict links in a single dataset that uses i) *the creation time of the links*, and ii) *the existing structural patterns* in the dataset. We call this method a *time-aware link prediction*.

We validate the method on a dataset from ProgrammableWeb [2], a leading Web APIs and mashup directory, that allows developers to publish information about their Web APIs and mashups and to join a social network of developer fellows. At the time of creating a Web API or a mashup in the directory, a developer provides various technical and functional descriptions such as categories, tags and defines links between APIs and mashups. A link prediction method applied on the dataset from ProgrammableWeb may be used to find links to other categories, tags or Web APIs based on structural patterns in which the Web APIs, mashups or developers occur. However, such method would ignore the fact that a Web API or a mashup can be outdated. Our link prediction method provides more precise results as it can effectively combine time information with structural patterns. We use i) *tensors* as an underlying mechanism to model RDF data, ii) *time information and an ageing function* to model the age of the data and iii) a *tensor factorization technique* to evaluate an existence of new links. We adopted a widely used ageing to simulate the loss of the links' significance; decrease the impact of older links and promote the more recent ones. Our assumption is that older links are less important due to their age, however, they can still have an influence on the link prediction due to structural patterns. We evaluate the method on a real-world dataset from the Web services domain and we present its performance and capabilities.

The paper is structured as follows. Section 2 describes the time-aware link prediction method, its notations, definitions and the supporting algorithm. Section 3 describes several experiments we conducted to evaluate its performance and capabilities. In Section 4 we discuss various aspects of the method. In Section 5 we give an overview of the related work, and finally,

[1] http://www.w3.org/DesignIssues/LinkedData.html

[2] http://www.programmableweb.com/

Kuchař J., Dojchinovski M. and Vitvar T..
Time-aware Link Prediction in RDF Graphs.
DOI: 10.5220/0005428403900401
In *Proceedings of the 11th International Conference on Web Information Systems and Technologies* (WEBIST-2015), pages 390-401
ISBN: 978-989-758-106-9

Table 1: Example of modelling data.

(a) Tensor $\mathcal{Y}$ model without ageing

	s_1	s_2	s_3	s_4
m_1	0	0	0	0
m_2	$\mathbf{1}_{(t_0-t_3)}$	0	$\mathbf{1}_{(t_0-t_1)}$	0
m_3	0	0	0	0
m_4	0	$\mathbf{1}_{(t_0-t_{15})}$	0	0

(b) Tensor $\mathcal{X}$ model with ageing

	s_1	s_2	s_3	s_4
m_1	0	0	0	0
m_2	$\mathbf{0.97}_{(t_0-t_3)}$	0	$\mathbf{0.99}_{(t_0-t_1)}$	0
m_3	0	0	0	0
m_4	0	$\mathbf{0.86}_{(t_0-t_{15})}$	0	0

Section 6 concludes the paper and presents the future work.

2 TIME-AWARE LINK PREDICTION METHOD

2.1 Definitions

Tensor. *A multi-dimensional array of numerical values* (Kolda and Bader, 2009). The *order* of the tensor is the number of dimensions that the tensor uses. In our method we use a tensor of order three denoted by $\mathcal{Y}^{I \times J \times K}$, where $I, J, K \in \mathbb{N}$ and $I = J$. The (i, j, k) element of a third-order tensor is denoted as y_{ijk}.

Information Ageing. *A process of retention of information in a memory over time*. We represent the relation between time and retention using a *forgetting curve* (Ebbinghaus, 1913); defined as $R = e^{-\lambda T}$ where R is the memory retention, T is the amount of time since the information was received and $1/\lambda$ is the strength of the memory.

Based on the definition of the forgetting curve, we propose an ageing function

$$\mathcal{A}(t_0) = \mathcal{A}(t_x) * e^{-\lambda t}; t_0 > t_x, t = t_0 - t_x \qquad (1)$$

where $\mathcal{A}(t_0)$ is the amount of information at the time t_0, $\mathcal{A}(t_x)$ is the amount of information at the time t_x when the information was created, λ is ageing/retention factor and t is the age of the information. The information ageing is influenced by the the λ parameter as the strength of the memory. The higher the value of the λ parameter is, the faster the loss of information is. Similarly, the older the information is, the lower is the amount of held information.

Note that Linked Data community has adopted several approaches to represent temporal information (Rula et al., 2012; Gutirrez-Basulto and Klarman, 2012). In this paper we use a single *starting time point* t_x which defines the existence of the link, i.e. the link exists since t_x (see Section 4 for discussion). We refer to this time as the creation time. We have no information about the duration of the existence of the link and we cannot conclude whether it is still valid (Open World Assumption).

2.2 Tensor-based Model with Temporal Information

Simple graph structures can be modelled as matrices, which is preferred for graph structures with one type of links. However, since RDF data contain more than one type of links, we use a third-order tensor notation, which was proposed in (Nickel et al., 2011). We can project the third-order tensor as a set of incidence matrices, where each matrix contains only links between entities for a corresponding type of the link.

Let $\mathcal{Y} \in \{0,1\}^{N \times N \times M}$ be a tensor representing an RDF dataset. The tensor consists of two identical dimensions N representing a domain of entities (concepts and instances) in the dataset, and the third dimension M representing a domain of link types (properties) that explicitly exist in the dataset. The tensor element $y_{ijk} = 1$, if the i-th entity has link of a type k with the j-th entity, for $i, j \in \langle 0, N)$ and $k \in \langle 0, M)$. Otherwise, the tensor element $y_{ijk} = 0$. Each tensor element in the model has a value of 1 or 0 if a link between two entities exists or does not exist, respectively.

In this paper, we propose an extension of this model to include also temporal information. We focus on the situation, when the creation time of the links is available (see Section 4 for discussion). We use this information to modify the initial tensor $\mathcal{Y}$ such that values of tensor elements are reduced with respect to the creation time of the corresponding link. Let $\mathcal{X} \in \mathbb{R}^{N \times N \times M}$ be a tensor at the time t_0. We then compute a value of a tensor element x_{ijk} using the ageing function (1) as follows

$$x_{ijk} = y_{ijk} * e^{-\lambda t} \qquad (2)$$

where $y_{ijk} \in \{0,1\}$ is the initial value of the tensor element, λ is the ageing factor and t is the link's age computed as a distance of the link's creation time and the time t_0 (see Section 2.1 for additional details about the ageing function).

Example 1. Consider an RDF dataset consisting of the four instances of concepts *ls:Mashup*

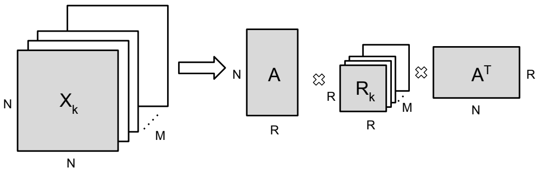


Figure 1: Visualization of RESCAL (Nickel et al., 2011).

($m1, m2, m3, m4$) and *wl:Service* ($s1, s2, s3, s4$), and three links *ls:usedAPI* that indicate usages of the Web APIs in the mashups, i.e. ($m_2 \xrightarrow{t_0-t_3} s_1, m_2 \xrightarrow{t_0-t_1} s_3, m_4 \xrightarrow{t_0-t_{15}} s_2$). In this formula, each arrow indicates the age of the link as the number of weeks since t_0. For example, $m_4 \xrightarrow{t_0-t_{15}} s_2$ indicates that the link was created 15 weeks ago.

Table 1 shows this information modelled as a tensor, both, with and without ageing (in this example we set parameter $\lambda = 0.01$). Note that the link between the mashup m_4 and the service s_2 has a lower value due to the fact that this link was created earlier than the other two.

2.3 Learning Hidden Latent Factors

We use a tensor factorization technique to perform structural analysis of an RDF dataset. We propose an extension of the RESCAL approach (Nickel et al., 2011) which uses the time information. Each incidence matrix $\mathbf{X}_k$ of a tensor is factorized as

$$\mathbf{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^T, k = 0...M \tag{3}$$

where $\mathbf{A}$ is a matrix $N \times R$ which models a participation of an entity in a latent factor R, and $\mathbf{R}_k$ is a matrix $R \times R$ that models interactions of latent factors for the k-th relation (Figure 1). The R is a configurable parameter of the factorization algorithm. It indicates the number of latent factors to be learned.

The matrix $\mathbf{A}$ and the matrices $\mathbf{R}_k$ are computed by solving the minimum optimization problem

$$\min_{\widehat{\mathbf{X}_k}} \| \mathbf{X}_k - \widehat{\mathbf{X}_k} \|_F \text{ , where } \widehat{\mathbf{X}_k} = \mathbf{A}\mathbf{R}_k\mathbf{A}^T \tag{4}$$

Although there exist other tensor factorization algorithms, RESCAL (Nickel et al., 2011) is the most suitable method for an analysis of multi–relational data and link prediction tasks, it scales well for larger datasets and it shows good performance (Nickel et al., 2012).

In our extension of the algorithm, we use a tensor with elements as real positive numbers; lower values for older links and higher values for newer links. By using this tensor, latent factors can learn regularities in the model while reconstructed values are approximately the same as the original values. The extra non-zero values in the reconstructed matrices reflect the temporal information and the higher values are influenced by the higher values in the original model. The higher values represent the predicted links influenced by the recent links in the original model.

	s_1	s_2	s_3	s_4
m_1	0	0	0	0
m_2	$0.95_{(t_0-t_3)}$	*0.04*	$0.98_{(t_0-t_1)}$	0
m_3	0	0	0	0
m_4	*0.11*	$0.83_{(t_0-t_{15})}$	*0.18*	0

Figure 2: Example of reconstructed tensor $\mathcal{X}$ ($R = 3$).

2.4 Time-aware Link Prediction

The *link prediction* task evaluates a possible existence of a link between a pair of entities by using structural patterns in the dataset. Our *time-aware link prediction* task, on the other hand, evaluates a possible existence of a link between two entities while taking into account the age of explicit links in the dataset as well as structural patterns in the dataset.

To evaluate an existence of a link between i-th and j-th entity we do a reconstruction $\widehat{\mathbf{X}_k} = \mathbf{A}\mathbf{R}_k\mathbf{A}^T$ of a matrix $\mathbf{X}_k$ for a link of type k. The algorithm solves a minimum optimization problem with goal to predict links of type **k** from domain M from the **i**-th entity from domain N. Note that in the following algorithm the terms *source entity*, *link* and *target entity* refer to the RDF terminology *subject*, *predicate* and *object*, respectively.

Inputs:

- An RDF dataset where each link contains information when the link was created.
- Ageing constant λ.
- A link of type **k** and an entity **i** as a source of links.
- A maximum number of target entities L.

Outputs:

- A set of Top-L entities as targets of links.

Algorithm:

1. Model a tensor $\mathcal{X}$ for the input RDF dataset and the ageing constant λ.
2. Compute factorization for the tensor $\mathcal{X}$ with the extended RESCAL algorithm (see Section 2.3).
3. Reconstruct a matrix $\widehat{\mathbf{X}_k}$ using the latent factor $\mathbf{R}_k$ and a matrix $\mathbf{A}$, where k indicates a link type in the query.

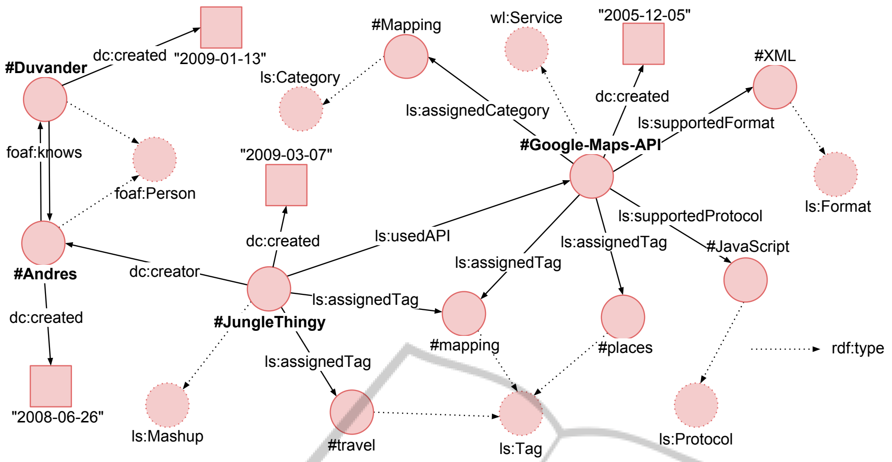


Figure 3: Excerpt from the extended Linked Web APIs dataset.

4. Read values x_{ijk} for the **i**-th row and each **j**-th column. The values indicate whether a link between the **i**-th entity and entity in the **j**-th column should exist.
5. Sort values in decreasing order and return Top-L values. These values indicate target entities that should be linked with the source entity using the link type k. Note that the Top-L entities can also be evaluated by comparing $x_{ijk} > \theta$, where θ is some threshold.

Example 2. Consider data from Example 1 as an input RDF dataset. It contains only one type of link ($k = usedAPI$) to make it clear. Tensor $\mathcal{X}$ on Table 1 corresponds to the first step of the algorithm for $\lambda = 0.01$. The second step factorizes tensor to matrices $\mathbf{A}, \mathbf{R}_k$ and the third step provides approximation of the tensor. Example of the reconstructed matrix $\widehat{\mathbf{X}_k}$ is on Figure 2 ($R = 3$). For entity $i = m_4$ the corresponding row contains three possible candidates as new links (s_3, s_1, s_4) sorted decreasingly by the reconstructed value. From the list of candidates can be selected either a set of Top-L elements or elements with the value above predefined threshold θ.Please note that the higher value for s_3 was influenced by the existing link with higher value, that was created more recently than the second one.

3 EVALUATION AND EXPERIMENTS

In this section we demonstrate the time-aware link prediction method on the real-world dataset from ProgrammableWeb.

The questions we address in experiments are as follows:

- *How temporal aspects influence the link prediction?*
- *How the evolution of dataset structure influences the link prediction?*

On several experiments, we evaluate the quality of the proposed method when compared with a set of baseline algorithms. The first experiment shows the difference of the proposed time-aware link prediction and a link prediction without temporal information. The following two experiments clarify the connection between predicted links, the time information and the structure of the dataset.

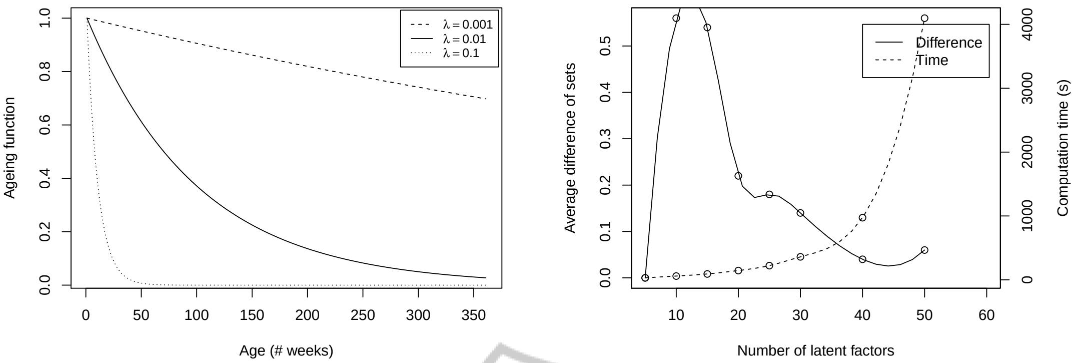
3.1 Linked Web APIs Dataset

For evaluation purposes, we created extended version of the *Linked Web APIs* dataset. The dataset is an RDF representation of the ProgrammableWeb[3] directory, the largest mashup and Web APIs directory. It contains information about developers, mashups they created and Web APIs they used, together with categories they belong to. In addition, the dataset has information about tags assigned to each mashup and a Web API, formats and protocols that Web APIs support. We also collected information about the time when users, mashups or Web APIs appeared in the directory for the first time. The dataset contains information from June 2005 till the end of March 2013, it has in total 22 286 entities, 8 types of links and contains approx. 123 000 links.

The dataset (Figure 3) uses several well know ontologies and vocabularies: FOAF[4] ontology (*prefix foaf*) - concept *foaf:Person* describes users and property *foaf:knows* describes a social relationship

[3] http://www.programmableweb.com/

[4] http://xmlns.com/foaf/spec/



(a) Ageing function

(b) Number of Latent factors

Figure 4: Experiments settings.

between users, WSMO-lite (Vitvar et al., 2008) ontology (*prefix wl*)- concept *wl:Service* describes Web APIs, Dublin Core[5] vocabulary - property *dc:creator* describes relation between a user and a mashup, and property *dc:created* indicates creation date of a mashup, a user or a Web API, SAWSDL (Kopecky et al., 2007) vocabulary (*prefix sawsdl*) - property *sawsdl:modelReference* describes a tag or a category of a Web API or a mashup. Additionally, we create new concepts and properties (*prefix ls*): *ls:Protocol* that identifies a protocol, *ls:Format* that identifies data format, and ls:Tag and ls:Category which identify a tag or a category respectively. We also create following new properties: *ls:usedAPI* - between concepts *ls:Mashup* and *wl:Service*, *ls:supportedFormat*, *ls:supportedProtocol* - between concepts *wl:Service* and *ls:Format* or *ls:Protocol*, *ls:assignedTag* and *ls:assignedCategory* - between concepts *wl:Service/ls:Mashup* and *ls:Tag/ls:Category*.

3.2 Experiments Settings

Implementation. We implemented the proposed method in *R*. It contains functionalities to construct a tensor with temporal aspects, RESCAL factorization algorithm, link prediction method and a running example[6].

Time Information. Our dataset does not contain the time information for each link. Therefore, we derive this information from $< n, dc : created, t_{cn} >$, where n represents a mashup, a Web API or a person and t_{cn} denotes the time the entity was created. Since all entity links are created in our dataset at the same time as the entity is created, we propagate t_{cn} as a creation time to all the links of the entity n.

Snapshots. For purposes of analysing data over different time periods we prepared 22 snapshots of the dataset. The first snapshot contains data from June 2005 until January 2008. It contains approx. 21 000 links which is a significant portion of the total number of links while it is a sufficient information for the link prediction. We then created subsequent snapshots with a step of 3 months where each snapshot always contains the data of a previous snapshot. In order to compare capabilities of the time-aware link prediction and the link prediction that does not use time information we modelled all 22 snapshots as tensors with and without time information. The ageing function parameter t_0 (see Formula (1)) denotes the end of a snapshot.

Setting the Ageing Constant. In the experiments, we set the ageing constant empirically to $\lambda = 0.01$ and the age period t in weeks. Figure 4(a) depicts the influence of the ageing function for different λ. Value $\lambda = 0.01$ provides a distribution of values over the whole seven years period. Note that a higher λ value (i.e. $\lambda = 0.1$) promotes less than the last 50 weeks while a lower λ value (i.e. $\lambda = 0.001$) does not provide significant change of values over the period. This is a configurable parameter that can be used to control the forgetting rate and it depends on specific requirements and dataset. Since we want all data in the dataset to participate in our experiments, the value $\lambda = 0.01$ provides us with the best setting. The results from the evaluation also supports this setting in terms of overall quality of the predictions.

Setting the Tensor Factorization. In the tensor factorization, we experimentally set the number of latent factors to 40. We terminate the factorization when a change of the factor matrices between two iterations is < 1. This is a terminating condition for the minimum optimization problem which means that the solution found during the iteration will not change in

[5] http://dublincore.org/documents/

[6] https://github.com/jaroslav-kuchar/Time-Aware-Link-Prediction

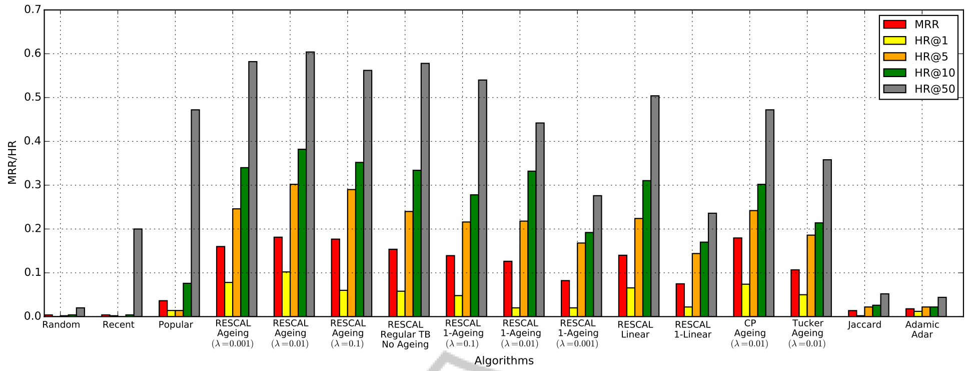


Figure 5: Mean Reciprocal Rank (MRR), HitRatio at top-k (HR@k).

subsequent iterations. Figure 4(b) depicts the impact of various settings on the method. We performed 10 runs on the same model and measured the difference of predicted sets of links. The same figure also illustrates a computation time on a computer with 1,6 GHz Intel Core i5 and 4GB RAM. Note that in this paper we do not focus on the performance and scalability of the algorithm. We refer the reader to (Nickel et al., 2011) for more details on the performance of the RESCAL factorization.

3.3 Evaluation

In this section, we describe the results from the experimental evaluation where the goal is to measure the quality of the time-aware link prediction. We created two sets, namely a *training set* and a *testing set*, from the whole dataset. We randomly selected 1% of the newest links from the last snapshot (the last 3 months) and put them to the testing set. The rest of the data we put to the training set. We performed repeated random sub-sampling cross-validation.

We evaluated our method (including different functions and parameters for ageing) compared to the following set of algorithms.

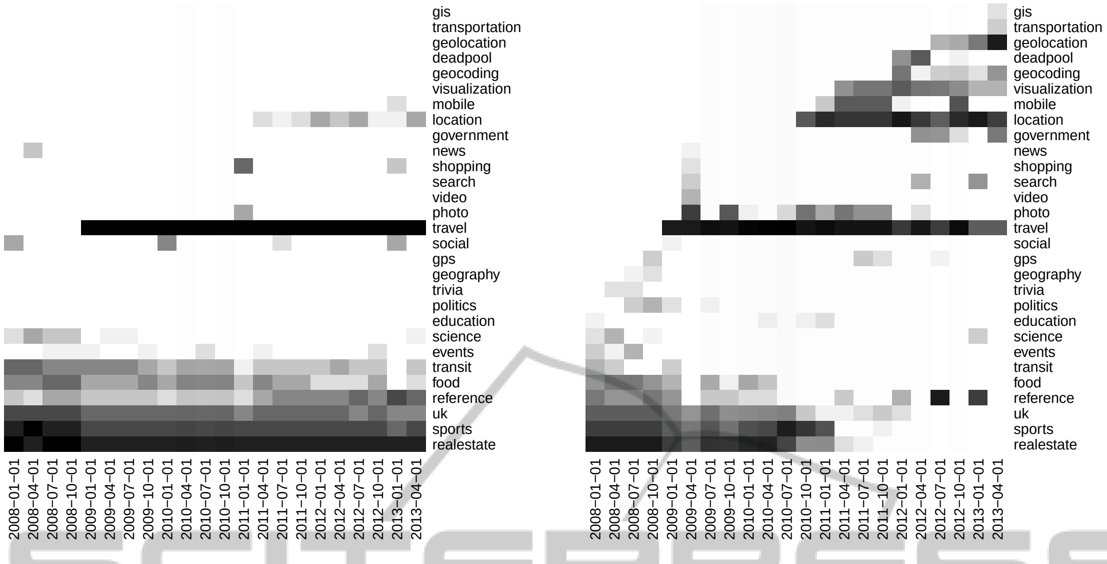
- *Random:* for each source of a link in the testing set, randomly choose a set of targets that correspond to the type of the link. For example, for a *Mashup* and a link *usedAPI* it randomly chooses a set of *Web APIs*.
- *Recent:* select targets from the testing set that are connected to the newest links in the training set.
- *Most Popular:* select targets from the testing set that are connected to the highest number of links in the training set.
- *Regular TB Link Prediction:* a tensor model without ageing and the original RESCAL tensor factorization.
- *Time-aware Link Prediction with Ageing:* our proposed method with different values of λ parameter for ageing function. "*Linear*" decreases importance of older links linearly over the whole time period, "$1-Ageing$" and "$1-Linear$" promotes older links.
- *CP and Tucker:* tensor decomposition CP (CANDECOMP/PARAFAC) and Tucker (Kolda and Bader, 2009) using tensor model with ageing function and $\lambda = 0.01$.
- *Jaccard and Adamic Adar:* baseline graph based methods for link prediction in social networks (Oyama et al., 2011) that use node neighbourhoods to predict new links.

Note that the *Recent* and *Most Popular* are exploited as recommendation methods in the ProgrammableWeb service repository.

Since we only have one relevant target for each testing item, and we measure a position of a predicted link, we did not perform evaluation related to Precision and Recall. Instead, we measured Mean Reciprocal Rank (MRR), which is appropriate for evaluation tasks with a single target. It is computed as a reciprocal value of a position at which the relevant target was evaluated and is averaged across all testing items (TI): $MRR = \frac{1}{|TI|}\sum_{i=1}^{|TI|} 1/position_i$.

The second metric we evaluate is HitRatio at top-k ($HR@k$) that indicates whether the relevant link occurs in the top-k predicted links. It is computed as $HR@k = \frac{1}{|TI|}\sum_{i=1}^{|TI|} hit_i^k$, where $hit_i^k = 1$ if the relevant link is in top-k predicted links, otherwise it is 0.

Figure 5 shows results from the evaluation. *Random* neither works with structural nor temporal information and has the lowest values for all metrics. *Recent* has slightly better results since it takes into account temporal aspects. Taking into account popularity leads to better results with *Most Popular*. *Regular Link Prediction* has good results since it considers the data structure. *Time-Aware Link Predictions* based on



(a) Without Ageing (b) With Ageing

Figure 6: Visualization of positions for each snapshot.

Linear, 1 − *Linear* or 1 − *Ageing* do not show better results than the Regular Link prediction since they do not reflect properly temporal aspects of links in the dataset. *Jaccard and Adamic Adar* does not perform well since they consider only information about the closest neighbourhood of each node in graph and they do not take into account types of nodes or semantics of links. *CP decomposition* achieved comparable results with RESCAL in terms of MRR but lower results in HR@k. *Tucker* decomposition has good results since it takes into account structure but does not have better results than *Regular Link Prediction* with RESCAL. Our time-aware link prediction based on RESCAL ($\lambda = 0.01$) outperforms other baseline algorithms in MRR and HR@1, HR@5, HR@10. It is able to predict links on better positions (lower k) than the other algorithms. In the following experiments, we focus on the *Time-Aware Link Predictions* with ageing function ($\lambda = 0.01$).

3.4 Significance of Time-aware Link Prediction

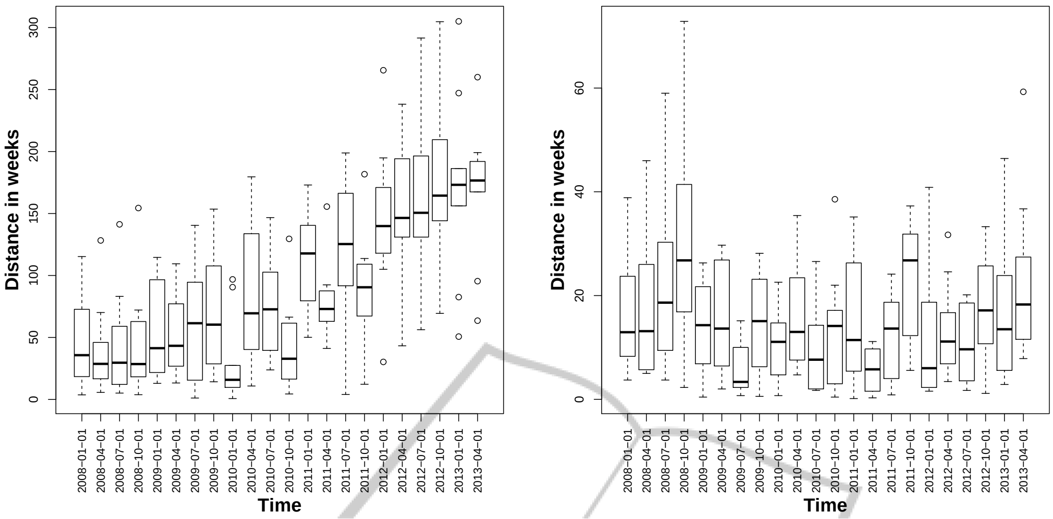
In this experiment we test how the time information influences items and their position in a list of top-L predicted links. To study the influence of time, we focused on a simple tagging task. The goal is to find a set of tags which should be assigned to a specific API (predicted links to tags can be used to improve description of APIs). We run this experiment for the well-known *Google Maps API*.

Table 2 shows results using the tensor models with and without ageing for the last snapshot. The column *Without ageing* contains a list of tags representing targets of predicted top-10 links. This list is influenced only by structural patterns in the whole dataset, since the snapshot without ageing is used. The column *With Ageing* contains a list of tags, which is not only influenced by structural patterns, but also by time. Some of the predicted tags are the same in both sets, but on different positions. For example *travel* lost the first position, but *location* or *government* moved up to better positions.

Table 2: Top 10 tags for *Google Maps API* on the 1st April 2013.

Position	Without Ageing	With Ageing
1	**travel**	geolocation
2	realestate	**location**
3	sports	**travel**
4	reference	**government**
5	uk	geocoding
6	**location**	visualization
7	transit	transportation
8	food	gis
9	science	weather
10	**government**	deadpool

In order to explore differences in both sets we run the same experiment over time (i.e., by using the 22 snapshots). Figures 6(a),6(b) depict positions of tags in a top-10 set for each snapshot. The position is represented by a color on a scale from white to black where a darker color corresponds to a better position of a tag. Figure 6(a) depicts positions when the ageing is



(a) No Ageing (b) Ageing

Figure 7: Distance of predicted Mashups from the ending time of snapshot.

not used. It can be observed that a position of tags do not change very much over time once a tag gets to a certain position (e.g., realestate, travel). This is influenced by global structural patterns that the algorithm uses once they appear in the dataset. Note that each snapshot always contains data of a previous snapshot (see Snapshots paragraph in Section 4).

Figure 6(b) depicts positions when the ageing is used. There is a group of tags (food, reference, uk, sports, realestate) that were on better positions in the past (the darker colors in the bottom-left corner), however, they lost significance in recent time. On the other hand, a group of tags (e.g., geolocation, geocoding, location) had no significance in the past but is more preferred in recent time (darker colors in the top-right corner). This is caused by evolution of the structure of the dataset over time. Intuitively, this also proves the fact that mapping APIs and mashups (i.e, tags geocoding, location, geolocation) started to gain a popularity only 5 years ago and travel mashups and APIs are all-time popular. Please also refer to experiment in Section 3.6 for more details.

3.5 Influence of Time Information on Prediction

In this experiment, we present a relation of predicted links and time information of entities which participate in the predicted links. This experiment is motivated by a need to predict links between mashups and APIs. For example, to find top-10 mashups that could benefit from the Flicker API.

We run the experiment for all 22 snapshots. Figures 7(a) and 7(b) depict a distance in weeks of top 10 mashups from t_0 of every snapshot. We use a standard box plot to examine distributions of distances graphically. Figure 7(b) presents much lower distances than Figure 7(a). These results support our assumption that predicted entities in top-10 lead to links between entities with time information closer to t_0 (i.e., the present time of a particular snapshot) than the link prediction that does not use time information.

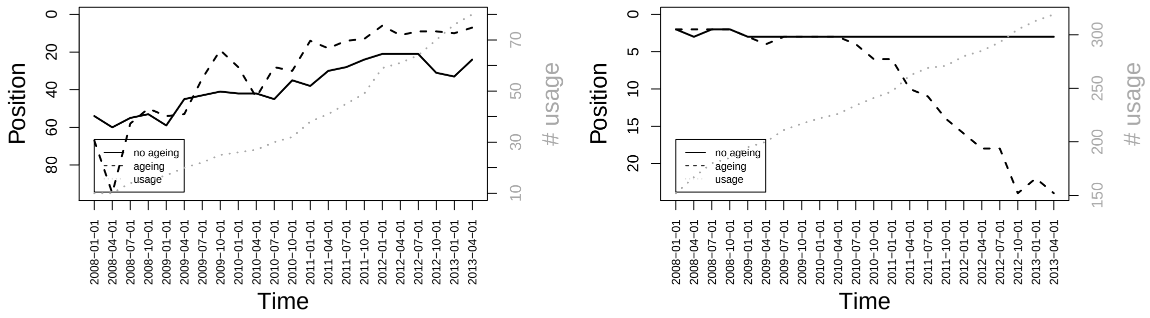
We also performed a quantitative experiment of this prediction task. We randomly selected 100 tags and predicted top-10 APIs that should be assigned to each tag. At the same time we randomly selected 1000 Mashups and predicted top-10 APIs which should be used in the specific Mashup. The mean value of distance is 33 weeks for the time-aware link prediction and 184 weeks for the link prediction that does no use time information.

3.6 Impact of Evolution of Structure

In this experiment, we demonstrate how the proposed method takes into account the evolution of the datasets' structure when predicting new links.

We run the prediction for two tags *realestate* and *geocoding* and evaluate their positions in top-L predicted links over time for the well-known *Google Maps API*. Figure 8(a) and 8(b) depict an evolution of the position for both tags on the left axis and a number of usages of the tags on the right axis (a usage of a tag means that an explicit link between an entity and the tag exists in the dataset).

Figures 8(b) shows a high position of the tag *realestate* when no ageing is used. This is influenced by the high number resources (APIs and mashups)



(a) Position of tag *geocoding*

(b) Position of tag *realestate*

Figure 8: Evolution of position over time for a specific tag.

tagged with this tag and the supporting structural patterns that exist throughout the history. However, when ageing is applied, the tag is gradually loosing its position since the structural patterns were created earlier in the past rather than in recent time (in a snapshot's time t_0). Figure 8(a) shows that the tag *geocoding* gets to slightly better positions when ageing is applied. This is caused by the fact that supportive structural patterns for this tag appeared in recent time. The next paragraph describes an example of elementary structural patterns that may influence positions of tags in link prediction.

Significant Sub-graphs. Our method is based on identification of hidden patterns in the structure of data (tensor factorization) in connection to the time information and ageing. Identified hidden patterns are used to predict new links in data. In order to find such significant patterns we can use an existing local property of graphs, called motifs. Motifs are defined as recurrent and statistically significant sub-graphs. We adopted the idea of motifs in this experiment as an ”evidence” of influence of structure and temporal information in tensor factorization with ageing. The goal of this experiment is to some extent provide an explanation of results from the previously described experiment in this section.

New links for *Google Maps API* can be predicted only when a similar pattern exists in the data and the pattern contains information related or similar to the *Google Maps API* structure. Based on the dataset structure, we define several elementary patterns which may influence the link prediction of the tags *realestate* and *geocoding* for the *Google Maps API*. By looking at the *Google Maps API* structure, we can see that it is a service, it has assigned a category mapping, a tag mapping, and supports JavaScript protocol. We breakdown this structure to the following queries (that we call patterns), where *X* can be either *realestate* or *geocoding*. We then measure the number of occurrences for each of the 8 patterns in the 22 snapshots.

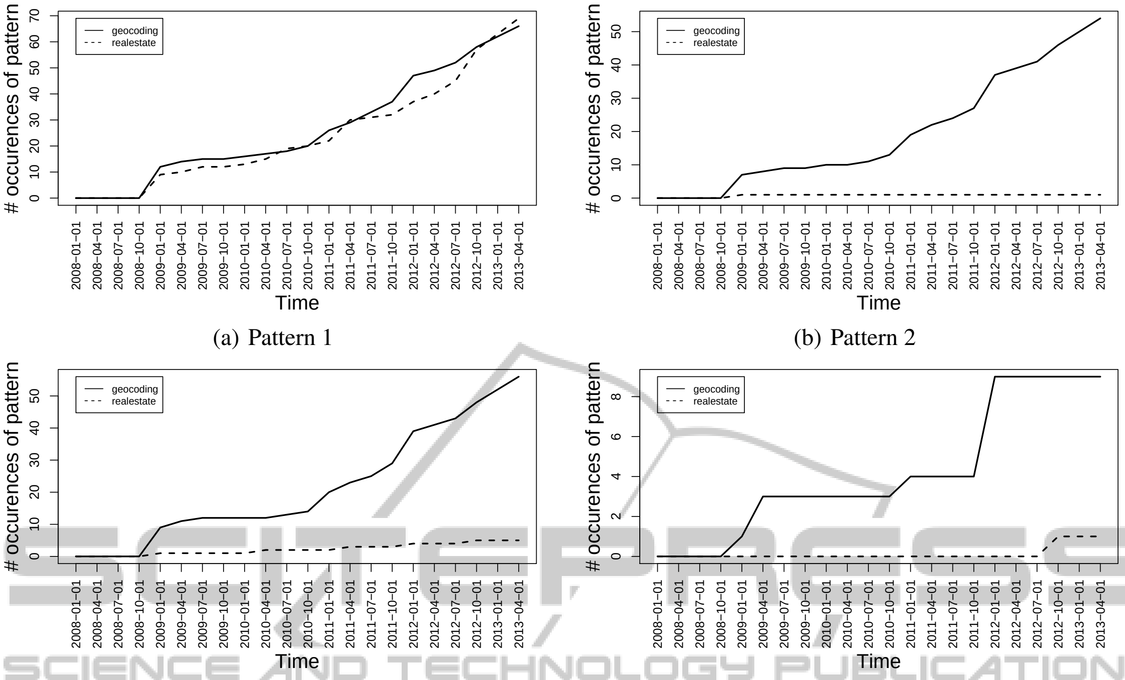
1. *?var rdf:type wl:Service AND*
   ?var ls:assignedTag ?X
2. *?var ls:assignedCategory ls:Mapping AND*
   ?var ls:assignedTag ?X
3. *?var ls:assignedTag ls:mapping AND*
   ?var ls:assignedTag ?X
4. *?var ls:supportedProtocol ls:JavaSript AND*
   ?var ls:assignedTag ?X

Figures 9(a)-9(d) depict a number of occurrences for each pattern over time (i.e., for each of the 22 snapshots). The tag *geocoding* has a higher number of occurrences of the patterns than the tag *realestate*. This means that there are more structures similar to the Google Maps API structure that have assigned tag *geocoding* rather than the tag *realestate*. Although this does not provide much evidence for the tag *realestate* and its high positions when no ageing is used (in Figure 8(b)) which is influenced by other structural patterns not shown here, it shows that a higher presence of the patterns in recent time promotes the tag *geocoding* to better positions when compared to positions when no ageing is used (Figure 8(a)).

4 DISCUSSION

Robustness. Although we evaluated our method on a domain-specific dataset from ProgrammableWeb, the method is capable to predict links in a dataset from any other domain. We have chosen the dataset from ProgrammableWeb as it contains sufficient information about creation time of entities that we can propagate to relevant links. We plan to evaluate our method on different datasets in our future work.

Temporal Information. Due to the nature of the data from ProgrammableWeb we deal with a specific form of time assigned to an entity as the created time



(a) Pattern 1

(b) Pattern 2

(c) Pattern 3

(d) Pattern 4

Figure 9: Number of occurrences for each pattern.

(see also Section 3 for information how we propagate this time to corresponding links). We understand the created time as a starting time from which the link exists in the dataset and we have no information about a duration of the link's existence. It is our future work to study various representations of time in linked datasets and incorporate them into the time-aware link prediction method.

Further, there are two basic types of expressing an existence of data - an explicitly defined *time point* using a document-centric and a fact-centric information (e.g., reification, N-ary relationships, snapshots of graphs, provenance, PROV-O, Memento etc.) (Rula et al., 2012) or deduced from other facts in an RDF dataset. The first category can be immediately used in our model. Since the availability of temporal information in Linked Data is still limited (Rula et al., 2012), especially for links, we derive the temporal restrictions from available data in dataset.

The types of links that never evolve or should not evolve (e.g. *dc:creator*, *rdf:type*) can be excluded from the temporal extension of tensor using value 1.

Ageing Function. Our goal was to show that time information is a very important aspect for link prediction and how a method to predict links can be extended with time information by modelling a retention of information using the ageing function. The formula we use for the ageing function is inspired by a representation of forgetting and retention mechanisms in the human mind.

Structural Patterns. Results of the time-aware link prediction highly depend on a structure of the RDF graph and a time when links were created. In Section 3.6 we identified simple structural patterns that may influence the link prediction in this specific dataset. However, there is no reason to assume that there cannot be present also other, more complex structural patterns that influence the link prediction. In our future work we plan to explore methods for automatic detection of more complex patterns.

Snapshot Creation. We have chosen the size of snapshots so that they have a sufficient amount of data for learning. Note that the data of some snapshots can be differently distributed with respect to time. Some snapshots might have data normally distributed but in some snapshots the majority of the data can be at the start or at end of the snapshot. Such distribution of the data has an impact on the link prediction.

5 RELATED WORK

There are two main topics closely related to our time-aware link prediction method, namely tensor factorization and relational learning. The models and methods covered by these topics are used to model multi-relational data and to perform the link prediction.

Most researches in relational learning are based on a statistical relational learning. These approaches are build upon the Bayesian or Markov networks (Friedman et al., 1999; Khosravi and Bina, 2010) or their combinations with tensor representations (Gao et al., 2012).

There is a growing interest in tensor models and factorizations in multi-relational data modelling. An overview of tensor factorizations and their applications is in (Kolda and Bader, 2009). There are two basic approaches, namely link-information-based approaches and node-information-based approaches. We adopted a model from link-information-based approaches by (Nickel et al., 2011), where each frontal slice of a tensor represents a relation. A similar model was also used in (London et al., 2013). These modelling approaches, however, do not work with time information. They only take into account entities and relations among them.

On the other hand, node-information-based approaches, take into account attributes of entities (Taskar et al., 2003; Raymond and Kashima, 2010). An extension of this work in (Nickel et al., 2012) is able to work with attributes (time attribute can also be included) and combine both approaches.

There are also existing approaches related to frameworks LIMES (Ngomo and Auer, 2011) and SILK (Bizer et al., 2009) that are focused on link discovery between different datasets. Our approach is focused on link prediction within one dataset.

There are existing researches, that use time for predicting links. In (Spiegel et al., 2012), the authors use the third-order tensor factorization, where two dimensions are used to represent relations and the third dimension represents time. This approach is however suitable only for one type of relation. A similar work was done in (Acar et al., 2009; Dunlavy et al., 2011; Ermis et al., 2012) where authors also work with a dataset with one type of a relation.

There are also other approaches that use either multi-modal representation of graph or temporal information for link prediction in Social Networks, e.g. prediction links in asynchronous communication (Oyama et al., 2011), prediction based on hypergraph (Li et al., 2013), prediction in multi-modal networks (Symeonidis and Perentis, 2014), however, they are less relevant to our work.

6 CONCLUSION AND FUTURE WORK

The popularity of publishing RDF datasets as Linked Data is significantly growing in recent time. A rich dataset contains a sufficient amount of links, however, not all links may explicitly exist in the dataset while some existing links may not be valid. Link prediction algorithms can be used to find links that do not explicitly exist in the dataset. Although there exists a number of sophisticated approaches for link prediction, there is still a lack of methods that can work with time information. The time information is important to work with datasets in dynamic environments and for the link prediction it helps to provide more relevant results. In this paper, we proposed the time-aware link prediction method that extends the tensor factorization to solve link prediction task with temporal information about existence of links. While existing methods use either multi-relational data or data with one type of relation and time information, our method utilizes both, the multi-relational data and time information in order to create tensor-based model. The results from the experiments on the real world dataset from ProgrammableWeb show, that the method effectively exploits both the structure of the datasets and the temporal information.

In our future work, we want to focus on the evaluation of the method on other datasets from Linked Data Cloud that incorporate links across data sources. We also plan to investigate other sources for temporal information. Further, we plan to explore different applications of the method. In particular, we want to evaluate the performance of the method in evaluation of existing links.

ACKNOWLEDGEMENTS

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/104/OHK3/1T/18. We also thank to ProgrammableWeb.com for supporting this research.

REFERENCES

Acar, E., Dunlavy, D. M., and Kolda, T. G. (2009). Link prediction on evolving data using matrix and tensor factorizations. In *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, ICDMW '09, pages 262–269, Washington, DC, USA. IEEE Computer Society.

Bizer, C., Volz, J., Kobilarov, G., and Gaedke, M. (2009). Silk - a link discovery framework for the web of data. In *18th International World Wide Web Conference*.

Dunlavy, D. M., Kolda, T. G., and Acar, E. (2011). Temporal link prediction using matrix and tensor factorizations. *ACM Trans. Knowl. Discov. Data*, 5(2):10:1–10:27.

Ebbinghaus, H. (1913). *Memory: A contribution to experimental psychology*. Number 3. Teachers college, Columbia university.

Ermis, B., Acar, E., and Cemgil, A. T. (2012). Link prediction via generalized coupled tensor factorisation. *CoRR*, abs/1208.6231.

Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. (1999). Learning probabilistic relational models. In *In IJCAI*, pages 1300–1309. Springer-Verlag.

Gao, S., Denoyer, L., and Gallinari, P. (2012). Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks. *CoRR*, abs/1204.2588.

Gutirrez-Basulto, V. and Klarman, S. (2012). Towards a unifying approach to representing and querying temporal data in description logics. In Krtzsch, M. and Straccia, U., editors, *Web Reasoning and Rule Systems*, volume 7497 of *Lecture Notes in Computer Science*, pages 90–105. Springer Berlin Heidelberg.

Khosravi, H. and Bina, B. (2010). A survey on statistical relational learning. In *Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence*, AI'10, pages 256–268, Berlin, Heidelberg. Springer-Verlag.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500.

Kopecky, J., Vitvar, T., Bournez, C., and Farrell, J. (2007). Sawsdl: Semantic annotations for wsdl and xml schema. *Internet Computing, IEEE*, 11(6):60 –67.

Li, D., Xu, Z., Li, S., and Sun, X. (2013). Link prediction in social networks based on hypergraph. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 41–42, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

London, B., Rekatsinas, T., Huang, B., and Getoor, L. (2013). Multi-relational learning using weighted tensor decomposition with modular loss. *CoRR*, abs/1303.1733.

Ngomo, A.-C. N. and Auer, S. (2011). Limes: A time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2312–2317. AAAI Press.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 809–816, New York, NY, USA. ACM.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2012). Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 271–280, New York, NY, USA. ACM.

Oyama, S., Hayashi, K., and Kashima, H. (2011). Cross-temporal link prediction. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 1188–1193, Washington, DC, USA. IEEE Computer Society.

Raymond, R. and Kashima, H. (2010). Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In Balczar, J., Bonchi, F., Gionis, A., and Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 131–147. Springer Berlin Heidelberg.

Rula, A., Palmonari, M., Harth, A., Stadtmller, S., and Maurino, A. (2012). On the diversity and availability of temporal information in linked open data. In Cudr-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J., Hendler, J., Schreiber, G., Bernstein, A., and Blomqvist, E., editors, *The Semantic Web ISWC 2012*, volume 7649 of *Lecture Notes in Computer Science*, pages 492–507. Springer Berlin Heidelberg.

Spiegel, S., Clausen, J., Albayrak, S., and Kunegis, J. (2012). Link prediction on evolving data using tensor factorization. In *Proceedings of the 15th international conference on New Frontiers in Applied Data Mining*, PAKDD'11, pages 100–110, Berlin, Heidelberg. Springer-Verlag.

Symeonidis, P. and Perentis, C. (2014). Link prediction in multi-modal social networks. In Calders, T., Esposito, F., Hllermeier, E., and Meo, R., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8726 of *Lecture Notes in Computer Science*, pages 147–162. Springer Berlin Heidelberg.

Taskar, B., fai Wong, M., Abbeel, P., and Koller, D. (2003). Link prediction in relational data. In *in Neural Information Processing Systems*.

Vitvar, T., Kopecký, J., Viskova, J., and Fensel, D. (2008). WSMO-Lite Annotations for Web Services. In *ESWC*, pages 674–689.