Czech Technical University in Prague
Faculty of Information Technology
Department of Software Engineering

Rich Semantic Representations in Web Usage Mining

by

Ing. Jaroslav Kuchař

A dissertation thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

Dissertation degree study programme: Informatics

Prague, August 2016

Supervisor:

doc. Ing. Tomáš Vitvar, Ph.D.
Department of Software Engineering
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Abstract and contributions

With a growing number of users browsing various web sites, the need of proper analysis and understanding of their behaviour becomes one of the most studied areas last years. Users interacting with a specific content provide huge amount of data during the behaviour. Such interactions are not self-explanatory till they are not properly represented and connected to the well described content items. Technologies of the Semantic Web become a part of many areas of informatics and they play a significant role in a representation of knowledge. With help of the Semantic Web we can build rich representations connecting users and content they are interacting with. Those rich representations associate interactions performed by users and available knowledge about the content and they allow to infer and utilize multiple relations.

This doctoral thesis studies a particular aspect of the recent research in using semantics for building and utilizing rich representations connecting users and the content. Our contributions address specific issues of using semantics in following areas: 1) Data acquisition - we deal with situations of modern user interfaces when a user performs multiple interactions per one content item and the required output is one relation representing user interest in the content 2) Semantization - we focus on linking of content to a knowledge base allowing a consecutive extraction of additional features and build its semantic representation 3) Enhancement - since most of existing knowledge is created by humans and fragments of links within a knowledge base can be missing, there is a need to manage the knowledge base using link predictions in order to get rich semantic representations 4) Utilization in Preference learning and Recommendation - rich semantic representations allow to utilize their relations and perform an intelligent selection of resources based on existing relations or they allow to build readable and concise user preference models that are applicable for recommendations of content items.

In particular, the main contributions of the dissertation thesis are as follows:

1. Method for an aggregation of semantically enriched user interactions.
2. Algorithm for linking content to a public knowledge base and a method for semantic aggregation.
3. Link prediction method that allows enhancement of semantic representations with respect to temporal information.
4. Method for selection of the most relevant target among a predefined set of candidates.
5. Preference learning and recommendation technique profiting from semantic annotations.

Keywords:
Semantic Web, Web Usage Mining, Link Prediction, Preference Learning, Association Rules, Recommendation.

Acknowledgements

First of all, I would like to express my gratitude to my dissertation thesis supervisor, doc. Ing. Tomáš Vitvar, Ph.D. He has been a constant source of encouragement and insight during my research and helped me with numerous problems and professional advancements.

I am also thankful for the input from doc. Ing. Ivan Jelínek, CSc and his supervision throughout the first years of my research.

Special thanks go to the staff of the Department of Software Engineering, who maintained a pleasant and flexible environment for my research. I would like to express special thanks to the department management for providing most of the funding for my research.

My research has also been partially supported by the Grant Agency of the Czech Technical University in Prague (grant No. SGS10/200/OHK3/2T/13, SGS12/093/OHK3/1T/18, SGS13/100/OHK3/1T/18 and SGS14/104/OHK3/1T/18), by the grant FRVS 395/2011, by the Czech Educational and Scientific Network - National Grid Infrastructure MetaCentrum (CESNET 540/2014) and by the European Community's Seventh Framework Programme (FP7-ICT) under grant agreement nº 287911 LinkedTV.

I would like to express thanks to my colleagues from Web Intelligence group and others, for their valuable comments and proofreading.

Dedication

This thesis is dedicated to my family.
For their endless love, support and encouragement.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

Miscellaneous Abbreviations

URI	Uniform Resource Identifier
HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language
OWA	Open-World Assumption
RDF	Resource Description Framework
RDFa	Resource Description Framework - in attributes
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
SAWSDL	Semantic Annotations for Web Service Description Language and XML Schema
RDFS	Resource Description Framework Schema
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
FOAF	Friend of a Friend
WM	Web Mining
WUM	Web Usage Mining
WSM	Web Structure Mining
WCM	Web Content Mining
WIE	Web Information Extraction
WI	Wrapper Induction
LOD	Linked Open Data
ARC	Association Rule Classification
CBA	Classification Based on Associations
NER	Named Entity Recognition
GA	Genetic Algorithm
VSM	Vector Space Model
CF	Collaborative Filtering
KNN	k-Nearest Neighbours

Algorithms and Contributions related Abbreviations

brCBA	business rules Classification Based on Associations
rCBA	Classification Based on Associations for R
CMAR	Classification based on Multiple Association Rules
ID3	Iterative Dichotomiser 3
MAE	Mean Absolute Error
MRR	Mean Reciprocal Rank
HR	Hit Ratio
TSP	Time Spent on Pages
CTR	Click-Through Rate
InBeat	Interest Beat
GAIN	General Analytics INterceptor
PL	Preference Learning
IF	Information Filtering
RS	Recommender System
ALS	Alternating Least Squares
FF	Ford-Fulkerson
PW	ProgrammableWeb
LHS	Left Hand Side
RHS	Right Hand Side
CAR	Class Association Rule
BoW	Bag of Words
BoE	Bog of Entities
TF-IDF	Term Frequency-Inverse Document Frequency
TD	Transactional Database

Chapter 1

Introduction

This dissertation thesis studies a particular aspect of the recent research in using semantics for building and utilizing rich representations connecting users and the content they are interacting with. Those rich representations associate interactions provided by users and available knowledge about the content. We focused on the benefits of semantics from data acquisition over semantization and enhancement to utilization of rich graph-based representations.

The rest of this chapter describes an overview of motivations, problems and contributions of this dissertation thesis. The brief summary of related work, previous results and structure of the dissertation thesis is presented as well.

1.1 Motivation

In recent years, the number of users on the web has grown significantly. Each user on the web can both consume and produce content items at the same time. This kind of user is called a *prosumer* [8]. Users interacting with a specific content, either while they are consuming or producing the content, provide huge amount of data about their behaviour. Those interactions themselves are not self-explanatory till they are not properly understood and connected to the well described content items.

Let consider a user that viewed a web page and developed an application. Without any more descriptive information about each item, we cannot elaborate on relations between content items with respect to interactions. Basically, we can only state the user is a visitor and a developer. Now consider we have features representing details about items: the web site is about news and application is about world maps. Simple conclusion can be that the first interaction relates to his private interests "visits of a news portal" and probably does not relate to the second one. The second one represents his professional interest in maps. However, in case we know that the visited web page describes new features of an API for a public geolocation service, both interactions probably relate to each other. Geolocation service is semantically close to the map application, since we usually use geolocation coordinates on a map.

The main motivation is to have a rich representation that allows to infer and utilize such relations. The assumption is that we need a well formatted representation of content items. Producers of the content can provide additional features for the content items at the time of publishing. Another possibility is to link the content item to any of publicly available knowledge bases. This link can be thus used to extract a set of desired features. The important aspect of all features provided by producers or even extracted from the knowledge bases is that they are originally created by humans. Fragments of information can be missing or be outdated, for that reason we also focused on enhancement of semantic representations while taking into account temporal information.

In this dissertation thesis we are focused on situations when users perform a set of interactions on content items while at the same time the content is semantically annotated (before or after the user performed interactions). The output is a concise representation that links users and content items described using publicly available knowledge bases. Furthermore, we enhance those representations to get extended information. The final enhanced representation is a good candidate for further processing and utilization in preference learning or recommendations.

1.2 Problem Statement

In this dissertation thesis we exploit semantics in rich representations that link information about user interactions, often called usage data, and descriptions of content items that users are interacting with. Fig. 1.1 demonstrates an overview of the overall methodology highlighting key areas of this dissertation thesis. The pipeline includes phases from data acquisition over semantization and enhancement to utilization of rich semantic representations. We define the main hypothesis as:

Main Hypothesis *Rich representations that link usage data and content descriptions profit from semantics, which enable to exploit extensive amount of linked information.*

We substantiate our main hypothesis by following research questions. Each question covers a different aspect of the main hypothesis with focus on a particular problem.

Research Question 1 (RQ1) - Acquisition *Can we incorporate semantics to processing and aggregating users' interactions and provide unified extended representation of relations between users and content?*

Since users can provide feedback by implicit or explicit interactions during their active work with the content, the feedback has to be properly processed, represented and transformed to a unified format. The main goal of the acquisition is collecting of users' interactions and transforming them to relations between users and content items. The special case is when the user provides multiple interactions per one content item. We focused on designing a methodology how to aggregate those interactions to one representative information.

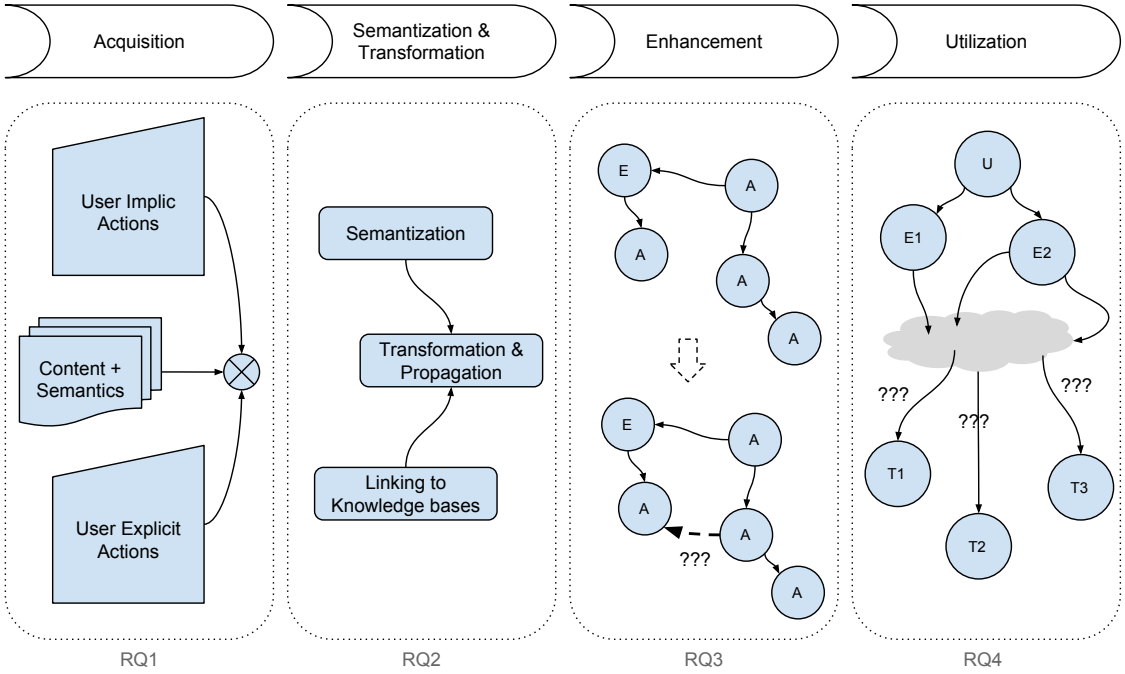


Figure 1.1: Overview of the overall methodology.

Research Question 2 (RQ2) - Semantization and Transformation *How can we extend already existing set of features describing the content by additional features using existing semantic sources and transform them to a semantic representation?*

The second important step in building the final representation is a set of subsequent steps covering semantization and constructing a semantic representation. Each content item is already semantically annotated, it has a link to a knowledge base or there is no semantic information. Semantically annotated content needs further transformations and post-processing. Especially in situations when multiple annotations are provided. Content with an URI identifier linking to a knowledge base has to be extended about additional semantic features using the identifier. For content without any semantic information, we have to design a methodology to link the item to a knowledge base. Finally, the data are provided in a semantic representation.

Research Question 3 (RQ3) - Enhancement *Is there a possibility to enhance a semantic representation about new links, while taking into account semantics of links and temporal information?*

The semantic data provided by publishers of content or even the data in knowledge-bases are originally generated by humans. There can be fragments of information missing or they can be outdated. In order to create rich semantic representations we focus on the methodology to predict missing links with respect to temporal information about links.

The enhancement transforms the input semantic representation to the rich semantic representation.

Research Question 4 (RQ4) - Utilization *How can we utilize the rich semantic representation connecting users and content for selection of content items or recommendation according to users' preferences?*

We focused on utilization of semantics and temporal information of rich semantic representations. They contain users linked together with semantic content. The goal of utilization is to get information about which item from a set of candidates selected from the whole representation is more relevant to a specific user. We also focused on approaches for representation of user preferences and semantic-aware recommendations.

1.3 Related Work/Previous Results

There are many researches dealing with problems and challenges of data acquisition and preprocessing. In our research we are focused on the semantics and semantic technologies in the Web Usage Mining (WUM) and related tasks. One of the first researches about exploiting semantics in the WUM is [9, 10]. The semantics generally provides more expressive form of the knowledge representation and can improve the quality of further recommendations [11]. At the time of the creation of WUM research topic, there were no rich user interfaces or interactive web applications allowing multiple interactions with one object item. According to our review of related works, there is a lack of existing approaches in the Web Usage Mining preprocessing focused on an aggregation of multiple interactions to one representative information.

From the semantization and building semantic representations point of view, we utilize well-known and general applicable approaches based on the Named Entity Recognition [12] to annotate and link content items to a knowledge base. Those approaches are widely used for annotation of various information sources. For domain specific data formats and content items, we focused on a mapping of well-known movie ratings datasets [13, 14] to a knowledge base. Existing approaches are based on "guessing" the *URIs* as links to the DBPedia [15] or computing similarity measures (e.g. Jaccard coefficient) to find the most relevant entity [16, 17]. For a post-processing of annotated content items we propose a semantic aggregation and propagation within ontology. The most relevant existing researches are about an activation of interests over a full ontology [18] or a vector space preference aggregation [19].

To enhance a semantic representation we use a link prediction algorithm, where most researches are based on a tensor factorization or a relational learning. Models and methods covered by these topics are used to model multi-relational data and to perform the link prediction. Generally, most of existing link prediction algorithms are focused on graphs and networks with single type of relation [20]. 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 [21]. We adopted a model from link-information-based approaches

by [6], where each frontal slice of a tensor represents a relation. It only takes into account entities and relations among them. Our model also incorporates temporal information. Existing researches [22, 23, 24, 25] are able to handle temporal information. However, they can work with a dataset with only one type of a relation.

We are focused on user-centric utilizations of rich semantic representations: a selection from candidates based on connections within the representation and a preference learning leading to recommendations. A particular method that relates to our selection method is a spreading activation. It is a graph-based technique, originally proposed as a model of the way how associative reasoning works in the human mind [26]. The spreading activation requires directed semantic network, e.g. an RDF graph [27, 28, 29]. Compared to our maximum activation method, the spreading activation does not guarantee an activation of a particular node while our method always assigns an activation if there exists a path between source and target nodes. Although there exist constrained spreading activation methods which utilise semantics of links [30], no version of the spreading activation takes into account the "age" of links as our method does. In our approach for preference learning we represent user profiles composed as a set of rules. Apart from practical issues, such as speed and the possibility to display the user profile in an intelligible way, rules are also considered as the most expressive form of encoding preferences [31]. The idea of using semantic web technologies, ontologies and Linked Data during the recommendation is widely adopted in recent research. The semantics used in either Collaborative and Content-based RS [32, 33, 34], or in the context of news recommendation [35]. Using rule-based approaches in recommendation tasks is also a research topic that was also covered by several researches and applications [36, 37, 38, 39, 40]. According to our review of related works, there is no exiting approach that is focused on using rules, semantics and is also applicable for client-side recommendations.

1.4 Contributions of the Thesis

The main contributions of this thesis are as follows:

1. **Method for an aggregation of semantically enriched user interactions (RQ1).** We design a method for an acquisition and an aggregation of user interactions, resulting in one concise relation between a user and a content item. The relation aggregates information from multiple interactions per one item to a single value representing a user interest. We focused on the general design and domain independence of the proposed technique. We have also implemented a proof of concept prototype that was evaluated in domains of web analytics, Smart TVs or recommender systems. The evaluation shows that it can be used in acquisition not only for research purposes but also in real world applications.

   Approaches were presented in following author papers [A.9, A.10, A.12, A.16, A.17, A.14, A.15] and contributions to projects' reports and deliverables [5].

2. **Algorithm for linking content to a public knowledge base and a method for semantic aggregation (RQ2).**
   We design a mapping of domain specific content items to a knowledge base. Predefined SPARQL queries identify the corresponding entity using features associated with content items and provide its unique URI identifier to the Linked Open Data (LOD) cloud. We explore the identifier using LOD principles in order to further augment features of the item. We evaluated the approach on a multilingual movie ratings dataset and we published mappings as a public dataset. The dataset was used for our evaluations in the next contribution of this thesis. It is also valuable for communities around Semantic Web and Recommender systems. Since each content item can have several semantic annotations, we also designed an approach to aggregate semantic annotations of one content item using an ontology propagation. It was evaluated within trials of the LinkedTV EU Project.

   The proposed semantization and propagation is presented in papers [A.3, A.10, A.11, A.12] and contributions to projects' reports and deliverables [5].

3. **Link prediction method that allows enhancement of semantic representations with respect to temporal information (RQ3).**
   We design a method to enhance the semantic representation. As the enhancement we consider a management of links: updates, removals or insertions of links. In our proposed approach we are focused on inserting: a prediction of links within one dataset. The key concept we use is a forgetting factor to decrease the influence of older links on the link prediction. The link prediction algorithm was evaluated on two domain specific datasets: a semantic version of a Web APIs directory and a movie ratings dataset. The algorithm is implemented in R and is publicly available for any further research.

   The link prediction approach is described in [A.4] and its extended version [A.1].

4. **Method for selection of the most relevant target among a predefined set of candidates (RQ4).**
   We designed an approach how to utilize the links of rich semantic representations connecting users and content items. We developed a novel method that allows a personalised selection of entities in the semantic representation. We use flow networks as an underlying concept for evaluation of the preference between a predefined set of candidates. The method also incorporates the temporal information as a key input allowing to decrease the influence of older links in processing flow networks. The method was evaluated on a semantic version of a Web APIs directory and the implementation of the proof-of-concept algorithm is also publicly available.

   The approach was published in the most cited paper [A.13].

5. **Preference learning and recommendation technique profiting from semantic annotations (RQ4).**
   We designed methods profiting from the semantics in the domain of user modelling,

personalizations and recommender systems. In our methods we focused on well understandable, explainable and justifiable user preferences and recommendations while taking into account the semantics. We use rule learning and rules as an underlying concept. Rules are considered as one of the most understandable representations for models. Humans can even add, edit or delete specific rules explicitly. The advantage of the approach we propose is that it is also applicable for client side solutions. The client side solutions preserve the privacy of users while rule based models supports understandability of models and recommendations. We evaluated proposed approaches for preference learning and recommendations in domains of Smart TVs and News recommendation challenges. Proof-of-concept implementations are publicly available as open source projects. The implementation of the rule based classifier is also available for the community of R language.

Preference learning and its application for news recommendations were presented in [A.8, A.7, A.6, A.5, A.2, A.18, A.19].

1.5 Structure of the Thesis

The thesis is organized into several chapters as follows:

1. *Introduction*: Describes the motivation behind our efforts together with our goals. There is also a list of contributions of this dissertation thesis.

2. *Background and State-of-the-Art*: Introduces the reader to the necessary theoretical background and surveys the current state-of-the-art. It summarizes the theory about the Semantic Web, Web Usage Mining, Web Information Extraction, Semantization and Enhancement, Preference Learning and Recommender Systems.

3. *Contributions*: Provides details about contributions of this dissertation thesis. The chapter is divided into several sections. Please note that each section is focused on a specific and clearly defined problem. They contain their own definitions, theory and related experiments. Dealing with the acquisition and aggregation is covered by Section 3.1. Section 3.2 introduces an approach for the mapping to a knowledge base used for the semantization of a specific movie dataset and a semantic propagation is also described. The enhancement of the constructed semantic representation with respect to the temporal information is described in Section 3.3. The utilization of rich semantic representations using a personalised selection is in Section 3.4. The last Section 3.5 introduces a semantic aware personalization and our efforts leading to recommendations.

4. *Conclusions*: Summarizes the results of our research, suggests possible topics for further research, and concludes the thesis.

Chapter 2

Background and State-of-the-Art

This chapter will describe all necessary terms and theory that will be operated within this work later on. Its intention is to explain all necessary topics to understand the motivation and the impact of this thesis.

Section 2.1 provides a definition of basic notions and a theory related to Semantic Web, Web Usage Mining, Web Information Extraction, Semantization and Enhancement, Preference Learning and Recommender Systems. It includes steps from the data acquisition to the utilization of rich semantic representations connecting users and content items. Section 2.2 summarizes recent results in this area and related work.

2.1 Theoretical Background

In this Section, we summarize a theory that is used throughout the rest of this thesis.

2.1.1 Semantic Web

This section is abstracted from the Semantic Web summary [41]. The main idea of the *Semantic Web* initiative is to publish information on the Web in a form that is processable and understandable not only by humans but also by machines. The Semantic Web is de facto an extension of the already existing Web. The vision is to build a platform for exchanging and sharing data, information and knowledge. The platform comprises three main branches [41]. The first field focuses on an extension of present knowledge representations and knowledge management systems so that they can be used in an open and distributed Web environment. The goal is to manage languages for a knowledge model (further also an ontology) representation and create systems for querying and reasoning. The second field is focused on an ontology interoperability (also known as an alignment), ontology management, annotation and information extraction using predefined ontologies. The last field is responsible for a development and management of domain specific ontologies.

The Semantic Web is mainly developed and maintained by academics with support of W3C standardization organization [1]. The Semantic Web becomes a part of many areas of informatics: knowledge engineering, software engineering, services and middleware, data and system interoperability solutions, logical languages, human-computer interaction, social networks, business applications, health, e-government, telecommunication or transportation.

2.1.1.1 Knowledge

The Semantic Web defines a set of languages for so called *semantic layer*. It allows to explicitly represent a knowledge as ontologies while the representation of data on the syntactical level is not affected.

Representations of the knowledge and the knowledge itself is crucial for the Semantic Web. The Semantic Web initiative has adopted all main kinds of knowledge:

- *Implicit* - latent knowledge usually represented using natural language formulations. It can be also inferred using already existing knowledge.
- *Explicit* - expressed formally but not originally part of the initial representation.
- *Declarative* - expresses the facts about real world objects. Anything that is known and proven.
- *Procedural* - reflects the way it was discovered or inferred. Usually represented as rules.

Although all categories of knowledge are supported, the Explicit knowledge is the most important one for the present Semantic Web. Since the Web is an open world (Open-World Assumption - OWA) and no presented fact can be considered as the final and complete one without any additional information, we need models allowing us to represent such situations. We need models and representations to describe real world objects on different levels of abstraction. They also have to allow to adjust and extend those models over time. Main responsibilities of models is object categorization and expression of relations among them. First-order (predicate) logic fulfils those requirements and it is thus used as a basis for models and their representations in the Semantic web.

2.1.1.2 Languages

The Semantic Web builds on top of fundamental principles of the WWW Infrastructure:

- Interlinking of resources using links.
- Using open, standard and freely available technologies.
- Separation of layers allowing independent innovations.

[1] `https://www.w3.org/`

The following three concepts realize those principles and form a basis of the Semantic Web: URIs (Uniform Resource Identification) for an identification of resources, protocol HTTP for an interaction with resources and XML format for a representation of resources. The Semantic Web languages use URIs as identifiers of resources (concepts, relations, objects) that act in knowledge representations using ontologies. XML format is used for exchanging of resource representations on the Web. Semantic Web languages utilize XML as a serialization format for a syntactic representation of ontologies. HTTP is widely used as a transport protocol allowing access to resources.

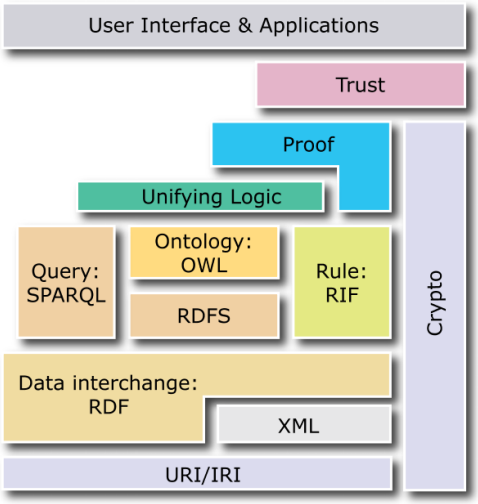


Figure 2.1: Semantic Web Stack - the hierarchy of languages and technologies.

Figure 2.1 [2] illustrates languages of the Semantic Web that rely on fundamental architecture of Web: URIs and HTTP protocol. Layers in the stack are important to meet various requirements on the level of semantic expression. XML language or other serialization formats (e.g. N3 or Turtle) stand for the lowest layer responsible for a syntactic representation of the knowledge. The second layer (RDF) allows to define any relation between objects and corresponding categories. RDF does not allow to explicitly define the meaning of relations or objects. However, the third layer represented by RDFS extends the abilities about a so called lightweight semantics - a definition of Classes and taxonomical relations (subClassOf). The complex semantics is introduced using a description logic (OWL) and a procedural knowledge (rules) in the highest layers.

The advantages of layers and the corresponding scalability are mainly used in design of applications that can reuse the features of lower layers. Each application can thus use only specific level of knowledge according to its requirements. Each layer of the knowledge model associated with the language is placed into a namespace. The namespaces allows to separate the layers and independently maintain the model of knowledge.

[2] Available from: `http://www.w3.org/2007/03/layerCake.png`.

XML eXtensible Markup Language is a markup language. The main purpose is to describe hierarchical structures of textual documents using tags. The pair of tags (start and end tag) together with the content (another pairs of tags or textual content) form an entity. Even if the tag can be identified as a certain level of meta information, there is no formal definition of the semantics. XML Schema is used to define the structure (syntax) of the XML document. Although there are possibilities to define rules within XML Schema, they have no connection to formal logic mechanisms. XML Schema itself is not a language to describe knowledge. However it defines primitive data types (integer, string, ...) that are used in languages of the Semantic web.

RDF Resource Description Framework forms a basis for the representation of knowledge. It is considered as a model for data interchange on the Web [42]. RDF uses so called triples to construct a graph structure: <*subject, predicate, object*>. Subjects and objects are two resources that stand for nodes and predicates stand for relations/links in this directed graph structure. This graph view is often used as an easy-to-understand visualization of RDF representations. [3] RDF does not define the semantics of the subject, predicate or object. However, it allows to define a categorization (using *rdf:type*). It also defines containers (*rdf:bag, rdf:seq, ...*). Moreover, RDF introduces a serialization mechanism that enables to convert models constructed in RDF into formats suitable for the processing in lower layers (e.g. transportation over HTTP). The main serialization format is RDF/XML defining rules to convert RDF to XML. The advantage of XML representations is the possibility to use any standard tools for processing and managing XML. Other known serialization formats are Notation 3 (N3) or Turtle.

RDFS Resource Description Framework Schema is a language allowing to express a lightweight semantics. It is an extension of the RDF. It provides constructs to describe classes of objects, types of relations and their hierarchy. The class is defined using *rdfs:Class* and the association to class is expressed using *rdf:type*. In RDFS we can specify constraints on properties using *rdfs:domain* and *rdfs:range*. They both define the allowed types acting as subjects and objects in triples. The hierarchical relation of classes and properties can be defined using constructs: *rdfs:subClassOf* and *rdfs:subPropertyOf*.

OWL Web Ontology Language allows to define the knowledge on the level of description logic. Since each layer increases the complexity of tools for knowledge management and computational complexity of reasoning algorithms, the OWL is divided into: OWL-Lite, OWL-DL and OWL-Full. OWL-Lite mainly defines a taxonomic hierarchy; class and property equivalence; transitive, symmetric and inverse properties; property restriction, binary cardinality and class intersection. OWL-DL extends the OWL-Lite about class disjunction, union and complement; cardinality. The overall possibilities of expressions are limited by decidability of constructed models. OWL-DL preserves decidability. OWL-Full

[3] `http://www.w3.org/RDF/`

offers full set of expressions of OWL. It does not preserve the decidability during reasoning. There are no effective algorithm that are able to use all features of OWL-Full.

2.1.1.3 Ontologies

Semantic Web languages are used to describe knowledge models called ontologies. An ontology is not only the knowledge model described by a language, it is also a method to ensure the interoperability on the Web. It is focused on a study of entities, concepts and their relationships. Ontologies are usually referred as complex and formal collections of terms and their relations. The structure can be represented in different ways: for example a set, hierarchy or taxonomy. Ontologies use formal languages (for example *Web Ontology Language - OWL*). [4] Vocabularies are considered as a special and light-weight form of ontologies. For example a collection of URIs structured in an hierarchical way. [5] The term came originally from philosophy: "An explicit and formal specification of a conceptualization" [43]. The definition is explained as follows:

- *Formal* and *explicit* are associated with the meaning of expressing the knowledge that is formulated using a particular formal and logic language.
- *Shared* means that the ontology is adopted and used by wider communities. They accept to use the ontology to describe knowledge of the specific domain. It is also widely understood as a social contract.
- *Conceptualization* is related to the definition of concepts that capture the structure of the domain with possible restrictions.

Ontology is a description of a domain specific knowledge that is collaboratively designed, developed and maintained. It is a common model allowing communication within communities and systems using such ontology. Ontology also acts as an effective support for interoperability.

Types of Ontologies The semantic web community defines two main types of ontologies:

- *Upper Ontologies* - define general and widespread concepts usually available in all domains. Those ontologies can be shared in all domains. Using upper ontologies we ensure the upper-levels interoperability between various ontologies. The examples of upper ontologies are: Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) and WordNet.
- *Domain Ontologies* - they define concepts of a particular domain. The domain specific ontologies usually use concepts defined in upper ontologies and extend them about particular concepts. Examples of domain specific ontologies are in datasets

[4] `http://www.w3.org/standards/semanticweb/ontology`
[5] `http://semanticweb.org/wiki/Ontology`

forming Linked Open Data [6]. They are collaboratively designed and managed. Most important examples: DBpedia[7] (RDF data from Wikipedia), FOAF[8] (ontology to describe personal profiles), SIOC [9] (ontology for social communities and networks) or DBLP [10] (ontology for bibliography).

Linked Open Data The goal of the Linked Open Data community is to build on top of the existing web and extend it about datasets that are interlinked. Linked Open Data is defined as: "A way to link different data sources and therefore connect these sources into a single global data space. It provides a publishing paradigm in which not only documents, but also data, can be a first class citizen of the Web, thereby enabling the extension of the Web with a global data space based on open standards - the Web of Data." [44].

The term Linked Data refers to a set of best practices for publishing and interlinking structured data on the Web using URIs and RDF.

Four basic principles of Linked Data[11]:

- Use URIs as names for things.
- Use HTTP on URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
- Include links to other URIs, so that they can discover more things.

DBpedia is constructed using extractions of knowledge from the well known Wikipedia and is publicly available as a multilingual knowledge base [12]. DBpedia maps all information available in Wikipedia to a single shared ontology. The release DBpedia 2014 consists of 3 billion pieces of information (RDF triples), 320 classes and 1,650 properties. All data are available for download as exports in appropriate serialization formats. The available knowledge can be also accessed using searching and querying mechanisms via the provided SPARQL endpoint. Although DBpedia consists of links to several external data sources, other existing data sources and knowledge bases publish links pointing to DBpedia as well. Therefore, DBpedia is considered as one of the central interlinking hubs in the Linked Open Data (LOD) cloud [45]. The advantage of DBpedia is that it covers many domains, it represents real community agreement, it automatically evolves as Wikipedia changes, and it is truly multilingual [46].

[6] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[7] http://wiki.dbpedia.org/about
[8] http://www.foaf-project.org/
[9] http://sioc-project.org/
[10] http://www4.wiwiss.fu-berlin.de/dblp/
[11] https://www.w3.org/DesignIssues/LinkedData.html
[12] http://wiki.dbpedia.org/about

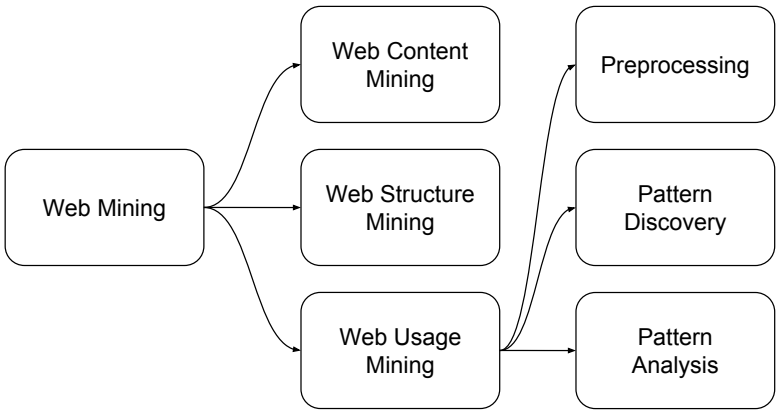


Figure 2.2: Structure of Web Mining [1].

2.1.2 Web Usage Mining

The basis for an analysis of users' behaviour on the web was laid together with a growing number of users browsing various web sites. Such situation has allowed to emerge a new field of interest - Web Mining (WM) [47] that is focused on an analysis of data related to the Web. Generally speaking, WM is focused on an extraction of interesting or potentially useful patterns, regularities or latent information from web artefacts or activities on the Web[47]. Figure 2.2 demonstrates a basic structure of the Web Mining. One aspect of WM is to analyse relationships between users and content items of visited websites - Web Usage Mining (WUM) . One of first attempts allowing to get insights about behaviour of users was performed by an analysis of server side web logs. The drawbacks of this old-fashioned analysis are that the processing incorporates a set of complex procedures to get meaningful information and they contain limited details about each user or content items. Together with WUM, two fields of research become popular: Web Content Mining (WCM) and Web Structure Mining (WSM). WCM is able to analyse a content - get details about the content and its semantics. WSM targets on an analysis of the structure - to get details about relations of the content, how is the content organized to categories and subcategories. These three fields allowed to establish more sophisticated approaches that allow the complex analysis of behaviour and modelling of user preferences.

The input data for the WM can be collected from various sources. There are four main groups acting as data sources for WM [48]: 1) *Web Content* - a representation of web resources usually used for the presentation to users in a browser. They contain textual and multimedia content. Multimedia content becomes recently more popular than textual information that was originally the main representation of any information. The content also takes into account metadata coming from resource representations or HTTP headers. 2) *Web Structure* - organization of information using features of hypertext: links connecting various resources as a graph and intra-page links forming a structure of information within one resource. 3) *Usage data* - data describing information about user interactions and

usage patterns of web resources. They are collected based on the real browsing behaviours of users. 4) *User data* - main source of explicit information about users (e.g. demographic information) coming usually from registration forms.

As was previously mentioned, basic approaches of behavioural pattern recognition and user modelling are based on the Web Usage Mining. Such approaches monitor the user behaviour and they typically deal with an analysis of sequences of user actions within websites. The WUM can be divided into two main categories [49]: *General Access Pattern Tracking* and *Customized Usage Tracking*. General Access Pattern Tracking is focused on an analysis of user patterns and general trends to get overview about the overall behaviour. Customized Usage Tracking analyses individual trends where the goal is to customize pages to individual users. Main possible applications, but not limited to, of WUM are [50]: 1) Personalization, 2) System Improvement, 3) Site Modification, 4) Business Intelligence, 5) Usage Characterization.

Further in our research, we will focus on the WUM from the point of view related to applications in the personalization. WUM can be divided into following main steps: data collection, preprocessing, pattern discovery, pattern analysis and finally an application. The typical input of WUM is a clickstream - a sequence of clicks/visited pages that the user performed during the visit of a web site. This clickstream is originally transformed from raw Web log files stored on the server. At this phase, the information can be integrated with data from other sources, such as data about web structure, relations between pages, a semantic description of pages and others. Preprocessed data are analysed using various algorithms (application of several data mining approaches). The outputs of the pattern discovery and analysis process are user profiles that represent patterns of behaviour, interests and intents of users. They can be used for prediction of future interests in a recommendation phase [51].

2.1.2.1 Data Collection and Preprocessing

First step of the WUM analysis is a data collection and appropriate preprocessing. It is an important part of the whole process, because the data quality is a significant prerequisite for all following steps. The data are collected, cleaned, filtered, merged from multiple sources and also transformed to a unified format.

There are two fundamental categories for data collection: implicit and explicit data collecting. They can be used independently or as a combination. Explicit data collection is typically based on filling forms by users. Those forms contain personal or demographic information, information about interests and others. The drawback is that each user has to fill in information. It consumes time of each user, users are not willing to fill any forms and many users don't fill accurate information. In contrast with the explicit one, an implicit data collection does not require an intervention of any user. The table 2.1 summarizes approaches of implicit feedback collection [2]. It also presents main pros and cons of all approaches.

There are four main steps of the subsequent preprocessing as follows [52]: 1) *Data Cleaning* - removing of irrelevant items, log entries produced by spiders and crawlers or

Table 2.1: An overview of implicit collection techniques [2].

Collection technique	**Information collected**	**Information breadth**	**Pros**	**Cons**
Browser Cache	Browsing history	Any Web site	User need not install anything	User must upload cache periodically
Proxy Servers	Browsing activity	Any Web site	User can use regular browser	User must use proxy server
Browser Agents	Browsing activity	Any personalized application	Agent can collect all Web activity	Install software and use new application while browsing
Desktop Agents	All user activity	Any personalized application	All user files and activity available	Requires user to install software
Web Logs	Browsing activity	Logged Web site	Information about multiple users collected	May be very little information since only from one site
Search Logs	Search	Search engine site	Collection and use of information all at same site	Cookies must be turned on and/or login to site, may be very little information

error log entries. 2) *User identification* - assigning unique user identifier to all entries coming from one user (combination of basics approaches related to the same IP address or advanced fingerprint based approaches can be used). 3) *Session Identification* - performs grouping of log entries to sequences related to one user visit (using time-based or navigation based heuristics). 4) *Path Completion* - automatic detection of missing entry in the log that can be caused by a caching, proxy servers or any corrupted communication.

2.1.2.2 Pattern Discovery

This section summarizes basic approaches of the pattern discovery phase for modelling of users [53, 54]. Although there are also attempts to perform those approaches in real-time (on-line), they are typically processed in batch (off-line) [55].

Clustering divides data or users into groups, where similarities in clusters are maximized and similarities between clusters are minimized. Users in clusters have similar behaviour, interests and intents. Three main categories exist:

- Partitioning methods, that creates k partitions (k-means algorithm).
- Hierarchical methods, using top-down or bottom-up approach.

- Model-based methods, where the best fit between members of cluster are usually specified using probabilities.

Association Rules typically uses well known Apriori [56] algorithm (originally proposed for market basket analysis), where groups of items occurring together are found. These are called frequent sets. Association rules are generated from those sets. For example, the following rule [57]: $\{/special-offers/, /products/software/\} \Rightarrow \{/shopping-cart/\}$ identifies that the special offer positively affects sales. Users that visited pages about special offers and software products together with shopping cart indicate purchase of the product.

Sequential and navigational patterns are similar to association rules but they find sequences of items that are time-ordered. Let consider the sequence of items: $\{i_1, i_2, i_3\}$. When we perform a discovering of sequential patterns, two types are identified: Closed and Open Sequences. The sequence pattern $i_1, i_2 \Rightarrow i_3$ is satisfied as the Closed type by $\{i_1, i_2, i_3\}$ but not by $\{i_1, i_2, i_4, i_3\}$, because i_4 appeared between items i_2 and i_3. But it is satisfied as the Open by both of them. Markov models can be used as the underlying concept for the sequential modelling.

Classification The goal of the classification is a labelling of users or mapping of users to classes based on models build on top of the previous browsing history. A decision tree classifier, naive Bayesian classifiers, support vector machine or any other classifier can be used for the classification [58].

2.1.2.3 User Profiling and Profile Representations

The representation of user profiles is a specific output of the WUM especially for personalized applications [2]. Profiles are typically built from topics of interest to the user. Profiles can be considered static and dynamic. Static profiles maintain the same information over time and dynamic profiles can be modified. If the profile takes into account time, the short-term and long-term profiles exist. Short-term profile represents current interests and long-term represents unchanging interests. Those profiles can be constructed and updated manually or automatically. Automatic techniques are more popular. Some approaches use genetic algorithms, neural networks, probabilistic techniques or vector space models.

The profiles can be represented as sets of weighted keywords, semantic networks, weighted concepts or association rules [2]. The simplest to build are keyword profiles, but they need large amount of user intervention in order to learn terminology. The profile is generally represented as a set of keywords and weights, which denote a level or importance of interest associated to the keyword. The keywords are extracted automatically from text or explicitly defined by the user and can be grouped. The example can be: $Music\{Rock = 0.7, Pop = 0.2, Metal = 0.1, ...\}, Sport\{Football = 0.2, Hockey = 0,5, ...\}$

The semantic network profile is a weighted network, where each node represents a concept (a keyword) and links and weights represents relations and associated levels of

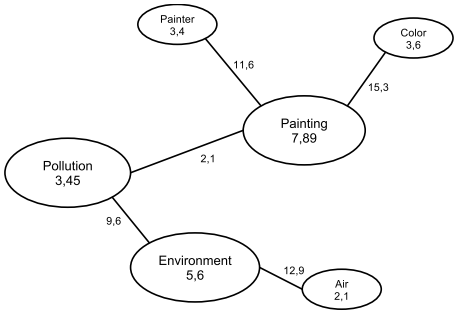


Figure 2.3: Example presenting a semantic network [2].

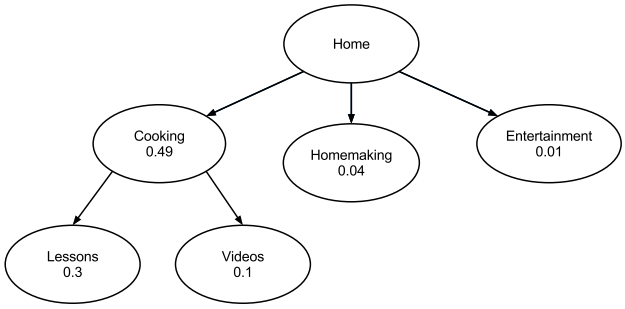


Figure 2.4: Example presenting a concept profile [2].

interests. The example is on the Figure 2.3. Concept profiles are similar to the semantic network profile. There are conceptual nodes and relations between nodes. The difference is that nodes represent abstract topics. The hierarchical structure can be also used. The Figure 2.4 depicts an example of a hierarchical concept profile.

Main idea of profile representations is in maintaining interests and levels associated to each individual interest. In Cambridge Advanced Learner's Dictionary, interest is defined as "activities that you enjoy doing and the subjects that you like to spend time learning about" [59]. What is more, the interest does not relate only to an assigned level, it also relates to a temporal dimension (interest can appear, disappear, change, ...). Consistent description and representation method of user interests are required for personalized Web applications. In the paper [59] was presented formal definition of "e-foaf:interest" vocabulary for describing user interests. The vocabulary is based on RDF/OWL[13], Linked data[14] and FOAF[15] vocabulary. The interest can be interoperable across various applications. Formal definition of interest form the e-foaf:interest vocabulary is:

$$< Interest\_URI, Agent\_URI, Property(i), Value(i), Time(i) >$$

where $InterestURI$ denotes the URI address that is used to represent the interest, $AgentURI$ denotes the agent/user that has the specified interest. $Property(i)$ is used to describe the name of the $i-th$ property of the specified interest. $Value(i)$ denotes the value of $Property(i)$. $Time(i)$ is the time that $Value(i)$ is acquired for the $Property(i)$. This formal representation takes into account the time of interests, when it appears or disappears. It is important indicator of current interests.

2.1.2.4 Dynamical modelling

The behaviour and interests of each individual user can change over time very quickly. The traditional approach is based on a so called off-line modelling where the models of user

[13] http://www.w3.org/standards/semanticweb/

[14] http://linkeddata.org/

[15] http://www.foaf-project.org/

profiles and representations are periodically created and updated. There might be several issues for specific domains dealing with large amount of data and the need for repetitive processing of all data. Another approach of processing data is in a dynamical way, often called on-line. The specific versions of algorithms for incremental learning are used to update models over time. The main advantage of the dynamic approach is an incremental modelling. The whole data are not periodically processed, recently appeared information update existing models. The final model is available in real time without any delay caused by the repetitive processing of off-line versions.

The introduction to the area of "real-time web usage mining" was presented in papers [60, 61]. The overview provides an introduction to on-line web usage mining and presents an overview of the latest developments. The main idea is based on the incremental versions of algorithms that are used in Web Usage Mining. This paper identifies major challenges in the field [62]:

- *Change detection*: issues associated with changes and evolution of content items, behaviour or interests.
- *Compact models*: since large amount of profiles exist on each web site, the compact representation of each model is required.
- *Maintenance of page mapping*: how to maintain consistency between pages and web usage data, because the content can change very rapidly over time.
- *New types of web sites*: how to collect usage data from the new rich interfaces and applications (e.g. based on AJAX, Flash etc.).
- *Public data sets*: the lack of publicly available web usage data sets influences the research and evaluation of new methods.

2.1.3 Web Information Extraction and Semantization

Since huge amount of potentially useful information is still formatted as a free text that is presented to human users (except recent metadata annotation efforts, open data activities and growing availability of several Web APIs), it is difficult to extract the relevant data. Web Information Extraction (WIE) approaches together with Semantic annotation (further Semantization) transform web resources to structured, machine processable and understandable representations. The Web Semantization is considered as a process that provides semantic annotations for web content as an enrichment [4]. It uses concepts from an ontology for the annotation in order to explicitly represent a knowledge for a machine [63].

2.1.3.1 Web Information Extraction

The heterogeneity of web resources leads to design of several sophisticated approaches that can extract information from unstructured and continuously changing resources. Formal

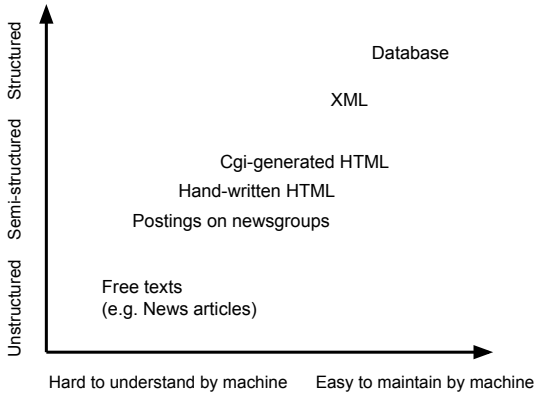


Figure 2.5: Task difficulties for wrappers based on the structure of documents [3].

definition of a WIE task is represented by a specification of its inputs and required extraction targets. As an input we can consider the unstructured document usually formatted as a free text or semi-structured documents in table or list formats. The output is an extraction target in the format of a relation of tuples or complex objects, hierarchically organized [3].

Generally, elemental extraction approaches are called recognizers, where the main responsibility of such extractions is a procedure to find a piece of information based on its appearance (e-mail addresses, phone numbers or street addresses). Most of nowadays crawlers and search engines are able to automatically collect such information [64]. Main WIE solutions are based on so called wrappers. *A Wrapper* is defined as software solution that encloses ("wraps") an information source (e.g. a web server, a database etc.). They provide the information for other systems in a predefined structural format, such system can thus access the information without changing internal mechanisms of the original information source [3].

The wrapper can be constructed by hand preparing dedicated extraction rules. Nevertheless, it is time-consuming end error-prone operation especially in the open web environment where the resources can frequently change and evolve over time. Wrapper Induction (WI) [65] is a method to automatically construct wrappers based on annotated samples of resources. It learns a set of extraction rules using those samples and performs performs a pattern matching task. The WIE are usually categorized based on task difficulties, used techniques and levels of an automation.

Task Difficulties The task difficulty is influenced by a specification of inputs and extraction targets. WIE consumes various input formats where each provides various levels of structured information. It affects the complexity of processing and understanding by machines. Figure 2.5 summarizes levels of the structure, corresponding levels of machine understandability and examples of inputs. Extraction targets also influence difficulties in the way they aggregate inputs to output entries - record extracted as an item appeared on a single page, record including all information per a single page and record covering

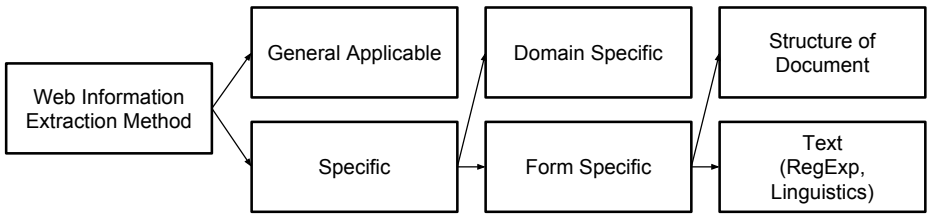


Figure 2.6: Web Information Extraction methods also used in Semantization [4].

a whole site. The record representation is important too: plain text, light hierarchical structures or complex objects [3].

Used Techniques Fundamental steps of each technique is a tokenization, extraction rules application and construction of a final record. The tokenization is influenced by the level of used granularity - on the level of words, HTML tags etc. Several approaches can be used for extraction rules induction: top-down or bottom generalization, pattern mining or logic programming. Extraction rules can be represented using regular grammars, logic programming or tree path traversing HTML DOM. Extraction ontologies can be used too [66]. Application of extraction rules to construct records is also dependant on the count of iterations required to complete each record [3].

Level of Automation The main difference is in a need of participations of humans during the WIE process. Several approaches requires labelled input data, where each input is annotated and used for learning of a wrapper. There are also approaches that use original data but need the assistance during learning of extraction rules. The difficulty is also affected by the automation of data collection, application of wrappers and robustness of approaches in terms of usage in different tasks or domains [3].

2.1.3.2 Web Semantization

Resent research has been focused on semantization (semantic annotation) of resources. The closing of the so called "semantic gap" is a key to significant improvements in information retrieval, browsing of resources and enabling creation of new applications and services [67]. It is generally understood as an automated process to provide annotations. Semantization approaches are tightly connected to general WIE methods.

Techniques (Figure 2.6) can be generally divided into two main approaches [4]: *General Applicable* and *Specific*. The representative of the general applicable approach is a so called Entity Resolution [68]. The task is usually defined as : "given a general ontology, find all the instances from the ontology that are present in the processed resource" [4]. It can be achieved performing two subtasks: Named Entity Recognition and Disambiguation of ontology instances that can be interlinked with named entities detected in the source [4]. In the following text we understand the Named Entity Recognition as a task of text analysis to

recognize information units like names, including person, organization and location names, and numeric expressions including time, date, money, percent expressions and others [69]. As the important aspect of semantic named entity recognition we consider to possibility to provide the reference for the detected entity as URI linking to a LOD knowledge base such as the DBpedia.

Domain specific approaches are widely used. Those approaches are more sophisticated but only applicable for the certain domain. Form specific approaches are focused on particular input formats. As particular input formats we can consider tables, structured documents in XML or HTML. Pure text based inputs are processed using regular expressions, patterns or detailed linguistics analysis.

2.1.3.3 Web Content Annotation

Semantic Web defines so called semantic layer that extends languages to describe Web resources. The goal is to annotate existing resources so that they can be used in intelligent tools allowing searching, reasoning and interoperability. Connecting the semantic and the non-semantic layer using so called annotations ensure the possibility to design intelligent tools that can effectively use the semantics. Annotation allows linking the semantic data with data in XML, HTML and other formats. It allows to use the same representation in applications that are able to work with non-semantic layer and semantic layer as well. There are two main existing standards specifying annotations for XML/HTML.

Embedding This annotation can be used for XML/HTML and is defined by specifications Resource Description Framework - in attributes (RDFa)[16] and Gleaning Resource Descriptions from Dialects of Languages (GRDDL) [17]. The output is a document consisting of both semantic and non-semantic data. RDFa defines the ways for embedding RDF into the HTML using special attributes (e.g. property, content, datatype, typeof, ...). GRDDL defines a transformation (e.g. XSLT, XQuery) enabling the extraction of semantic data from a document. Practically, annotation are used in HTML documents where HTML is used for presentation and RDF in special crawling and indexing services.

Model references This annotation can be used for XMLSchema. It is defined in a specification Semantic Annotations for Web Service Description Language and XML Schema (SAWSDL)[70].The specification allows to define references between XML elements of XML Schema and concepts in ontologies. For transformation can be also used XSLT or XQuery. Those annotation can be practically used for the annotation of web services.

2.1.4 Semantic Knowledge Transformation and Enrichment

Contemporary semantic knowledge-bases have been created either manually or with help of automatic extraction tools. The manual approach usually leads to a detailed and complex

[16] `http://www.w3.org/TR/rdfa-syntax/`
[17] `http://www.w3.org/TR/grddl/`

knowledge representing multiple facts. Since it is a time-consuming operation to construct extensive and complex structures, it is limited in terms of amount of information or they are only domain specific. Automatic approaches can provide comprehensive amount of extracted information but usually with limited complexity. Both approaches are also error-prone and have thus their limitations to create consolidated and integrated knowledge-bases [71]. The need of the semantic knowledge bases management has introduced several approaches that operate with graph-based or semantic representations (e.g. RDF). They allow to update and enhance single knowledge-base or even interconnect them. The goal is to provide a clean, rich and interlinked knowledge-base [72].

2.1.4.1 Ontology Mapping

The specific knowledge-base can be enriched using approaches on the level of ontologies - called ontology mapping or alignment. They are usually used to ensure the data interoperability in the semantic web. The goal is to "determine which concept and relation symbols of one ontology are mapped to concept and relation symbols of the other" [73].

2.1.4.2 Link Prediction and Discovery

The significant portion of knowledge is represented by relations between entities within a knowledge-base. Since RDF is a graph, each entity is represented as a node and the relation as a link connecting nodes. In some cases, not all links are present. They can be explicitly or automatically hidden by users to protect their privacy (especially in knowledge about users, their profiles, social relationships etc.)[74]. The links can be missing due to several issues that can happen during a construction of a knowledge-base - crawling resources, extraction and annotation approaches etc. Presence of links can also evolve over time. The main goal of a link prediction method is to determine the likelihood of a possible (currently not existing) association between nodes based on the knowledge about the current state of the graph. There are several existing ways to predict links in a graph. The main differences are in a complexity, prediction performance, scalability, and generalization [75]. They are based on several techniques from feature-based classification and kernel-based method to matrix factorization and probabilistic graphical models. Link Prediction approaches can be divided into [75]:

Feature based Link Prediction is a supervised classification task where we can consider each pair of nodes as a data point in the train dataset. The existence of a link between nodes is represented as a number 1, 0 for non-existing links. Any popular algorithm can be used for such kind of binary classification (e.g. naive bayes, neural networks, support vector machines or k-nearest neighbour, ...). The most critical aspect is a selection of used features. Feature based approaches usually use graph topological features based on node neighbourhood or node paths. Since there is no need for any domain knowledge, they are generic and domain independent. Examples of features are: Common Neighbours, Adamic/Adar, Shortest path distances, Katz, Preferential Attachment or Similarity ranks.

Probabilistic Models use Bayesian concepts. The main idea here is to "obtain a posterior probability that denotes the chance of co-occurrence of the node pairs we are interested in" [75]. Those approaches uses Local Probabilistic Models - e.g. Markov Random Fields or hierarchical probabilistic models.
Relational Models are focused on incorporating both node and link attributes. The approaches are based on Bayesian or Markov Networks.
Linear Algebraic Methods are general approaches that "generalize several graph kernels and dimensionality reduction methods to solve the link prediction problem" [75]. The advantage is that those approaches are able to use directly the graph adjacency matrix. The main idea is to find a transformation function F that maps A to B with minimal error solving the following optimization problem: $\min_F \parallel F(A) - B \parallel_F$, where A and B are adjacency matrices of a training and a test set with same dimensions.

Previously described approaches are limited to link prediction tasks within homogeneous graphs and networks. Since there are generally more complex networks in the real word and semantic web technologies (such as RDF graphs), we also focused on multi-relational link predictions. Most of approaches we described are also applicable to heterogeneous networks. The projection to a homogeneous network is needed but with a loss of information [76]. The adoption of multiple relations is needed to successfully predict links in heterogeneous networks. Linear Algebraic Methods use concepts of multidimensional incidence matrices called tensors, where additional dimensions can represent heterogeneity within a graph such as multiple types of links.

The complementary approch to the prediction is a Link Discovery task that is focused on finding relationships between entities within different data sources [77]. Such approaches usually use specific language to define which types of links can be discovered. They also define conditions that needs to be fulfilled. Those metrics are based on several similarity metrics for entity attributes or surrounding graph structures.

2.1.5 Preference Learning and Information Filtering

Since Preference Learning is often mentioned together with information filtering and recommendation approaches, we present only a brief overview of specific terminology and ideas. However, is is also considered as a one step of the Web Usage Mining as we already discussed in Section 2.1.2.3. As a main objective of the Preference learning is considered "a learning of (predictive) preference models from observed (or extracted) preference information" [78]. Methods for learning and predicting preferences can be seen in areas such as machine learning, knowledge discovery, adaptive and personalized user interfaces and recommender systems.

There are several preference learning problem dimensions that can be distinguished, but not limited to [78] :

- representation of preferences, type of preference model
  - utility function

  - preference relation
  - logical representation
- description of individuals/users and alternatives/items
  - identifier, feature vector, structured object
- type of training input
  - direct or indirect feedback
  - complete or incomplete relations
  - utilities

In our research we are focused on a subset of those problems related to the representation using rules and semantic description of items.

2.1.5.1 Learning Tasks

All tasks of the Preference learning are related to the domain of "learning to rank". Generally, preference learning tasks consume as an input a set of labelled items for which preferences are already known. The goal is to learn a model that can provide preferences (labelling) for an unseen set of items. Preferences usually form the total order of items according to user preferences. From the point of view of supervised learning terminology input variables form instances and class label forms a target [78].

According to the general classification of learning tasks [78], there are three main tasks: Label Ranking, Instance Ranking and Object Ranking. The most relevant task to this thesis is the Object ranking where the goal is to learn a model which produces a ranking of objects as an output. This is typically performed by assigning a score to each instance and then sorting by scores.

2.1.5.2 Recommender Systems

There are several applications of the preference learning. It is applicable for ranking problems e.g. learning to rank results of a query to a search engine. Another main area are *Recommender Systems* (RS) as a subdomain of Information Filtering (IF). The main conceptual difference between IF and RS is that IF approaches remove irrelevant items from the list of candidates based on user preferences, while RS generate the list of recommended items based on preferences [79]. Dealing with preferences is the important part for providing relevant recommendations.

RS play important role in present applications that provide huge amount of information (often referred as Information Overload problem). Providing personalized information according to user preferences is the most important aspect of currently developed applications. The primary step in a recommendation process is the preference learning using

explicit or implicit user feedback. The RS support people in their decision by filtering exponentially growing amount of information.

Based on the recent research and studies, main categories of RS are: *Content-based*, *Collaborative*, *Demographic*, *Knowledge-based* and *Hybrid* solutions [80]. There are three main components of each personalized recommendation process: a user profile, an algorithm to adjust the profile over time according to new updates and the algorithm to exploit the profile for personalized recommendations. The following overview is an abstraction of [79, 80].

Content-Based RS recommend items that are similar to those the user were interested in the past. The core is to process content or a set of features describing each item. *Vector Space Model* (VSM) [81] is considered as a representation for content items. The VSM represents each item as n-dimensional vector where each dimension corresponds to a term from the vocabulary specified for the set of items (usually a set of words for text documents or a set of attributes (genre, release date, author, ...) for items such as movies, books, ...). The preference learning process utilizes those features of items the user was interested in the past (he watched movie, visited web page, ...). Based on learned preferences, items with similar features are recommended to the user. Several similarity metrics can be used to evaluate the relevance of items for the recommendation, usually a cosine similarity metric is used.

The advantage of content-based RS is a user-independence - information about other user preferences is not required for recommendation. Transparency and explanation of recommendation can be presented using a listing of the set of features that were used as decisive for the recommendation process. There are also no problems with a new item appearing in the system, they can be recommended according to features describing the item.

Limitations of content-based RS are reflected by the set of associated features of the content (generated manually or automatically). For example for a web page, the VSM with words and their frequencies ignores the semantics, multimedia content and other meta-information associated to the web page. Features should distinguish items to provide good recommendations based on user preferences. They also suffer from overspecialisation, they recommend the content matching the user preferences. No novel or surprising recommendations are provided (often referred as a serependity issue). Finally, such RS are not able to reliable recommend items for a new user, until user preferences does not contain sufficient information.

Collaborative RS recommend items according to the paradigm of sharing tastes, often called Collaborative Filtering (CF). They identify similar users and recommend items that were previously interested by those similar users. The main representative is a nearest neighbour approach. The CF use a human judgement, therefore they are often considered more useful than the Content-based RS based on an information filtering [79]. Since no attributes describing items are usually used, the advantage of CF is a possibility to

recommend items from different domains or categories. Books, movies or music can be recommended at the same time.

Three essential prerequisites are needed to achieve good recommendations based on CF [82]: 1) the adequate number of users has to participate (more users increase the probability of finding matching user to another one), 2) there is a need to efficiently and clearly represent user interests in an easy manner, 3) the matching of similar users has to be possible.

The limitations of CF are reflected by the nature of provided user judgement. The CF is not able to provide recommendations for new users, since matching of users with no or limited history is not accurate. CF also fails with new items - each item has to be pioneered by at least one user before it can be recommended. Another issues can be observed with unusual users. Those individuals does not conform with tastes of any group in the system. CF deals with a sparsity issue too. The number of provided judgement of each user is too small if compared to the number of items in the system. CF needs an efficient way to compute similarities of users. Sparsity issues, large number of users and items leads to problems with scalability that has to solved to provide recommendations.

Demographic RS uses demographic features (location, language, age, ...) of each user to distinguish them. There is no need to have any historical information about users, their profiles or ratings. The limitation is in difficulties to collect such demographic features. Some of them can be collected automatically (location, ...), others using forms. Users usually do not like to fill any forms and share such information. The possible source of demographic features are social-networks.

Knowledge-Based RS use usually a specific domain knowledge. Knowledge-based RS use a functional knowledge: how the certain item features are important for a user (how they meet his preferences) or how the item is useful for the user. Representatives are case-based reasoning or constraint-based systems. Those systems do no need to collect any information about user, since decisions of a system are independent of individual preferences.

Hybrid combine two or more previously described approaches. Such combinations are performed to overcome specific issues with new users, new items, ... The hybrid solution focuses on the utilization of benefits of RS.

Server-Side and Client-Side Recommender Systems

Recommender Systems can be also distinguished to server-side and client-side solutions. Most of existing solutions are located on the server-side. The advantage of the server-side solution is in a centralization. All information and functionalities are located at the same place, available for providing recommendations. It is also easy to maintain such solution and deliver new versions of algorithms with respect to the clients that are

using applications. The main advantage of server-side solutions is in possibility for a wide application to many domains and scenarios. The drawback is the need to send data about user interactions from clients to servers. The user privacy can be violated. The amount of information flowing between clients and systems can grow very rapidly with increasing number of users and needs of interactive interfaces.

Pure client-side solutions address privacy preserving solutions. All data about users are stored on the client-side. There is no need to send any information from the client to the server. Another advantage is the easily scalable system and suitability for highly interactive interfaces [83]. The responsiveness is not limited by the load of servers. The drawback of such solutions is the limitation of applications. Since no information about user interactions and interests are provided to the server. The very popular and effective solutions using collaborative techniques can not be used. They are limited to information filtering techniques and for the domain and scenarios with controlled number of recommended items. Possible applications are especially in domains with unidirectional broadcast systems, where is no connection back to the service [83]. The restricted amount of possible candidates is streamed from the server and the client side solution provides the candidate using filtering or sorting techniques.

Hybrid solutions using precomputed models based on collaborative techniques can be used. Client-side solutions provide data about interactions to servers. Recommendations models are computed on servers, compressed and afterwards delivered to clients in a regular intervals. The recommendations are provided on the client-side [84].

2.2 Previous Results and Related Work

We distinguished this section to several subsections, each corresponds to the related work and relevant applications associated to topics and contributions of this dissertation thesis. Please note that more comprehensive list of relevant publications are covered by appropriate sections of author's publications.

2.2.1 Data Acquisition and Preprocessing in Web Usage Mining

The important aspect of the Web Usage Mining is a data acquisition. Many researches an applications are based on a browser cache, proxy [85, 86], desktop or browser agents [87]. Hybrid approaches are available too. All approaches were compared, but there is no clear answer which one is more or less accurate or error-prone [2].

The preprocessing and general analysis of collected data are in many researches performed using methods of clustering data about visitors: clustering on clickstream [86, 88, 89], clustering demographic or social relations and many more [57, 53]. For sequential modelling can be used Markov models [90, 91]. The methods that are close to our approaches are intrduced in researches that use association rules [92, 93]. For example, the following rule [57]: $special-offers, products/software \rightarrow shopping-cart$ identifies that the special offer positively affects sales.

There are many researches dealing with problems and challenges of the data acquisition and preprocessing. Growing number of users and web sites affects the selection of algorithms. Conventional data mining techniques were proved to be inefficient, as they need to be re-executed every time [94]. Since the clickstream in a web log is naturally incremental, there are researches focusing on using incremental mining techniques to extract usage patterns and study characteristics of users. The paper [94] investigates incremental association rule mining. From experiments is obvious that the incremental technique seems to be more efficient. Another research is based on incremental clustering of documents [95]. The algorithm is based on the semantic similarity histogram, which measures the distribution of semantic similarities within each cluster and the semantic representation of the document. Similar approach based on clustering was proposed in [96]. They generate initial model off-line and it is periodically updated. The time consuming part is done off-line, only updates are performed on-line. Another approach utilizes dynamic-agglomerative clustering algorithm [97].

In our research we are focused on the semantics and semantic technologies in the Web Usage Mining, user profiling, personalization and related tasks. One of the first researches about exploiting semantics in the WUM is [9, 10]. They propose to create a behaviour model as an ontology. Related issues covered in these researches are transforming web access activities into the ontology and deducing a personalized usage knowledge from the ontology. The semantics provides more expressive form of the knowledge representation and can enhance the quality of further recommendations [11].

The paper [98] presents a novel approach to track user interaction on a web page based on JavaScript-events combined with the Semantic Web standard Microformats to obtain more fine-grained and meaningful user information. The advantage of this approach is in application of Microformats that allows add semantic information in a safe way and it is based on open standards.

According to our survey of related works, there is a lack of existing approaches in the Web Usage Mining preprocessing focused on an aggregation of multiple interactions to one representative information. It is caused by the design of the Web Usage Mining. At the time of the creation, there were no rich user interfaces or interactive web applications allowing multiple interactions with one object item. All user implicit feedback interactions was limited to one type of interaction, usually a pageview. New interfaces allow multiple type of interactions per one object item with different interpretation. There is a need to aggregate those interactions to one representative value. We address this issue with focus on the semantics. The semantic description and possible relations between them allow to interpret them and to significantly improve representations of user preferences.

2.2.2 Semantization of Data

There are several existing approaches for semantization of data - it means providing semantic annotations [4] usually as links to a knowledge base. In our research we focused on two main approaches: *General Applicable* and *Domain specific format* (See Section 2.1.3 for more details). From the general semantization point of view, we use a tool

`http://entityclassifier.eu/` [12]. It is based on a Targeted Hypernym Discovery [99] and it is able to recognize entities in free texts written in English, German and Dutch language. Recognized entities are linked to resources from DBPedia and associates with types from the DBpedia and YAGO knowledge bases providing high semantic interoperability [12]. There are also other existing state-of-the-art tools for Named Entity Recognition and linking to a knowledge base: DBpedia Spotlight [100], Open Calais or Alchemy API. They all are limited by supported languages, limited kind of entities and limited types they can provide. Another possibility is an evaluation framework NERD [101] that provides results from various extractions tools (AlchemyAPI, DBpedia Spotlight, Extractiv, Lupedia, OpenCalais, Saplo, TextRazor, Wikimeta, Yahoo and Zemanta). Other tools that provides semantic annotation of documents and textual contents can use human annotators or other automatic approaches [102].

In the category of domain specific datasets we focused on well known datasets MovieLens [13] and MovieTweetings [14] that are available in a CSV file with no reference to any semantic knowledge base. The format is specific with the availability of a title - the single short textual information assigned to each movie and a list of associated genres. Existing approaches are based on "guessing" the *URIs* as links to the DBPedia [15]. It uses the fact that the format of URIs in DBPedia contains the title as its component. Only simple transformation to replace special and white space characters is needed. There are several issues with this approach for situations, where multilingual or not properly formatted titles are available. However, the advantage of this approach is in its simplicity. Another approaches use computing similarity measures (e.g. Jaccard coefficient) to find the most relevant entity [16, 17]. The drawback of this approach is in requirements for downloading of all available data about movies in the DBPedia and compute a similarity measure between all titles in the source dataset and and all titles of existing DBPedia movies. Our approach is focused on ad-hoc queries to the DBPedia knowledge-base using SPARQL.

The semantic aggregation and propagation within ontology we propose in our research is close to general terms: ontology generalization, propagation or mapping; spreading activation, classification or unification. From the point of view of a user profiling: an activation of interests over a full ontology [18] or a vector space preference aggregation [19]. In [18] they use a slightly modified Spreading Activation algorithm [26] to activate concepts related to starting concepts describing the content. This approach is similar to our solution, they use a propagation in user profiles. The research in [19] is focused on the aggregation from the specific movie domain and the propagation is performed with an assistance of already existing user interactions. Our approach is focused on a general propagation within an ontology without any connection to users.

2.2.3 Link Prediction

There are two main topics closely related to a link prediction method we propose, namely a tensor factorization and a relational learning. Models and methods covered by these topics are used to model multi-relational data and to perform the link prediction. Generally, most

of existing link prediction algorithms are focused on graphs and networks with single type of relation [20].

Most researches in relational learning are based on a statistical relational learning. These approaches are build upon the Bayesian or Markov networks [103, 104] or their combinations with tensor representations [105].

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 [21]. 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 [6], where each frontal slice of a tensor represents a relation. A similar model was also used in [106]. 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 [107, 108]. An extension of this work in [109] 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 [110] and SILK [77] 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 [22], 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 [23, 24, 25] 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 [111], prediction based on hypergraph [112], prediction in multi-modal networks [113], however, they are less relevant to our work.

2.2.4 Personalized Selection of Entities

In our research related to a personalized selection of predefined candidates in an RDF graph we focused on the domain of services and Web APIs. Graph-based representation of services is a relatively new approach. The authors in [114] propose service selection based on previously captured user preferences using the "Follow the Leader" model. In [115] the authors construct collaboration network of APIs and propose a social API Rank based on the past APIs' utilisations. Other approaches that rank services based on results from social network-based analyses in social API networks can be found in [116] and [117].

A particular method that relates to our work is the already mentioned spreading activation. It is a graph-based technique, originally proposed as a model of the way how associative reasoning works in the human mind [26]. The spreading activation requires directed semantic network, e.g. an RDF graph [27, 28, 29]. Inputs of the basic spreading activation algorithm are number of nodes with an initial activation which represent a query or interests of a user. In sequence of iterations initial (active) nodes pass some activation

to connected nodes, usually with some weighting of connections determining how much spread gets to each. This is then iterated until some termination condition is met. The termination conditions is usually represented as a maximum number of activated nodes or a number of iterations. After the algorithm terminates, activated nodes represent a similar nodes to the initial set of nodes.

Compared to our maximum activation method, the spreading activation does not guarantee an activation of a particular node while our method always assigns an activation if there exists an improving path between source and target nodes. Although there exist constrained spreading activation methods which utilise semantics of links [30], no version of the spreading activation takes into account the "age" of links as our method does. The maximum activation is better suited for the Web API selection mainly due to following reasons: 1) it is not known at which nodes the spreading activation terminates while the Web API selection problem uses Web API candidates as an input (target nodes), 2) the spreading activation has a local meaning of activations that indicates a measure that can be used for recommendations on data whereas maximum activation uses the value as a global measure of connectivity from source to target nodes.

There are other works in the area of Web Service discovery and selection including QoS selection [118, 119], collaborative and content-based filtering methods [120, 121, 122, 123] which are less relevant.

2.2.5 Preference Learning and Recommender Systems

In our approach we represent user profiles composed as a set of rules. Apart from practical issues, such as speed and the possibility to display the user profile in an intelligible way, rules are also considered as the most expressive form of encoding preferences [31].

The advantage of the rule-based representation is the possibility to use those profiles for client-side recommendations. There are several approaches and also companies dealing with the idea using a client-side recommender system, e.g. BBC [83]. Their approach is a hybrid solution using concepts of collaborative and content filtering recommendations. The model is build on the server side and regularly distributed to clients. Our solution is designed as a pure client-side solution. No data are transferred to servers.

To be able to build a user profile we focused on an implicit feedback and sensor inputs. Eye tracking was recently proposed as an effective way of obtaining highly detailed user feedback [124]. Content-based recommendation has so far relied either on explicit information on user interest, or on those user actions (implicit feedback), which could be interpreted as a manifestation of user (dis)interest in a certain object [125]. While the latter does not require the user to perform any extra activity, the information obtainable in this way on a particular content item is often restricted to several discrete actions (e.g. user opening a web page) and the duration between them (the time spent on a web page). While this problem is typically alleviated by collaborative filtering, gaze tracking, and physical behaviour tracking in general, helps to solve the knowledge-acquisition bottleneck by providing a rich-stream of interest data on a specific user. Our solution addresses also the "semantic gap", the difference between natural language descriptions of the labelled

data and the items to recommend, by representing the text with concepts connected to the Linked Data Cloud.

Finally, it should be noted that this research hints at immediate commercial applications (E.g. a recent patent [126] implies personalizes media during commercial breaks in videos.), the use of physical behaviour tracking of TV users by governments has also been foreseen [127].

The idea of using semantic web technologies, ontologies and Linked Data during the recommendation is widely adopted in the recent research. In the research [128], they utilize Linked Data to mitigate the new-user, new-item and sparsity problems (lack of connections between items and users) of collaborative recommender systems. They propose aggregate data from different sources. The semantics is used in either Collaborative and Content-based RS [32, 33, 34], or in the context of news recommendation [35]. The paper [11] describes an approach to combine a social and content based filtering approaches using semantically enriched user profiles. They propose a simple ontology describing each interaction of the user. The formulas for computing user and item similarity were proposed too. The paper [129] describes an integration of a semantic information drawn from a web application's domain knowledge into all phases of the web usage mining process (preprocessing, pattern discovery and recommendation/prediction). Main application of the semantic information is in the pattern discovery phase: in the sequential pattern mining algorithm. A semantic distance matrix is also used in Markov models as a solution to ambiguous predictions problem. Another approach is based on combinations: a hybrid recommender system based on ontology and Web Usage Mining [130]. Unlike other approaches that use the completed ontology as a background knowledge for clustering, this approach constructs ontology from web pages in web usage mining process. The first step is Web Usage Mining process and pattern discovery. Then the semantic information is extracted from the web site. Finally the clustering is processed.

Using rule-based approaches in recommendation tasks is also a research topic that was also covered by several researches and applications [36, 37, 38, 39, 40]. Rule induction can be used to overcome the cold-start problem [36], to integrate several recommendation systems [37] or they can be also used to represent user preferences [40].

According to our survey of related works, there is no exiting approach that is focused on using rules, semantics and is also applicable for client-side recommendations.

Chapter 3

Contributions

This chapter provides details of methods that form the main contribution of the thesis: Section 3.1 describes an approach of collecting user interactions and its preprocessing. Section 3.2 demonstrates a method for semantization with focus on a movie ratings dataset and proposes an ontology propagation. Section 3.3 provides details of our method to enhance a semantic representation using a link prediction and Section 3.4 is focused on an utilization of rich semantic representations with respect to the semantics and temporal information. Finally, Section 3.5 summarizes our work in usage of semantics for a preference learning and its applications.

3.1 Acquistion of User Interactions

This section covers the first phase of the overall methodology - acquisition of user feedback (See Section 1.2 for more details). We designed methods for a processing and an aggregation of user interactions mainly related to situations when multiple interactions per one content item are performed. The goal is to provide a relation between a user and an object the user interacted with. The final value associated to this relation represents an overall interest level of the user for the object. There is an abundance of proprietary approaches and tools, but they are domain specific. We focus on a generic design that is domain independent.

According to our review of related works, we identified following issues of the state of the art approaches from the Web Usage Mining domain. The data output by existing tracking approaches and systems are typically unsuitable for direct processing by mainstream machine-learning algorithms. One reason is data sparsity, since the distance between objects or their values is not sensitive to the semantics of the domain. The second reason is that the interactions of individual users tend to be of irregular length (number of interactions per session and per object). Without special preprocessing, such input cannot be consumed by most mainstream machine-learning algorithms. The proposed approach for the acquisition together with approaches for the semantization (See next Section 3.2 for more details) implements an aggregation step which addresses both these problems. The data are semantically enriched either immediately during the data collection phase

or during the aggregation task. Multidimensional semantic (taxonomical) description of tracked objects are processed along with implicit user feedback to the lower-dimensional output representation with a tabular form suitable for analysis with mainstream data mining algorithms.

We designed two approaches for the processing and aggregation of user interactions:

- *Heuristically defined rules* - this approch is suitable for situations when we do not have any previous knowledge or even labelled data. The drawback of this approach is in a requirement for domain knowledge or a specialist to construct rules. Manual definition of these rules is also resource-intensive and possibly error-prone. We evaluated this approach mainly in a domain of users interacting with SmartTV interfaces. Main results were published in [A.9, A.10, A.12] and contributions to projects' reports and deliverables [5].

- *Genetic algorithm* - this approach requires labelled data for its training. It uses a symbolic regression as an underlying concept to learn scoring functions. The learned functions are designed to reflect user interests based on provided historical data. We performed initial experiments on a domain specific dataset collected from a travel agency website [A.16, A.17, A.14, A.15].

3.1.1 Definitions

Interaction. *A fragment of a user behaviour that can be recorded.* Each user controls an interface of an applications using a dedicated device. Specific pieces of his behaviour that occur during the controlling of the application are interpreted as interactions on various levels of granularity (e.g. from a mouse click or movement on the device level to a confirmation of an order on the application level).

Semantic attribute. *A feature of an object expressing the semantics or categorization.* Users interact with specific objects depending on the domain and application. Each object can be described by a set of features. The semantic feature is an attribute of an object that is linked to a semantic knowledge base or it is part of a predefined taxonomy used for the categorization of objects.

Interest level. *A degree of curiosity for a specific area/topic.* Depending on user preferences and on the object the user is interacting with, user behaviour reflects his degree of curiosity. To interpret the degree numerically we use an interest level. Interest level reflects the user interest in the object. The level is normalized in interval $[-1, 1]$, where values below zero mean a negative interest and values above zero mean a positive interest. Zero is a default value and represents a neutral level.

3.1.2 Data Acquisition and Aggregation Method

An important part of the usage data processing is obtaining the information for further construction of user preference models. There are two main input channels: an implicit and explicit feedback.While there are many tools supporting this task, most of them are domain specific: consider e.g. Google Analytics or Piwik[1] for the web analytics domain. There is an abundance of proprietary solutions for other domains, such as multimedia applications.

According to the state of the art research and approaches, the problem addressed in personalization tends to be algorithmically similar for most domains. In this section we propose approaches and a tool for capturing and preprocessing user actions. Although originally inspired by the web environment, the tool is not limited to the web. We also propose a general vocabulary to unify the terms across domains [A.12]. To prove the domain independent character of the designed approach, we deployed it to a travel agency website (as an extension of Google Analytics Tracking code), it is incorporated within the LinkedTV EU Project[2] to a Smart TV player to obtain TV usage data and we also experimented with those approaches during our participations in recommender challenges (See Section 3.5).

Table 3.1: Generic terminology for data acquisition.

general	web	brief definition
object	pageview	entity, which can be interacted with
session	visit-session	series of user actions
user	visitor	the one who interacts
interest level	conversion	explicit preference level
interaction	event	user feedback
attribute	custom variable	semantic description of object

Since we focus on a design of a general approach, we use a generic terminology fitting not only web analytics, but also other areas incl. the TV domain (ref. to Table 3.1). The table lists the generic terms along with what we consider to be the equivalent term used in the web analytics domain. We also present terminology for the multimedia domain, as used in the LinkedTV project.

Let consider a set of users $U = \{U_0, \dots\}$ and a set of objects $O = \{O_0, \dots\}$ each user can interact with. Interactions are defined as a set of tuples $I = \{(U_i, O_j, IA_k), \dots\}, U_i \in U, O_j \in O$, where each object and interaction are additionally described by a set of features that we call attributes $IA = \{IA_0, \dots\}$ for interactions' attributes and $OA = \{OA_0, \dots\}$ for objects' attributes. Interaction attributes mostly define a type of interaction and contextual features (e.g. time of interaction, geolocation information etc.). Object attributes define a semantic description in form of key-value pairs or identifiers to a knowledge base such DBpedia. Each object can be described by more than one identifier.

[1] `http://piwik.org`

[2] `http://www.linkedtv.eu/`

The goal of the collecting service and the aggregation is to provide a relation between a user and an object the user interacted with. The final value associated to this relation represents an overall interest level of the user for the object: $(U_u, O_o, interest\_level)$.

Since the user can provide multiple interactions per one object , the goal is also to aggregate all interactions per one object item to one representative relation between the user and the object. This relation would mean a level of interest:

$$\begin{aligned}(U_u, O_o, interest\_level) = aggreagtion(\{&(U_u, O_o, \{type = type1, \dots\}),\\ &(U_u, O_o, \{type = type2, \dots\}),\\ &\dots\})\end{aligned} \tag{3.1}$$

In our solution, we experimented with following approaches for the processing and aggregation of interactions: *1) Heuristically defined rules and 2) Genetic algorithm.*

Example 3.1.1. ***Illustrative scenario of a user using a web application***

Let consider the following simple scenario: A user uses a device to browse the web and he would like to search and watch videos about his favourite sport - Football. The user starts his session on a landing page of a video streaming site. He selects a Sports category and performs filtering. From the provided list of videos he starts watching the first video. Since the video is about Football in another country, he is not satisfied after couple of seconds, stops playing and returns to the list. He selects the second video. First part of the video is about a reconstruction of a stadium, he skips the first part and watch the rest of this video about the football match. He likes the second part of this video and bookmarks it.

The scenario covers both implicit and explicit feedback from the user. Implicit interactions are visit of a web page or playing a video. Explicit interaction is bookmarking of a video to the list of favourites for later use.

Example 3.1.2. ***Formally described scenario***

Based on the previously described example, we collected following set of interactions. User $U_{example}$ interacted with four different objects. Since our research is focused on semantically enriched objects, each object is described by a feature set representing the semantics. For our illustrative example:

- $O_{landing}$ *- landing page of a streaming service:* $OA_{landing} = ()$
- $O_{sports\_category}$ *- sports category page:* $OA_{sports\_category} = (Sport)$
- O_{video_1} *- the first video about football in another country:* $OA_{video\_1} = (Football, England)$
- $O_{video\_2-part1}$ *- the first part of the second video about the reconstruction of a stadium:* $OA_{video\_2-part1} = (Football, SportStadium)$
- $O_{video\_2-part2}$ *- the second part about the football match:* $OA_{video\_2-part2} = (Football, Germany)$

The user performed following interactions:

(I_1) $(U_{example}, O_{landing}, \{type = visit, time = t_1\})$

(I_2) $(U_{example}, O_{sports\_category}, \{type = visit, time = t_2\})$

(I_3) $(U_{example}, O_{video\_1}, \{type = play, time = t_3\})$

(I_4) $(U_{example}, O_{video\_1}, \{type = view, time = t_4\})$

(I_5) $(U_{example}, O_{video\_1}, \{type = stop, time = t_5\})$

(I_6) $(U_{example}, O_{sports\_category}, \{type = visit, time = t_6\})$

(I_7) $(U_{example}, O_{video\_2-part1}, \{type = play, time = t_7\})$

(I_8) $(U_{example}, O_{video\_2-part1}, \{type = view, time = t_8\})$

(I_9) $(U_{example}, O_{video\_2-part1}, \{type = skip, time = t_9\})$

(I_{10}) $(U_{example}, O_{video\_2-part1}, \{type = skip, time = t_{10}\})$

(I_{11}) $(U_{example}, O_{video\_2-part2}, \{type = play, time = t_{11}\})$

(I_{12}) $(U_{example}, O_{video\_2-part2}, \{type = view, time = t_{12}\})$

(I_{13}) $(U_{example}, O_{video\_2-part2}, \{type = bookmark, time = t_{13}\})$

,where $U_{example}$ is our user, O_i is an object, type is the type of the interaction, time represents timestamp of the interaction.

3.1.2.1 Heuristically Defined Rules

Heuristically defined rules are appropriate for use cases when we do not have any already existing ground truth: data with explicit expression of interests related to each interaction. Each rule assigns a level of interest for each or subset of interactions. This approach also requires a domain knowledge. Domain specialist can usually prepare a set of rules for the aggregation. Although the construction of rules is straightforward, manual definition of these rules is resource-intensive and possibly error-prone. For example a following set of rules:

(R_1) $type = skip \rightarrow interest\_level = interest\_level - 1$

(R_2) $type = view \rightarrow interest\_level = interest\_level + 0.1$

(R_3) $type = bookmark \rightarrow interest\_level = interest\_level + 0.5$

Table 3.2: Example of the tabular representation suitable for machine learning algorithms.

Identifiers		Semantic Attributes					
user	**object**	**Sport**	**Football**	**England**	**Germany**	**SportStadium**	**interest**
$U_{example}$	$O_{landing}$						0 (neutral)
$U_{example}$	$O_{sports\_category}$	1					0 (neutral)
$U_{example}$	O_{video_1}		1	1			0.1 (positive)
$U_{example}$	$O_{video\_2-part1}$		1			1	-1 (negative)
$U_{example}$	$O_{video\_2-part2}$		1		1		0.6 (positive)

Those rules can be applied on a set of interactions from our example, where the aggregation step sums up changes of the interest level performed by the rules per each object. The final value is normalized to interval $[-1, 1]$. Output relations can be further used in preference or machine learning algorithms to express a user preferences. Other attributes of objects, including the semantics, can be also used to extend each relation. The output tabular representation can be constructed using conversion to binominal attributes for expressing a presence of values for specific semantic attributes. Since semantic attributes can be sparse or of uneven length. We designed an ontology propagation method that can be used for such use cases. We would like to refer the reader to Section 3.2.

Example 3.1.3. ***Application of heuristically defined rules:***
If we apply the rules to interactions from our example we get following values:

- $(U_{example}, O_{landing}, 0)$ - *No match of rules for* I_1, $\emptyset$ *rules applied.*
- $(U_{example}, O_{sports\_category}, 0)$ - *No match of rules for* I_2, $\emptyset$ *rules applied.*
- $(U_{example}, O_{video\_1}, 0.1)$ - *Rule* R_2 *matches* I_4.
- $(U_{example}, O_{video\_2-part1}, -1)$ - *Rule* R_2 *matches* I_8 *and* R_1 *matches* I_9 *and* I_{10}.
- $(U_{example}, O_{video\_2-part2}, 0.6)$ - *Rule* R_2 *matches* I_{12} *and* R_3 *matches* I_{13}.

Since we have no evidence, our experimental user has neutral interest in the landing and sports category page. he is a bit interested in the first video due to his watching of at least part of it. He skipped the first part of the second video and it means he is not interested at all. He is definitely interested in the second part of the second video. Example of tabular representation is on Table 3.2.

3.1.2.2 Genetic Algorithm

In contrast with Heuristically Defined Rules, the Genetic Algorithm requires a training set [A.15]. The training set has to hold not only regular interactions but also explicit expressions of user interest levels. Those explicit expressions are used as a ground truth for a learning algorithm. As explicit expression can be considered an explicit feedback interaction (e.g. Bookmarking an object, ...) or an implicit feedback interaction (e.g. user

Table 3.3: Set of allowed terminals and operations for symbolic regression.

Operation	Description
	Terminals
Const	Real constant
Var	Variable representing input attribute
	Unary Operations
Neg	Negation of an argument
Sin	Computes sine of an argument
Cos	Computes cosine of an argument
	Binary Operations
Add	Adds arguments
Sub	Subtracts arguments
Mul	Multiplies arguments
Div	Divides arguments
Max	Maximum of arguments
Min	Minimum of arguments
Left	Tree pruning - cuts off the right branch
Right	Tree pruning - cuts off the left branch

completed viewing video, user bought a specific product - conversion in a web analytics, ...).

The input for the algorithm is the training set, that is composed from interactions. Some of them are marked as the explicit expression of interest (binary attribute gt): $\{(U_u, O_o, \{gt = 0, \dots\}), (U_u, O_o, \{gt = 1, \dots\}), \dots\}$

The goal of this approach based on a genetic algorithm is to learn so called weight function. The weight function has to be able to assign an "interest" value (also called weight or score) to each interaction based on all attributes associated to the interaction. Assigned interest value reflects the importance of the interaction to the overall user interest. We use a symbolic regression as a main concept to compose the weight function that uses the genetic programming as an underlying concept [A.17]. The weight function is generally defined as:

$$weight(I_j) = interest\_level_{I_j} \tag{3.2}$$

,where $interest\_level_{I_j}$ is a real number from interval $[-1, 1]$ reflecting the overall interest of the user expressed by the interaction and I_j is an interaction. The interest level can be influenced by any attribute associated with the interaction.

Example 3.1.4. ***Weight function in illustrative example***

Let consider two selected interactions of our example user from the previous section: I_4 *and* I_{12} *with objects about* $(Football, England)$ *and* $(Football, Germany)$*, respectively.*

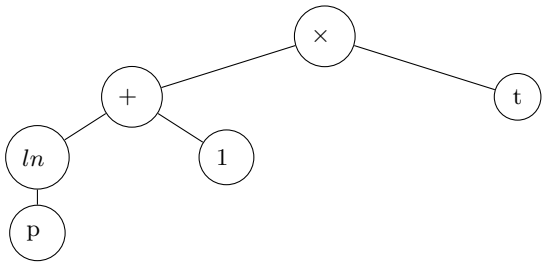


Figure 3.1: Symbolic regression - example of the syntactic tree for the formula $(ln(p)+1)\times t$.

In case our user is interested in German Football: $weight(I_4) < weight(I_{12})$ and thus $interest\_level_{I_4} < interest\_level_{I_{12}}$. I_{12} should reach higher interest level, because the object was about German Football that is part of the user interests.

Based on previous researches [131], [132], the weight function mostly depends on the time spent on the page (TSP) - a time interval between two successive interactions. Since TSP could indicate how much reading the user dedicated to a specific page, it is considered as the most important variable for the domain of web usage mining and web analytics. Another related and significant variable is a page/interaction order. The more interactions the user perfomed the more related and close to the interest they are. The user is gradually getting to the object that is reflecting his interests. One possible form of the weight function defined empirically [132] should be: $(ln(p) + 1) \times t$, where p is the page/interaction order and t is TSP. Higher interaction order and more spent time positively influence the weight.

Symbolic Regression

Symbolic regression [133] is an approach to analyse a search space in order to find a model that fits the space and requirements. The output of the symbolic regression are formulas composed from predefined operations. The advantage of symbolic regression is a readable form of the output (no black-box solution) and straightforward application (pure calculation of the constructed formula). The provided formula can be naturally used to assign the interest value to each interaction.

Symbolic regression uses a genetic programming as a way to learn the final set of formulas. Each formula is internally represented as a syntactic tree composed from allowed operations and terminals. Table 3.3 presents the set of allowed operations for the symbolic regression we propose. Example of the syntactic tree for the formula $(ln(p) + 1) \times t$ is on Figure 3.1. From the perspective of the genetic programming, the symbolic regression starts with an initial generation - a set of randomly generated formulas. The initial set of formulas is evolved using genetic operations: mutation and crossover. The mutation replaces a randomly selected subtree with another randomly generated subtree. The crossover exchanges a randomly selected subtree from one formula with a randomly selected subtree from the second formula.

Algorithm 1: Generic structure of a genetic algorithm

input : Train dataset D_{train}
Population size *populationSize*
Maximum number of iterations t_{max}
Minimum Fitness threshold *limit*
output: Best formula (weight function) $weight_{top}$

```
1  begin
2      // initialization
3      t = 0
4      // start with random formulas
5      P = randomPopulation(populationSize)
6      while t < t_max do
7          t = t+1
8          Q = ∅ ∪ P
9          while size(P) > 0 do
10             // select candidates using roulette wheel
11             alpha1 = rouletteWheel(P)
12             alpha2 = rouletteWheel(P)
13             // crossover or mutate with certain probability
14             if random()<pCrossover then
15                 crossover(alpha1,alpha2)
16             if random()<pMutation then
17                 mutate(alpha1)
18             if random()<pMutation then
19                 mutate(alpha2)
20             Q = Q ∪ alpha1 ∪ alpha2
21         P = bestFrom(Q, populationSize)
22         // compute fitness for all and return top
23         topFitness = evaluateFitness(P)
24         if topFitness >= limit then
25             break
26     // return equation with best fitness
27     returnBest(P)
```

To evaluate the quality of each formula, we need a fitness function. It computes how well the formula fits the provided train data. The best formula has to minimize error across all train data. The core of fitness function is computed as a sum of differences $diff$ between results of the formula f applied on all items and correct values, where item is one entry from the training data set and *correct_value* is the target value from the training

data set [A.15]:

$$diff = \sum_{i=1}^{\#item} \frac{|eval(f, item_i) - correct\_value_i|}{max(eval(f, item_i), correct\_value_i)} \tag{3.3}$$

Fitness function fit is computed as a reciprocal value of $diff$.

$$fit = \frac{1}{diff} \tag{3.4}$$

Algorithm 1 describes a generic structure of our genetic algorithm. The genetic algorithm starts with a random population and iteratively evolves the population over time. In each iteration, the population is crossed and mutated. The best candidates are selected to the next iteration. The algorithm stops if the maximum number of iterations is achieved or any formula satisfies the threshold for the fitness function.

Fitness Functions

We propose two alternative fitness functions in our approach: *Promoting the most similar interaction* and *Promoting sum of attributes' participations* [A.14]. They differ in the support of specific attributes during the evolution of formulas in the symbolic regression. Both fitness functions were designed experimentally based on previous researches and preliminary experiments during the design of our approach [A.15].

Promoting the most similar interaction. The most similar interaction from the set of interactions prior to the explicit expression is promoted. For the set of interactions belonging to one user, the interactions that are the most similar to the interaction annotated as the ground truth are expected to have higher score than the others. The computation is focused only on one most similar interaction.

Following equations formally define the fitness function. From all interactions x of a visit V, the highest weight is associated to an interaction mx (Equation 3.5), which is the most similar interaction to a conversion xc across all attributes (Equation 3.6).

$$mx = \operatorname*{argmax}_{x \in V} weight(x) \tag{3.5}$$

$$mx = \operatorname*{argmin}_{x \in V} \sum_{i=1}^{\#attributes} \frac{|a_i^x - a_i^{xc}|}{max(a_i^x, a_i^{xc})} \tag{3.6}$$

Example 3.1.5. ***Promoting the most similar one***

Let the ground truth interaction is annotated by $(Football, Germany)$ *and prior interactions are about* $(Football, SportStadium)$ *and* $(Sport)$*. Comparing the interactions, the first is more similar either using a simple measure calculating the overlap of features or even from the semantically point of view. The values assigned by the learned weight function should reflect this fact. The weight function assigns higher value to the first interaction.*

Promoting sum of participations. While the first fitness function was focused on one interaction, the second one considers influence of more interactions from the user session prior to the ground truth. The motivation is that users tend to perform more interactions that are related to the overall interest. More interactions are more similar either using the overlap measure or semantics. The overall proportion of attributes participations should reflect the ground truth.

Following equation formally define the fitness function. For each (semantic) attribute a is computed a sum of weights grouped by each value av (Equation 3.7). The maximum sum of weights must be assigned to the value, which is equal to the value of the ground truth interaction in the same attribute (Similar to Equation 3.6). Similarly to previous one, sum of differences hast to be minimized.

$$mx = \underset{av}{\operatorname{argmax}} \sum_{\forall i: a_i = av} w(x_i) \tag{3.7}$$

Example 3.1.6. ***Promoting sum of participations***

Let the ground truth interaction is annotated by $(Football, Germany)$ *and prior interactions are about* $(Football, SportStadium)$*,* $(Football, England)$ *and* $(Sport)$*. Since one attribute of two first interactions is the same as in the ground truth, the sum of weights assigned to both first two interaction should be higher than the weight for the third one (e.g. 4,4,5 respectively) . Sum of both first weights should be thus higher than the third one (*$(4 + 4) > 5$*).*

The proposed approach is originally mainly designed for one interaction performed with one content item. From the point of view of the aggregation, the genetic algorithm is dependent on a set of available attributes. In situations when the user performed multiple interactions per one content item, all semantic attributes are the same for multiple interactions per one content item. The other attributes describing interactions itself such as type of the interaction or temporal aspects (e.g. TSP) become the decisive for computing weights.

Example 3.1.7. ***Multiple interactions per one content item***

Let one ground truth interaction is annotated by $(Football, Germany)$ *and prior interactions are* $(Football, Germany, Play)$*,* $(Football, Germany, Bookmark)$*. Second ground truth interaction is annotated by* $(Politics, England)$ *and prior interactions are* $(Politics, England, View)$*,* $(Politics, England, Bookmark)$*. Semantic attributes does not provide sufficient information to properly compute the weight function. Other attributes can help to tackle this issue. Since the bookmark is repeatedly performed and same semantic attributes are also available, Interactions representing bookmarking action should get higher values assigned by weight function.*

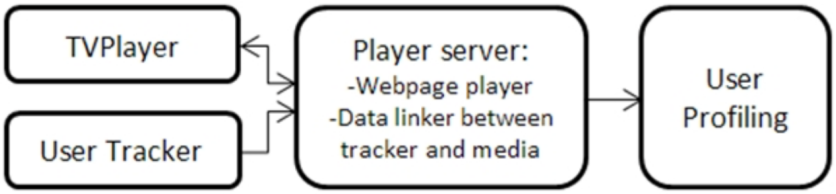


Figure 3.2: Main view to LinkedTV trials schema [5].

3.1.3 Experiments with Heuristically Defined Rules

The proposed approach based on heuristically defined rules was evaluated within experiments of a dedicated workshop [A.9] and user trials of the European LinkedTV[3] project, where the author of this thesis was involved [5]. The project was focused on an analysis of multimedia content, semantic annotations and personalized presentations to users in front of a Smart TV. The important part is dealing with a user tracking and modelling, with respect to the semantics of content items. The innovative techniques around body behavioural tracking are used.

To collect relations between users and semantically described content in the LinkedTV, two sources of interactions are considered: 1) user interactions in the player (e.g. buttons pushed on a remote control, viewed additional related content etc.) 2) user attention tracking (user is watching the screen, how many persons are in front of the TV etc). Kinect sensor is used to track bodies and face directions.

Partners of the project were responsible for the implementation of the attention tracking solution, selection and preprocessing of a media content, questionnaires and actual running of trials with invited users. We provided our implementation of a TV player and server performing the acquisition and the proposed aggregation. We also processed and evaluated the results together with additional experiments presented within this section. Tomáš Kliegr as the co-author of papers mainly participated on the design and architecture of systems and he contributed to evaluations with specific baseline algorithms.

3.1.3.1 Trials

The setup and trials were coorganized with other partners of the LinkedTV project. We would like to refer the reader to experiments [A.9], deliverables and project reports [5] for more details. Please note that particular details on trials are listed to demonstrate the overall methodology.

Trials Setup. Figure 3.2 shows the main conceptual view of the trials setup. The *Player server* is the main part in the middle of the pipeline. It is responsible for providing the media content that is presented in *TVPlayer* (Example of the player is on Figure 3.3). The server collects interactions from control buttons of the player. It also collects and

[3] `http://www.linkedtv.eu/`

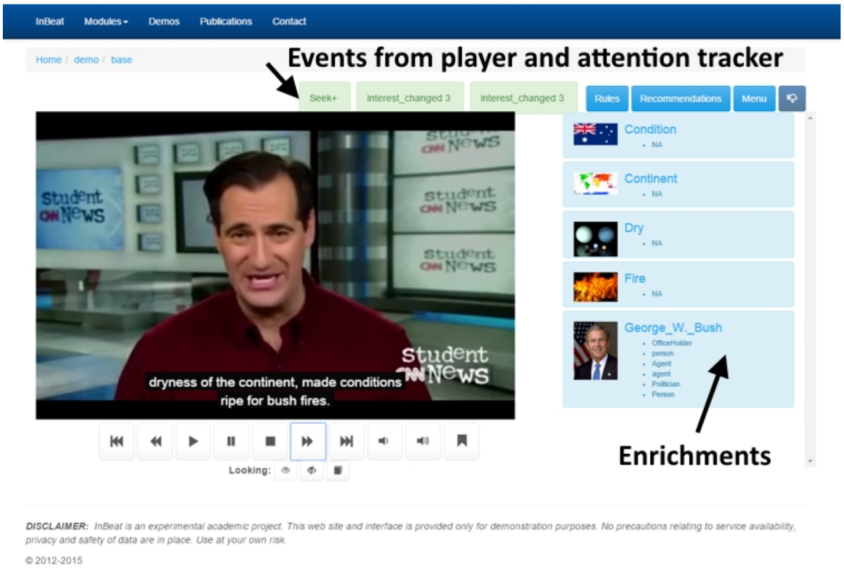


Figure 3.3: Test player. In addition to the video, it allows people to interact with the video and see the enrichment. [5].

forwards interactions detected by *User Tracker* that are visually presented on the screen of the player. Finally, the server provides processed and aggregated outputs for user profiling that is out of scope of this thesis.

More detailed setup is presented on Figure 3.4. It is focused on the behavioural body tracking using Microsoft Kinect that is one of the main objectives of trials experiments.

Content. For trials we selected a YouTube video as a media content presented to users in the player. The displayed content on the player consists of a selection of 7 videos with a Creative Commons Licence of a US news show. It is a mashup of CNN Student News for learning English. These TV shows are easy to understand even for non-native English speakers, and their subtitles are provided. The mashup covers seven different topics (North Korea, plane crash in Asia, Christmas tree recycling, bush fire in Australia, flu, American football, evolution of technologies in the last decades) for a total duration of about 13 minutes. The video is divided into specific pseudo-shots (corresponds to fragments of available subtitles). Each pseudo-shot subtitle is analysed using Named Entity Recognition Tool and detected entities are presented as enrichments alongside the video frame (Figure 3.3). All entities are click-able and opens corresponding wikipedia page to explore details related to the entity.

Users are invited to answer a questionnaire which focuses only on four of the seven topics. The users have simple control over the video player (play, pause, go forward for a few seconds, go backward), but they can also click on enrichment links.

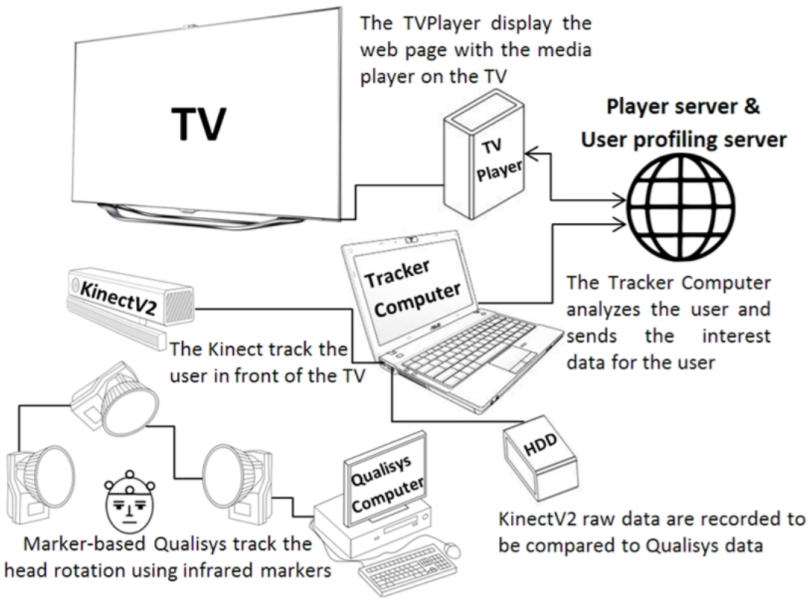


Figure 3.4: Main view of the experimental setup (top: viewer side, bottom: tester side) [5].

Questionnaires. Different questionnaires are submitted to the viewer. The first one concerns content-related questions linked to the viewer interest and it aims into simulating common interests between the different viewers. The participants are asked to answer a quiz of 7 questions, whose answers can be extracted either from the video or in the displayed enrichment. The questions concern a plane crash, a bush fire, the prediction of the flu and the current technologies compared to the previsions from the science fiction movies from the eighties. The questions are made in order to provide the viewer with a secondary interest (easy questions about plane crash and fire in Australia) and with a main interest (complex questions on flu prediction and new technologies).

The seconds questionnaire focuses on the assessment of all the presented enrichments. The user needs to rate "-1" if no interest, "0" if secondary interest and "1" if main interest each set of enrichments. Those values are used as a ground truth for the evaluation.

Trials procedure. The user has 2 minutes to get used to the system with a different video content: s/he can browse on the player, look at it, look at the second screen (tablet). Once this is done s/he has time to read the content-related questionnaire which shows him or her the main and secondary interests. On the four topics of interest the first two are of secondary interest which imply simple questions in the questionnaire and the last two are main interest which imply complex questions on the questionnaire. For the 4th topic, the user must browse the enrichments to be able to answer one of the questions. During the viewing the user also has a second screen (tablet) where s/he needs to play a game and get the maximum possible score. This game that s/he plays to is concurrent to the questions

Table 3.4: Predefined set of rules used in trials.

Action	Interest change	Interpretation
Play	+0.01	Play Video
Seek+	-0.5	Go forward 10s
Seek-	+0.5	Go backward 10s
Bookmark	+1	Bookmark a shot
Detail	+1	View details about entity
Volume+	+0.5	Increase volume
Volume-	-0.1	Decrease volume
Viewer looking 0	-1	Viewer not looking to screen
Viewer looking 2	-1	Viewer looking to second screen (tablet)
Viewer looking 1	+1	Viewer looking to main screen
Interest changed 0	-0.2	Viewer switched the screen
Interest changed 1	+0.2	Looking between 1.5 and 5 seconds to main screen
Interest changed 2	+0.5	Looking between 5 and 15 seconds to main screen
Interest changed 3	+0.8	Looking more than 15 seconds to main screen
Interest changed 4	-0.3	Looking between 1.5 and 5 seconds to second screen
Interest changed 5	-0.7	Looking between 5 and 15 seconds to second screen
Interest changed 6	-1	Looking more than 15 seconds to second screen

on the video content. The main idea behind this is that the user will mainly watch the main screen when interested by the content and play the game when the video does not bring any information to answer to the quiz.

3.1.3.2 Evaluation

Evaluation of implicit contextualized profiling aims at the assessment of the interest computation work-flow that is used to process collected implicit data and to generate a basis for creation of user profiles.

Setting. To compute the final interest from the interactions we use an experimentally predefined set of rules that either increase or decrease the default value of the interest. The default interest value is 0, which is interpreted as the neutral level of interest. The set of rules used in the trials is given in Table 3.4. If the computed value of interest exceeds 1 or is lower than -1, it is replaced by 1 or -1, respectively. The final value is extended with information describing the pseudoshot: identifier and a set of entities with corresponding DBpedia types. The final export of interest for each pseudoshot is compared with the explicit values provided by the participants of the trial.

Table 3.5: Overview of collected interactions.

Action	Ratio
Interest changed	69.50%
Viewer looking	24.44%
Seek+	4.35%
Seek-	0.95%
Pause	0.31%
Play+	0.30%
Detail	0.12%
Previous	0.01%
Next	0.01%
Stop	0.01%

Evaluation Metrics and Results

For evaluation we used the following ground truth and baselines:

- *Trials ground-truth*: explicit annotations of interest from questionnaires filled in by participants. Participants annotated each pseudoshot of video with value that represents negative,neutral or positive interest (-1,0,1).
- *Random baseline*: Baseline data computed as a random value from interval [-1,1] per pseudoshot.
- *Most frequent baseline*: Baseline algorithm where all shots are labelled with the most frequent value filled by participant in a questionnaire.

The interest computation algorithm was evaluated used in two setups:

- *Rules*: outputs computed using a set of interactions per pseudoshot and a predefined set of rules to interpret importance of an interaction. It provides outputs as real values from interval [-1,1] for each pseudoshot.
- *Rules-Window*: Sliding window approach - a mean value of the current, the previous and the following interest value is aggregated in order to decrease influence of transitions between subsequent pseudo-shots.

Basic trial ground truth statistics: 20 participants and 13,533 interactions (678 on average per participant). Table 3.5 presents the overview of the interactions collected during the trials.

As a metric for this evaluation we used Mean Absolute Error (MAE) computed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|t_i - y_i| \tag{3.8}$$

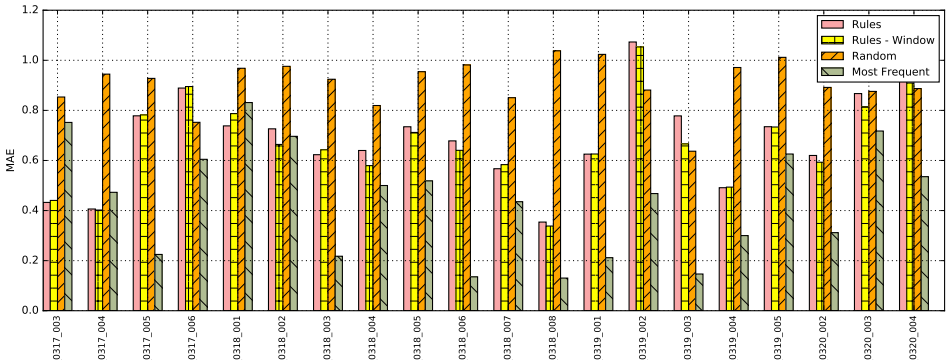


Figure 3.5: Evaluation results: MAE for each participant [5].

Table 3.6: Evaluation results: Macro-Average MAE for all participants.

Most Frequent	**Random**	**Rules**	**Rules-Window**
0.44	0.89	0.69	0.67

where n is a number of shots in user trials, t_i is interest value for specific shot from Trial and y_i is value of *Rules*, *Rules-Window*, *Random* or *Most Frequent*.

Figure 3.5 depicts the results of MAE for each user who participated in trials. The results of MAE averaged for all participants is in Table 3.6. Figure 3.6 depicts the average interest provided by server and compares it to the average interest annotated by the trial participants. On this figure two plots have good correspondence and we can see 4 peaks. The 2 first peaks correspond to the two videos where people had to answer to simple questions in the content questionnaire (which means that they have a medium interest for those videos). The 2 last peaks correspond to the 2 videos where people had difficult questions (and even they needed to go into the enrichment for the 4th video). The data from the Kinect (Figure 3.7) and after server processing (Figure 3.6, red curve) both also have higher values for those 4 peaks. The two first peaks are less well detected because simply the questions were easier, the user answered very quickly and then started to play the game on his tablet (which is interpreted as a disinterest to the related media segment). The two last peaks with difficult questions were much better spotted as the viewer needed to pay more attention to answer the related questions. For the last question the click on enrichment was quite complex and it logically brought a high interest to this video, that is why almost all the video has a high value of interest. Those results show that there is a coherence between the user ground truth interest and the one obtained by using the Kinect and the player interactions.

Figure 3.7 presents the data on "looking at the main screen" averaged across all users participating in the the trial.

The execution of the trials generated 173 pseudoshots (video fragments) for which user judgement is available. The results obtained by the evaluated workflow on this dataset indicate that the used feature set allows to estimate user interest in video content with

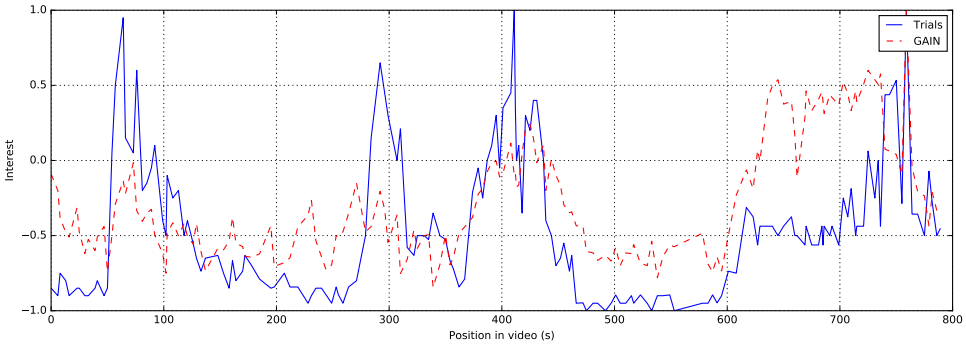


Figure 3.6: Timeline of the average interest from server vs the ground truth from the questionnaires [5]

.

significantly higher mean average error than a random guess. However, it should be noted that the default workflow is outperformed by the most frequent baseline.

The size of the groundtruth dataset (over 3000 instances) that came out of the final trial, allows to employ machine learning techniques instead of the hand-coded rule sets. As shown in the following subsection, the supervised approach provides an improvement over the most frequent baseline.

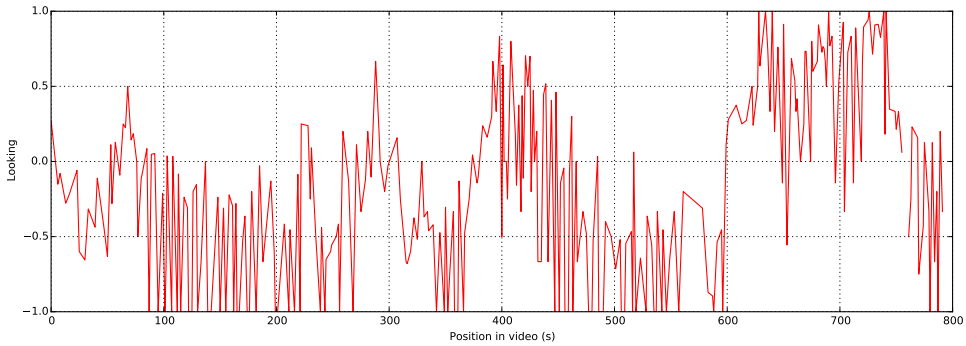


Figure 3.7: Timeline of the average viewer looking at the main screen (computed from the collected interactions) [5].

3.1.3.3 Experiments with Supervised Classifiers

The ground-truth dataset that contains explicit interest levels per shot and a set of recorded interactions per shot allows to build a classifier that can "learn" relations between the interactions and the interest level. The model can provide improved classification results over the ad-hoc rules presented in Table 3.4 for which the MAE is reported in Table 3.6.

Table 3.7: Example of matrix for experiments with classifiers.

p_pause	**p_ic_1**	**...**	**c_pause**	**c_ic_1**	**...**	**f_pause**	**f_ic_1**	**...**	**gt**
0	0	...	1	1	...	0	1	...	1
1	1	...	0	1	...	1	0	...	0
0	1	...	1	0	...	1	1	...	-1
1	0	...	1	1	...	0	0	...	1
...	...	...	...	...	...	...	...	...	...

Our benchmark includes he following set of supervised algorithms: Most Frequent, SVM with linear kernel, brCBA [A.19], k-Nearest Neighbour(KNN) and majority class voting. The Most Frequent classifier simply predicts the most frequent class. The SVM was run with default parameters (C=0, $\epsilon = 0.001$, shrinking). For kNN we used k=20 as empirically obtained value of the k parameter. The setting of the brCBA classifier was as follows: minConfidence = 0.01, minSupport= 0.025. The classifiers included into the majority class voting scheme include kNN, linear SVM and Most Frequent.

Input received by server for each participant and pseudoshot was represented as a fixed-length vector. Three binary features were generated for each possible values of the actions listed in Table 3.5: one feature corresponding to the event value in the current pseudoshot, one feature for the preceding pseudoshot and one feature for the subsequent pseudoshot.

The matrix created for each user thus contains columns that represent interactions relating to the previous shot, current and following shot (a.k.a. sliding window approach). Example of the matrix is in Table 3.7. Column names are prefixed with *p* , *c* , *f* for previous, current and following shot respectively, pause represents Pause interaction, *ic* 1 is Interest Changed with value 1. The last column (*gt*) holds the value provided by the participant as the interest level ground-truth.

We performed 10-Fold stratified cross-validation for each dataset (there was one dataset for each trial participant). Only 15 participants are used for experiments: *umons_0318_008* was excluded because of a very small variance of the assigned interest value in the questionnaire and the first four testing participants (*umons_0317_003...umons_0317_006*) were excluded since they were used to verify the trial setup. The evaluation results are presented in Table 3.8. As the evaluation metrics, we use the Mean Absolute Error (MAE), which unlike accuracy reflects the different costs of misclassification (the predicted value of interest is one of the three values -1,0,1).

The best performing algorithm with respect to the overall MAE is voting, which is a simple meta learning algorithm. However, from the perspective of the won-tie-loss record, the best performing algorithm is our brCBA.

Apart from the best won-tie-loss from the considered classifier, other advantage of brCBA is that the result of the algorithm is a rule set. Rules are in general one of the most easily understandable machine learning models. Within the personalization workflow, the fact that brCBA outputs rules, in theory, allows the model to be presented to the user, edited by the user (user discards some rules) and then deployed to server instead of the

Table 3.8: Classification results: MAE for all participants.

Participant	Most Frequent	SVM - linear	brCBA	KNN	Voting
umons_0318_001	0.828	0.627	0.676	0.577	**0.564**
umons_0318_002	0.704	0.594	0.603	**0.569**	0.582
umons_0318_003	0.214	0.214	**0.212**	0.214	0.214
umons_0318_004	0.505	**0.475**	0.541	0.499	0.481
umons_0318_005	0.513	0.525	**0.511**	0.550	0.513
umons_0318_006	0.136	0.136	**0.133**	0.136	0.136
umons_0318_007	**0.440**	0.463	0.492	0.451	0.440
umons_0319_001	**0.107**	0.107	0.213	0.107	0.107
umons_0319_002	0.521	0.521	**0.471**	0.526	0.521
umons_0319_003	0.222	0.222	**0.140**	0.222	0.222
umons_0319_004	0.303	0.303	**0.300**	0.303	0.303
umons_0319_005	0.522	0.522	0.759	**0.509**	0.522
umons_0320_002	0.314	**0.255**	0.302	0.273	0.255
umons_0320_003	0.709	**0.701**	0.796	0.701	0.708
umons_0320_004	0.607	0.607	**0.555**	0.618	0.607
Average	0.443	0.418	0.447	0.417	**0.412**

predefined set of rules presented in Table 3.4.

We consider further improvements in the induction of interest classifiers as one of the most viable directions of further work, as the accuracy of interest estimation is one of the key inputs for the personalization workflow.

3.1.4 Experiments with Genetic Algorithm

In this section we describe our efforts to use the symbolic regression to learn weight functions from web analytics usage data. As the input we use a semantically enriched clickstream from a travel agency web site.

3.1.4.1 Dataset/Data Collecting

As we already presented, the input for experiments is a semantically enriched clickstream. Users of a travel agency web site are browsing annotated web pages and produces a stream of interactions. Annotated web pages are standard web pages with additional information providing more details for each page. The annotations are provided directly by the travel agency web site and associated to each interaction. The set of characteristics used to describe each tour on the travel agency web site was predefined experimentally. It is also limited by the characteristics presented on pages at the time of collecting data (Summer 2010). Used characteristics: transport type, destination country and city, accommodation, price. Values of characteristic attributes can be represented as simple strings. For addi-

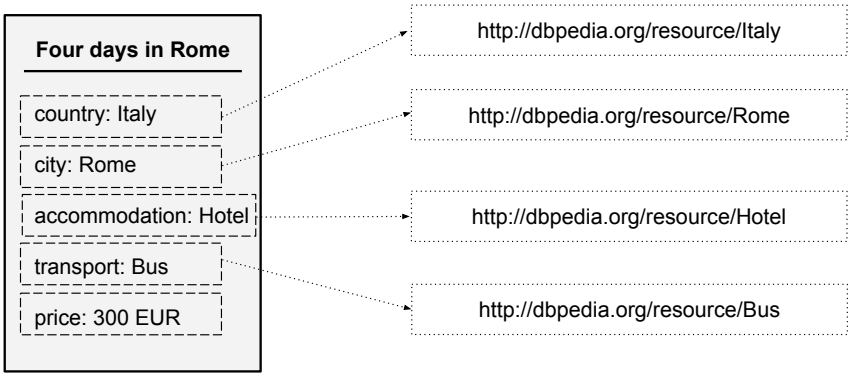


Figure 3.8: Semantic annotation of a tour offered by a travel agency.

tional processing it is appropriate to represent the attributes as concepts from an ontology or as URIs linked to a knowledge base such as the DBpedia.

Example 3.1.8. ***Example of a tour annotation.*** *There is a tour to Rome in Italy, transport is handled by a bus, accommodation is in a hotel, price 300 EUR. All values are linked to resources in the DBpedia knowledge-base. Illustration is on the Figure 3.8. Those links to the DBpedia allow us to extract additional features and better understand relations in data during the following processing of data.*

To collect the semantically enriched clickstream we used our tool InBeat - GAIN (See Section 3.1.5.1). At the time of collecting data, there was a possibility to forward all data from a standard Google Analytics measure code to any local service [4]. The measure code also allows to configure custom variables that we used to set semantic characteristics. GAIN was used as an intermediary element to process such inputs and save in our generalized format described within this chapter.

Example of a clickstream is on Table 3.9. Each row represents one view of a specific web page (pageview). This format we use as the input for the learning of weight functions. In all examples we use the weight function, which was defined in Section 3.1.2.2 as an empirical function: $w(x) = (ln(po) + 1) * tsp$. The reason is the simple and understandable form.

3.1.4.2 Preprocessing

Visit order, pageview order, time spent or prices are numerical attributes. Unlike numerical attributes, most of semantic attributes are textually represented information. Since the symbolic regression works in essence with mathematical operators, therefore we need convert those values from the textual representation to numerical values.

[4] `https://developers.google.com/analytics/devguides/collection/gajs/methods/gaJSApiUrchin`

Table 3.9: Example of semantically enriched clickstream - where *vo* is visit order, *po* pageview order, *tsp* time spent on the page in seconds, *co* conversion flag, *p* price in dollars, *tr* transport type, *de* destination and *ac* accommodation and *score* as the result of the weight function.

visitor	url path	vo	po	tsp	co	p	tr	de	ac	score
A	/egypt	1	1	18	0	950	Plane	Egypt	Hotel	18
A	/tunisia	1	2	24	0	400	Plane	Tunisia	Hotel	41
A	/egypt	1	3	13	1	950	Plane	Egypt	Hotel	27
B	/italy	2	1	35	0	400	Bus	Italy	Apartment	35
B	/croatia	2	2	17	0	200	Bus	Croatia	Apartment	29
B	/greece	2	3	41	1	400	Bus	Greece	Apartment	86

We solve this problem defining two categories of semantic variables: ordinal and categorical [134], [135].

In the ordinal category, there are values having a mutual relationship and are easily comparable. They are represented as an integer number. Example of an ordinal is the accommodation. Possible values are $\{tent, bungalow, hotel\}$ and it can lead to values $\{1, 3, 10\}$. Integer values simultaneously represent a level of the accommodation comfort.

Categorical variables are not easily comparable so that we represent them as indicator (binominal) variables [134]. Example is the destination. We transform one variable with possible values $\{Greece, Tunisia, Egypt\}$ in three variables with the binary expression $\{0, 1\}$ of the occurrence.

3.1.4.3 Results

Our experiments summarize results from our former works to utilize symbolic regression in the processing of the usage data [A.15], [A.14].

We experimented with synthetic and real datasets. The real dataset contains clickstreams of real visitors of a travel agency. Structure of this set is described on Table 3.9. The real dataset contains over 160 visits with conversion as an expression of the ground truth, with four pageviews in average per each visit. Total number of pageviews is 650. Because of limited amount of conversions in our real dataset we designed a random clickstream data generator with a probabilistic model on the background. We defined probabilities of visit counts, visit length and similarity with target conversion. Other possibilities are for increasing the time spent for interactions with features that are similar to the conversion. We use this solution for testing and generating larger data sets. However, probabilities can misrepresent real visitor behaviour. Better result could be with generators based on learned real visitor's models [60].

We experimented with the symbolic regression on three types of data sets, two synthetic (5000 visits) and one real. First dataset *syn1* has high probability of the long spent time on the pages that are similar to the conversion. Second dataset *syn2* has no priorities. It is pure random dataset. The last one is the real data set from the travel agency. Unlike

synthetic data sets, the real data set does not always have conversion pageview as the last pageview. Users can generally provide ground truth (conversion, tour reservation, ...) and continue reading the web site. Our setting of the genetic programming algorithm: starting population size = 100 randomly generated individuals, max count of generations = 1000, probability of crossover = 0.6, probability of mutation = 0.1, stop criterion = 1.0 (quality of individual computed by fitness function), count of nodes in syntactical trees has to be in interval from 2 to 30. We use the roulette wheel selection.

Since we used only specific datasets, we limited the evaluation to the manual examination of provided formulas instead of using any qualitative metric. The goal of those experiments is to provide a preview on usage of genetic algorithms in the domain of web analytics. We repeated each experiment 100 times and we also manually minimized the selected equations. The minimization covers steps to convert the equation to the simple and also presentable form, e.g a brackets removal or calculations with constants.

Promote the most similar:

- *syn1* - it stops in average in *17th* generation where individual formulas mainly promote *tsp*. Typical and minimized equation is: $w = 5 * tsp + 36$
- *syn2* - stops in average in *63rd* generation and *po* is mostly promoted. Typical and minimized equation is: $(3 * po + 6) * po + 7$
- *real* - from 100 restarts, 6% promotes *po*. In other cases it never stops at the stopping criterion related to the fitness function. It stops at the max number of generations. Maximal achieved fitness is 0.01 and there was no observed trend in final equations. We identified that this issues is caused by the nature of the dataset: small size, conversion can be generally at any time during the whole visit and data are too heterogeneous.

Promote sum of participations:

- *syn1* - it stops in average in *142nd* generation with results promoting *po*, but in some cases with influence of *tsp*.
- *syn2* - it stops in average in *135th* generation with results promoting *po*. Both equations for *syn1* and *syn2* are similar to *Promote the most similar - syn2*.
- *real* - from 100 restarts, 9% promotes *po*. In other cases reaches the threshold of maximum 1000 generations, best value of fitness is not greater than 0.01 and there was not observed trend in learned equations. The increased percentage of reaching the fitness threshold is due to the similarity of interactions' features. Based on our observation, interactions following the conversion have similar values of features to those associated with the conversion.

As a conclusion to our experiments, we can state that final equations tend to be similar to the empirical one (See Section 3.1.2.2). Our experimental results reflect previous researches, when the most important aspects are interaction order and time the user dedicate to the content item. Promoting of *po* is influenced by the fact that conversions have usually the same features as previous interaction. Users tend to iteratively get to the tour that is close to the tour they finally booked. Simultaneously, symbolic regression generally supports rapidly rising functions. This combination leads to situation when the last pageview before the conversion has maximal weight and sums of other pageview weights does not have major influence on results.

Based on the results, there is also no support for semantic descriptions of tours in final equations. The influence of the interaction order and time outbalance other characteristics. See Discussion section 3.1.6 for more information about proposed solutions.

3.1.5 Implementation

3.1.5.1 Interest Beat - General Analytics Interceptor

In this section we introduce a tool *Interest Beat - General Analytics INterceptor (InBeat - GAIN)*. This tool is used for data acquisition and it also incorporates the method for the aggregation of interactions proposed in this section. It is implemented in Node.js and available as an open source on GitHub [5].

InBeat is generally implemented as a service that exposes the RESTfull API for clients. There are two main resources: data collection and exporting. The API also exposes other services to manage heuristically defined rules and taxonomy that are important for a configuration of the aggregation. The details on the exposed API and several examples are in the on-line documentation.

3.1.5.2 Symbolic regression

We implemented the symbolic regression in Java. Since it is only an experimental implementation and the method also incorporates manual steps, the implementation is not yet publicly available. Listing 3.1 demonstrates a core of the symbolic regression implementation - the genetic algorithm.

```
1  public IChromosome ga() {
2          t = 0;
3          stopCriterion = false;
4          P = randomPopulation(startPopulation);
5          while (t < tmax && !stopCriterion) {
6              t++;
7              Q = new ArrayList<IChromosome>();
8              while (P.size() > 0) {
9                  IChromosome alpha1 = P.remove(rouletteWheel(P));
10                 IChromosome alpha2 = P.remove(rouletteWheel(P));
```

[5] https://github.com/KIZI/InBeat

```
11                  if (Math.random() < pRepro) {
12                      reproduction(alpha1, alpha2);
13                  }
14                  Q.add(alpha1);
15                  Q.add(alpha2);
16              }
17              P = Q;
18              // if (convergence criterion) stopCriterion = true;
19              for (IChromosome i : P) {
20                  if (i.isSatisfiedCriterion()) {
21                      stopCriterion = true;
22                      break;
23                  }
24              }
25          }
26          // find out best chromozome from final population
27          double max = P.get(0).fitness();
28          int maxi = 0;
29          for (int i = 0; i < P.size(); i++) {
30              IChromosome ch = P.get(i);
31              if (ch.fitness() > max) {
32                  max = ch.fitness();
33                  maxi = i;
34              }
35          }
36          // return best
37          return P.get(maxi);
38      }
```

Listing 3.1: Genetic Algorithm - Core of Symbolic regression

3.1.6 Discussion

Association Rule Mining We performed preliminary experiments with supervised classifiers to improve the quality of results. Since rules are one of the most easily understandable machine learning model, we also incorporated the rule learning algorithm to the evaluation. The advantage of such rule based output is that there is a possibility to launch rule learning and than use rules as a replacement of heuristically defined rules.

Fitness Functions Two fitness functions were defined heuristically based on previous researches. One is focused on promoting only one interaction prior to the ground truth interaction. The second one promotes all prior interaction, that are similar to the ground truth. Other functions are possible, especially the function incorporating penalizations for specific attributes and promoting semantic attributes.

Low Amount of Conversions We suggest a possible solution to the problem of small amount of conversions in real data set. Proposed solution is a co-training. It allows label

data, which are not labelled [136]. It is semi-supervised learning method that accommodates multi-modal views of data. First we state which semantic dimensions correlate with each other. Then an intermediate product during learning weight function is used to label all unlabelled data. Finally, the surely determined instances in semantic dimension X are labelled and added to dimension X and to other correlated dimensions. Instances are added during learning. More training data are available and it can help to avoid equations, which prefer the page/interaction order.

3.1.7 Summary

Although there is an abundance of proprietary approaches and algorithms for the data acquisition in the Web Usage Mining, they are domain specific and outputs are typically unsuitable for direct processing by mainstream machine learning algorithms and tools. One important reason is that interactions performed by individual users tend to be of irregular length. Modern rich interfaces or interactive web applications also allow to perform multiple interactions per one content item. According to our survey of related works, there is a lack of existing approaches in the Web Usage Mining preprocessing focused on the aggregation.

Our rule-based aggregation method is based on hand-written rules and is focused rather on simplicity than to act as a complex solution. The method is designed so that the rules are defined by humans and it can thus lead to error-prone solutions. We have also considered an automatic learning of such rules to overcome the requirement for a domain specialist. Our proof-of-concept solution and evaluations proves that the method is applicable in several domains. The second method we experimented with is a genetic algorithm to assign weights for specific interactions. Unlike the previous approach, the genetic algorithm requires labelled data for the learning.

Presented methods for the aggregation of multiple user interactions into one unified relation between a user and a content item addresses consequences of modern interfaces. It provides unified outputs that are processable by conventional algorithms and tools.

3.2 Semantization and Propagation

In this section we focus on the second step of our methodology: Semantization and transformation. We present our approach how to perform a semantization of content items - to provide a set of annotations that link content items to a knowledge base. Our approach is focused on a type of problems that uses the Web of data to augment the feature set. Original data are automatically mapped to the Linked Open Data (LOD) identifiers, and then additional features are extracted from public knowledge bases such as the DBpedia. It allows to build a semantic representation of the specific content item. The huge amount of achievable additional features can provide valuable information for various applications. We also present our aggregation based on an ontology propagation approach. It is suitable for a situation when the content item contains multiple annotations and we would like to represent it in a unified manner.

This section presents two approaches related to the semantization a transformation. Both are considered as a prerequisite for subsequent steps of our methodology:

- Semantization of domain specific content items that links each content item to a knowledge base. The method is based on predefined SPARQL queries to the DBPedia. We call the method *URI Alignment* and was evaluated on a movie ratings dataset [A.3].

- Ontology propagation as a transformation of multiple annotations to a unified representation. The proposed propagation was not independently evaluated. However, the propagation is incorporated in our tool called InBeat that was proposed and evaluated mainly in [A.10, A.11, A.12] and contributions to projects' reports and deliverables [5].

3.2.1 URI Alignment

This section presents an approach how to map an existing domain specific movie ratings dataset *MovieTweetings* [14] to the DBpedia. The dataset is publicly available and presents information about relations between users and content items: users rating movies. It is constructed from publicly available information - well-formatted tweets on Twitter and contains movie ratings extracted from Twitter for movies released from 1900s to the presence. Because the dataset is based on extraction of ratings from Twitter users around the world and it is daily updated, we have to deal with the following issues: multilingualism in movie titles, freshness (daily updates), inaccuracies and incompleteness of data.

The presented approach is focused on ad-hoc SPARQL queries instead of "guessing" *URIs* [15] or downloading all possible data to a local database and processing the data locally [16],[17]. According to our survey of related works, there is no existing mapping dataset of movies for *MovieTweetings* to the LOD. The goal is to provide a one-to-one mapping of movies from *MovieTweetings* dataset to Linked Open Data cloud as *URI* identifiers. We also designed a set of confidence values that express the relevance of URI alignment.

3.2.1.1 Definitions

Movie Dataset. *A dataset where each item (movie) is represented by a set of features.* Each movie is represented by a set of following features: title (F_{title}), release date (F_{date}) and a set of assigned genres (F_{genres}). Example: *Rocky (1976), Drama | Sport.*

URI Alignment. *A process of mapping a source item represented by various features to URI from a knowledge-base.* The source item $I_s \in I$ is described by a set of features $I_s = \{F_1, \dots\}$ that exclusively define the source. For each knowledge-base kb is the alignment defined as:

$$URI_s = alignment_{kb}(\{F_1, \dots\}) \tag{3.9}$$

Augmenting a Feature Set. *An extraction step to provide more features from a knowledge-base.* Let the I_s is a source item defined by a set of n features $\{F_1, \dots, F_n\}$. The augmenting extends the set of features using a URI identifier from a knowledge-base. After completion of the augmenting step, the size of features is $n + a$. Where a is a number of additional features extracted from a knowledge-base.

3.2.1.2 Partial URI Alignments

We compose the URI alignment process from a sequence of partial alignment methods in order to address several situations. Each partial alignment is designed to provide a URI identifier only when conditions of a specific situation are satisfied. However, the overall process as a sequence of those alignments should handle almost all situations that are specific for movie ratings dataset. Our proposed approach is designed to query the DBpedia using a set of predefined SPARQL queries.

Perfect match of a title

This partial alignment addresses the situation when all features match perfectly with features in a knowledge base. Therefore, there is no requirement for any additional tunings and alignments. The perfect matching uses the title and release date to find the desired identifier in the DBpedia. Main conventions for naming of resources in the DBpedia drive following three patterns that we use for perfect matching: 1) One-to-one match of titles "F_{title}" while the release date is part of a linked category: "$F_{year}\ film$"; 2) A title with suffix indicating a movie "$F_{title}\ (movie)$" and the release date in linked category: "$F_{year}\ film$"; 3) A pattern that contains the title and date in a resource title: "$F_{title}\ (F_{date}\ film)$".

$$URI = perfectMatch(F_{title}, F_{date}) \tag{3.10}$$

Listing 3.2 presents a template for the SPARQL query to perform the perfect matching of the title and year according to the existing conventions for titles of movies in the DBpedia (Example: *Rocky, Rocky (film) and Rocky (1976 film)*).

```
SELECT DISTINCT ?movie ?title ?category WHERE {
?movie rdf:type dbpedia-owl:Film ;
rdfs:label ?title .
?movie dcterms:subject ?category .
?category rdfs:label ?year .
FILTER (
  (
    (str(?title)="%s" || str(?title)="%s (film)")
    &&
    regex(?year,"^%s film", "i")
  )
  ||
  str(?title)="%s (%s film)"
  )
}
ORDER BY ASC(?movie)
```

Listing 3.2: SPARQL query - Perfect match of the title and year

Partial match of a title

For situations that are not handled by the perfect matching we use a partial matching. It queries the knowledge base without any constraints on surrounding textual fragments. The availability of any observation is thus marginal. The goal is to match titles or dates that are malformed in terms of missing parts:

$$URI = partialMatch(F_{title}, F_{date}) \tag{3.11}$$

Listing 3.3 describes a modification of the FILTER condition as a relaxation of patterns for the title and year.

```
...
FILTER regex(?title,"%s", "i") .
FILTER regex(?year,"%s", "i")
...
```

Listing 3.3: SPARQL query - Partial match of the title and year

Pattern-based match of an abstract

There are situations, when the title of a movie is not present in the same format as in a knowledge base. However, the title and year is named in an associated abstract that briefly introduces the movie. Those situations we address using two main patterns inspired by the nature of the DBpedia abstracts formatting: "... F_{title} *is a* F_{date} *film* ..." and "... F_{title} ... *released* F_{date} ...". Example: *Rocky is a 1976 film ...* or *... Rocky ... released 1976 ....*

$$URI = patternMatch(F_{title}, F_{date}) \tag{3.12}$$

Listing 3.4 describes a modification of the FILTER condition.

```
...
FILTER (
  regex(?abstract,"^%s is a %s", "i")
  ||
  regex(?abstract,"^%s .* releas.* %s", "i")
)
...
```

Listing 3.4: SPARQL query - Pattern-based match of the abstract

Any match of an abstract

The last alignment is designed to deal with most remaining situations that might appear. We use a concept of searching for any existence of the title in an abstract. This partial alignment is the least plausible method since it can also provide irrelevant results. e.g. in situations when the abstract only cite the movie title as a previous work of the same director etc. However, this method is able to reveal movies with titles in foreign languages that are also occasionally named in an abstract. Example: "*... also known as* F_{title} *...*" or "*... (Italian:* F_{title} *..., German: ...)*".

$$URI = anyMatch(F_{title}, F_{date}) \tag{3.13}$$

Listing 3.5 shows a modification of the FILTER condition.

```
...
FILTER regex(?abstract,"%s", "i").
FILTER regex(?year,"%s", "i")
...
```

Listing 3.5: SPARQL query - Any match of the abstract

3.2.1.3 Confidence Values

In order to express a basic relevance of the proposed mapping to *URI* identifiers from the *DBpedia*, we provide a set of confidence values. Those values are available as a component of final mappings and can be used together with a method name for filtering of results. The setting of the filtering is left to the end user of the mapping dataset. To compute the confidence we use following formulas:
Title confidence(titleCofidence) is computed using the Levenshtein distance metric of the title and label from the DBpedia. The disadvantage of this metric is the sensitivity on titles in foreign languages. It may happen that the two strings from different languages are compared.

$$titleConfidence(F_{title}, DBpedia_{title}) = 1 - \frac{levenshtein(F_{title}, DBpedia_{title})}{max(length(F_{title}), length(DBpedia_{title}))} \tag{3.14}$$

Year Confidence (yearConfidence) is computed as a pure distance of years. The year confidence make sense if we consider a movie with another release date as a possible candidate. Based on the algorithm that we use, we also take into account movies from the previous and next year (See Section 3.2.1.4 for more details). We address typos that may appear in features of the source item. We propose to compute the year confidence as:

$$yearConfidence(F_{date}, DBpedia_{date}) = 1 - \frac{|F_{date} - DBpedia_{date}|}{F_{date}} \tag{3.15}$$

Genre Confidence (genreConfidence) uses number of common genres as an underlying concept. Since source item genres and categories from the DBpedia are created by independent providers, we evaluate the presence of the genre as a case-insensitive partial substring match. It covers most of situations. The final value is an expression of how many genres match the DBpedia categories. The following formula demonstrates the approach we use:

$$genreConfidence(F_{genres}, DBpedia_{categories}) = \frac{\sum_{g \in F_{genres}} (is\ g \text{ in } DBpedia_{categories})?1 : 0}{count(F_{genres})} \tag{3.16}$$

3.2.1.4 Algorithm/Approach

The Algorithm 2 demonstrates steps we apply to align the source item represented by title, release date and genres (F_{title}, F_{date} and F_{genres}). First part aligns the source item to a URI, where the F_{title} is fixed and the F_{date} varies over the original year, previous and next year. It enables to overcome typos or incorrect facts that can be available either in the source item or in the knowledge base. Second part extracts required information from the knowledge base using the URI identifier and computes confidence values. As a results of the algorithm we provide the mapping of the item to a knowledge base that can be used to build a semantic representation of the item.

3.2.1.5 Augmenting a Feature Set

Since we have a *URI* as an identifier of a movie in the DBpedia, we use LOD cloud to get relevant information to augment the feature set. The *URI* as the identifier of data related to the associated movie can be used to extract additional features. Essentially, we use dereferencing of an *URI* identifier to collect all information about a movie. Basic SPARQL query can be used to extract a subset of those properties too.

We put together all extracted data from the DBpedia and all available data from the original dataset. As a result we have a structure that consists of relations between users and movies (as a rating relation) and additional features to describe movies. The output is a semantic representation of items connected with users. As the advantage of additional features we consider the possibility to explore relations between movies. Especially valuable can be relations that are not obvious from the limited set of original non-semantic features.

Algorithm 2: URI Alignment

input : A movie as a source item I_S with a set features F_{title}, F_{date} and F_{genres}.
output: A URI identifier of source item, confidence values tc, yc, gc.

```
1  begin
2      URI = null
3      // try original, previous and next year
4      for F_date ∈ {F_date, F_date − 1, F_date + 1} do
5          URI = perfectMatch(F_title, F_date)
6              || partialMatch(F_title, F_date)
7              || patternMatch(F_title, F_date)
8              || anyMatch(F_title, F_date)
9          // stop if any URI found
10         if URI! = null then
11             break
12     // compute confidences
13     DBpedia_title, DBpedia_date, DBpedia_categories = extractFromDBpedia(URI)
14     tc = titleConfidence(F_title, DBpedia_title)
15     yc = yearConfidence(F_date, DBpedia_date)
16     gc = genreConfidence(F_genres, DBpedia_categories)
17     // return results
18     return URI, tc, yc, gc
```

Example 3.2.1. ***Benefits of an Augmented dataset***

Given two movies m_1 : Batman (released 1966) and m_2 : Rocky (released 1976) with assigned genres and two different users (u_1, u_2) that interacted (e.g. rated) with those movies, we cannot infer any relation between those movies. They have nothing in common. The left part of Figure 3.9 demonstrates such situation. Considering the URIs for both movies as a result of the alignment, we can extract additional genres/categories and relations e.g. starring persons. The right part of Figure 3.9 demonstrates that excerpt of the possible augmentation. New genre/category "American films" is joined and it is common for both movies. The starring person "Burgess Meredith" is the common element as well. The augmentation can provide more relations useful for building extended and well connected rich representations.

3.2.1.6 Experiments with URI Alignment

For experiments with our URI alignment we selected a movie ratings dataset: The MovieTweetings [14]. It is constructed from publicly available information - well-formatted tweets on Twitter. Those tweets are published using applications dedicated to express movie ratings by users. First entry is from February 2013.

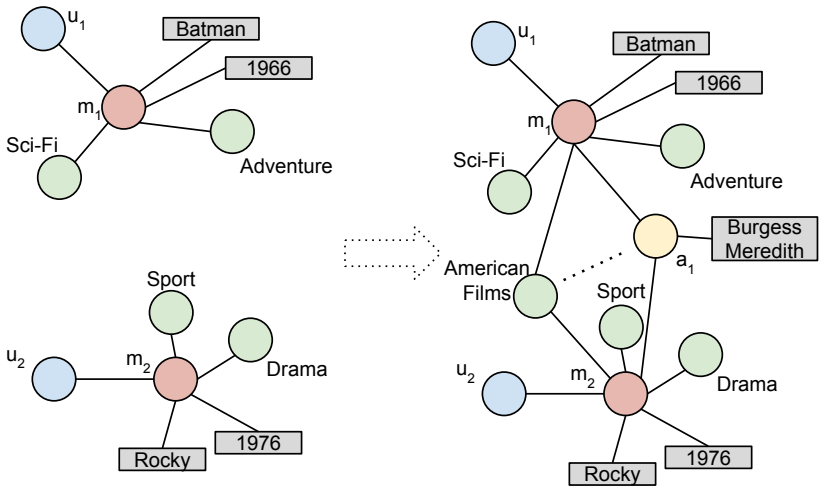


Figure 3.9: Example of the augmented dataset for two movies *Rocky (1976)* and *Batman (1966)*.

The dataset provides information that can be denoted as a set of triples (User, Rating, Movie). *User* is identified by its Twitter identifier. *Movie* is represented by a title, release year and a set of assigned genres. The *Rating* links both of them and represents a relation described by a timestamp and a rating level scaled from 0 to 10. The dataset contains movie ratings extracted from Twitter for movies released from 1900s to the presence.

The main advantage, compared to other existing datasets (MovieLens [13], Last.fm [137], Jester [138] or Book-Crossing [139]), is an availability of updates on a daily basis. Because the dataset is based on extraction of ratings from Twitter users around the world and it is daily updated, we have to deal with the following issues: multilingualism in movie titles, freshness (daily updates), inaccuracies and incompleteness of data. The goal of our experiments is to provide the mapping of movies to the DBpedia using our URI alignment method.

Results and Statistics

In this section we briefly describe results of the mapping. We use a snapshot of the dataset downloaded on June 1, 2015. It contains over 21000 movies. At the time of publishing of this dissertation thesis, the mapping provides URIs for 71.3% movies. The remaining movies were not mapped due to the issues mentioned at the end of previous section.

Figure 3.10 depicts distribution of years for movies that were not successfully mapped to any *URI*. There is a large amount of movies from recent time that were not successfully mapped due to their unavailability in the DBpedia. The reason is that the version of the DBpedia we used was published at the end of 2015 (based on Wikipedia dumps from February/March 2015)[6] and most recent movies are not available in this version. Figure

[6] `http://wiki.dbpedia.org/Downloads2015-04`

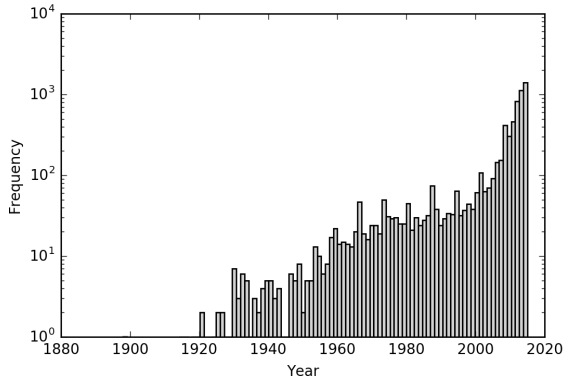


Figure 3.10: Distribution of years for unmapped movies

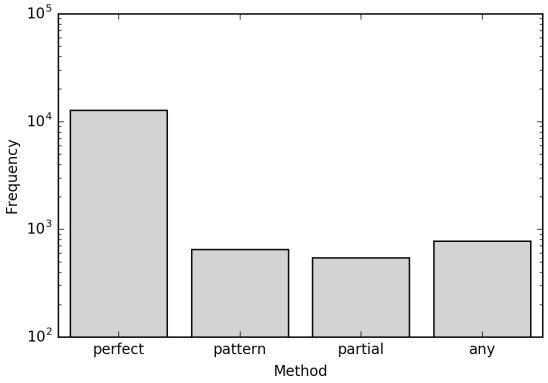


Figure 3.11: Overview of methods used for successful mapping

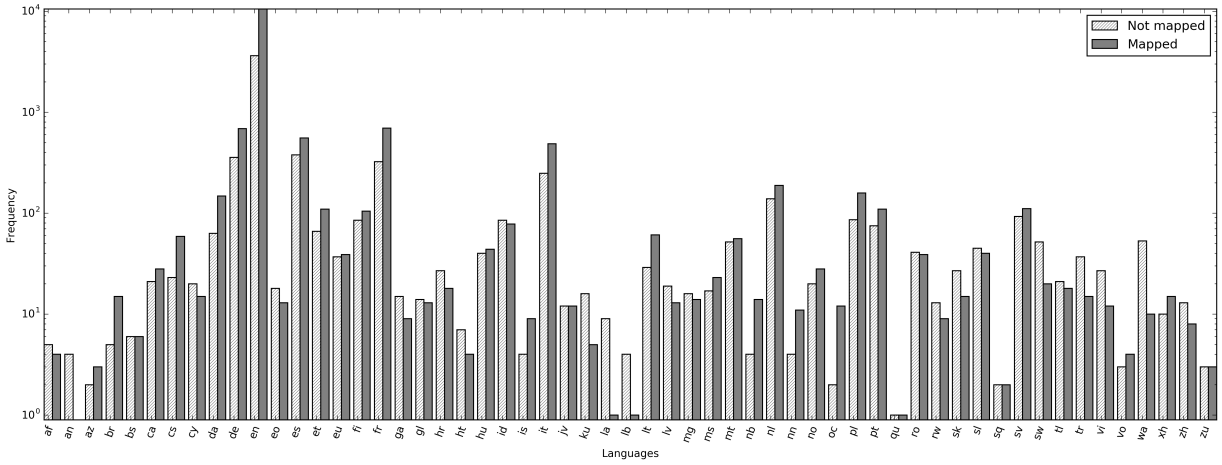


Figure 3.12: Distribution of mapped/unmapped movies with respect to languages detected in movie titles

3.11 demonstrates the usage of methods for successful mapping of the movies. The method that performs the perfect match of a title and a year is the most frequent (perfect: 86.61%, pattern: 4.38%, partial: 3.66%, any: 5.35%). Figure 3.12 provides an overview of the language distribution in titles.[7] This summary presents the availability of mappings to the *DBpedia* for various languages.

We also evaluated our approach using another existing mapping dataset for MovieLens [16]. We selected this dataset because both original datasets (MovieLens and MovieTweetings) are provided in the same format and authors of the mapping dataset for MovieLens deal with the same task: mapping of movies to the DBpedia. Furthermore, the dataset was

[7]Languages detected in titles using *LangID*: `https://github.com/saffsd/langid.py`

manually corrected, therefore we can use it as a ground truth. We launched the proposed mapping algorithm and compared to available mappings. Our approach achieved over 98.5% match, where the incorrectly mapped values were either missing *URIs* or incorrect links that can be filtered using the confidence values.

3.2.1.7 Implementation

The URI Alignment method for the MovieTweetings movies is implemented in Python and is available as an open source on GitHub [8]. Available script automatically downloads the latest existing movie dataset, merges with the already existing mappings of movies to the DBpedia and complete the mapping for new movies. The implementation also regularly renews the fraction of the oldest records to react on updates in the knowledge base.

The latest mapping for movie dataset is also available for download as a part of the repository on GitHub. It is formated as a CSV file where the records are separated by a semicolon: 1) id - id of the movie in MovieTweetings; 2) title - title (year); 3) genre - assigned genres; 4) uri - DBpedia URI; 5) method - the partial alignment method of the mapping (perfect | partial | pattern | any); 6) tc - title confidence; 7) yc - year confidence; 8) gc - genre confidence; 9) updated - datetime of the last update.

Example 3.2.2. ***Output dataset format*** *Table 3.10 demonstrates a simple example for the movie Rocky released in the year 1976, where both titles matches as well as release dates. Both genres are also present in associated categories in the DBpedia.*

Table 3.10: Example of one entry in the final mapping dataset for movie *Rocky (1976)*.

id	title	genre	uri	method	tc	yc	gc	updated
75148	Rocky (1976)	Drama|Sport	http://dbpedia.org/resource/Rocky	perfect	1.0	1.0	1.0	2015-06-01T11:33:06

3.2.2 Other Approaches for Semantization

In this section we briefly describe alternative approaches that we also use on the background of our research, especially for evaluations. The underlying concepts or research methods are not part of our research. We would like to kindly refer the readers to relevant publications.

Recognizing Named Entities

The required prerequisite of the approach we describe in the previous section is availability of specific features (namely title, release date and connected genres for each movie). Nevertheless, certain group of items may not be specified using well structured features. They can be represented only as a free text e.g. articles on web pages, short descriptions of items etc. We use *Named Entity Recognition (NER)* tools as an underlying concept. Their

[8] `https://github.com/jaroslav-kuchar/MovieTweetingsMappings`

responsibility is to provide for each textual fragment a set of entities appeared in the text. All entities are also connected using URI identifier to knowledge base such the DBpedia. We use this approach to semantize textual documents and articles (See Section 3.5).

Example 3.2.3. ***Entity recognition***

Given the following textual fragment of the DBpedia abstract for Rocky movie: "Rocky is a 1976 American sports drama film directed by John G. Avildsen and both written by and starring Sylvester Stallone.", three entities were recognized using Entityclassifier [12]. All entities are identified by URI to the DBpedia that can be used to extract additional features in the same way as was described in the previous section.

Wrapping and Transforming Structured Data

The structured data can be semantized as well. The common approach is to provide so called wrapper that usually provides the same data as in original source but in a semantic way. Optionally, certain transformations are also required to provide data in the RDF representation. We use wrappers and transformation to provide the semantic representation of publicly available data collected from web sites and public directories. We use this approach to semantize information about existing Web APIs (See Section 3.3 and Section 3.4).

Example 3.2.4. ***Wrapping and Transformation***

The ProgrammableWeb[9] *directory is 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. The basic information that can be found in the directory is that two developers know each other. FOAF*[10] *ontology (prefix foaf), concept foaf:Person that describes users and property foaf:knows that describes a social relationship between users might be used to express the structured information in the semantic way.*

3.2.3 Semantic Propagation and Aggregation

In our research we are focused on objects that are semantically annotated as a result of the semantization approaches described in the previous sections. The semantic information may come from entities (individuals or types/classes) assigned to the particular object. Supposing that the particular object can be generally connected to many entities representing persons, places etc. and each entity can be assigned to many ontology classes, there is a need to aggregate those multiple assignments to a single semantic representation of the object. The single rich representation allows to build a unified data format so that it can be further processed by regular algorithms (e.g. machine-learning algorithms).

[9] http://www.programmableweb.com/
[10] http://xmlns.com/foaf/spec/

3.2.3.1 Definitions

Semantically annotated object. *An object that is labelled with semantic information.* An object O_i is linked to any semantic unit using their *URI* identifiers as object attributes ($OA_i = \{E_I, \dots\}$). The semantic unit can be either a class or an entity from a knowledge base.

Taxonomy of classes. *A hierarchical structure interpreting the relations between classes in terms of subclasses.* The taxonomy T is consisted from a set of partial items $T_i \in T$, where each T_i defines its relation to another taxonomy item using subclass relations. For the DBpedia the root is defined as the *Thing*, others are connected as the subclasses, meaning the categorization of general classes to subclasses (e.g. Thing - Organization - Sports Team - Soccer Club - ...).

3.2.3.2 Approach

The goal of the semantic propagation and aggregation is to transform multiple semantic annotations $E_i \in E$ linked to the object O_i ($OA_i = \{E_I, \dots\}$) to one aggregated unit E_{merged} as a semantic representation of the object. Each semantic annotation may have multiple connections to the taxonomy on various levels of the hierarchy. We take into account all of those connections using a propagation and the subsequent aggregation. It allows to preserve the overall balancing of the classes participations. The proposed propagation is also able to work with confidence values for each class assignment representing similarity of the entity to the class ($C_{E_i} = \{(T_i, similarity\_value), \dots\}$). Those values can be generated by the publisher of the content or by tools to automatically detect and recognize the entities (e.g. Named Entity Recognition tools).

The proposed algorithm for the semantic propagation and aggregation we present in Algorithm 3. The first part of the algorithm propagates values (similarity values) within the taxonomy from leaf nodes to the root. Where the weighted sum of similarity_values for the current node and all descendants is computed as a weighted score reflecting the number of descendants. Secondly, the class assignments are merged and the sum of the same classes' values is computed. Finally, the proposed algorithm normalizes the values using the maximum value in the final merged unit.

Example 3.2.5. *Let consider an object O_e "news item about politics" that is annotated using three distinct entities: White House, U.S.Government and London. Each entity is associated to classes from the DBpedia ontology with a certain confidence level provided by the Named Entity Recognition tool:*

(E_1) White House=Corporation(0.4), Organization(0.8), Government Building(0.3), Civic Structure(0.1) and Place(0.1)

(E_2) U.S.Government=Corporation(0.3), Organization(0.8)

(E_3) London=Place(1)

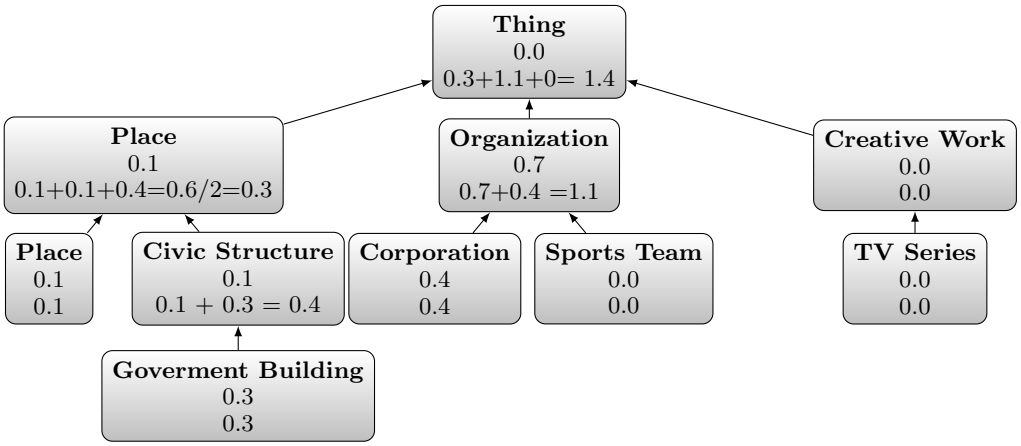


Figure 3.13: Semantic representation of the entity *White House* (E_1).

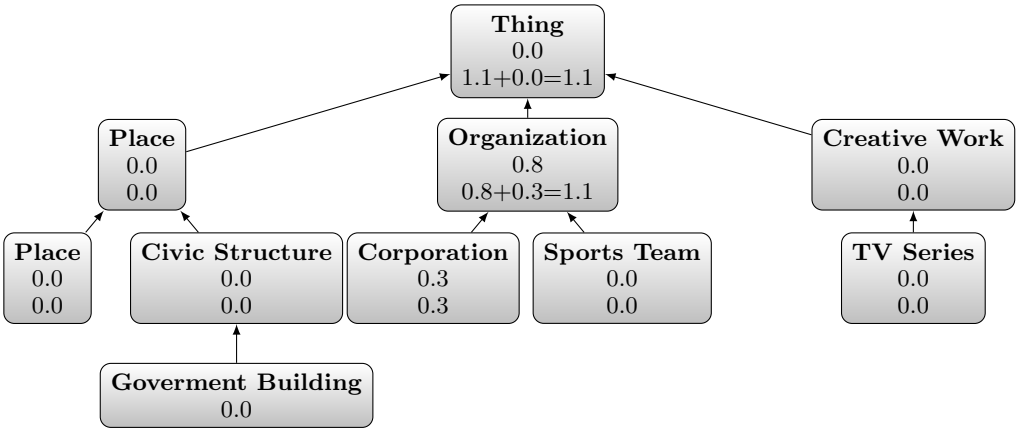


Figure 3.14: Semantic representation of the entity *U.S.Government* (E_2).

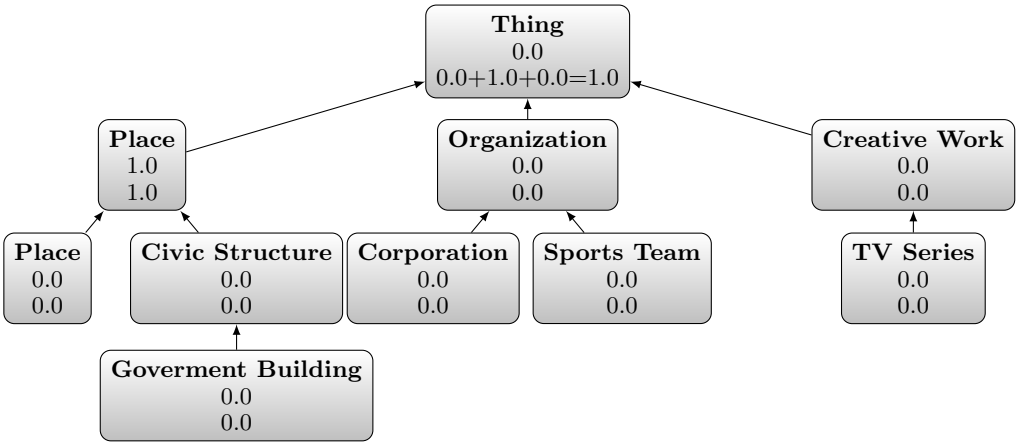


Figure 3.15: Semantic representation of the entity *London* (E_3).

Algorithm 3: Semantic Propagation and Aggregation

input : Taxonomy/tree of ontology classes $T = \{\dots\}$
A set of semantic units $E = \{E_1, \dots, E_n\}$ assigned to the object O_i.
Each unit is labelled by classes $C_{E_i} = \{(T_i, similarity\_value), \dots\}$
where $T_i \in T$ and $similarity\_value \in [0; 1]$

output: Entity E_{merged} with aggregated classes $\{(T_i, similarity\_value), \dots\}$

```
1  begin
2      // propagation within taxonomy
3      for E_i ∈ E do
4          // upwards propagation
5          for C_j ∈ from leafs to root of E_i do
6              C_j[similarity_value] = Σ C_j[similarity_value] and all descendants
7              if # descendants > 1 then
8                  C_j[similarity_value] = C_j[similarity_value] / # descendants
9      // merging class assignments and similarity values of units
10     E_merged = ∅
11     for E_i ∈ E do
12         E_merged = E_merged ∪ E_i
13         for C_j ∈ C_{E_i} do
14             // sum up similarity values for the same classes
15             C_j^merged = C_j^merged + C_j
16     // find max similarity value
17     maxValue = findMax(C_{E_merged})
18     // normalize
19     for C_j ∈ C_{E_merged} do
20         C_j[similarity_value] = C_j[similarity_value]/maxValue
```

Figures 3.13, 3.14 and 3.15 present an excerpt of the DBpedia ontology for E_1,E_2 and E_3. The first line denotes the similarity values assigned to classes. The second line presents the propagation step for each entity. Figure 3.16 demonstrates merging of entities and final normalization, where the first line is the merging and the second line is the normalization.

The final results (Red values on Figure 3.16) are thus the overall semantic representation of the "news item about politics" object O_e. Since in this example we use the taxonomy from the DBpedia, all processed objects will have the same dimension of the final representation that corresponds to the size of the DBpedia taxonomy.

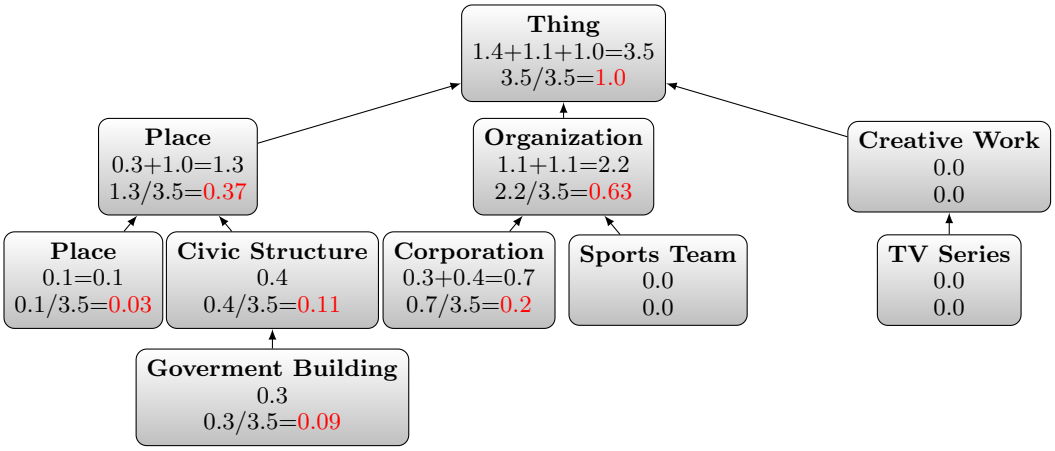


Figure 3.16: Results of semantic propagation and aggregation: merge and normalization of entities E_1, E_2 and E_3.

3.2.3.3 Experiments and Implementation

The experiments with semantic propagation were mainly performed together with the evaluation study for the tool Interest Beat - General Analytics Interceptor described in Section 3.1.3. InBeat uses the propagation and aggregation as an underlying concept to provide a unified semantic representation of each object users are interacting with. The outputs of semantic propagation was succesfully evaluated by partners within experiments and trials of the LinkedTV project [5]. Aggregated outputs were consumed by another services designed by co-workers within the project. Those service were designed to create semantic representations of content items together with user profiles based on logical constructs. Since it is out of the scope of this thesis and the author of this thesis does not participate on the processing of aggregation and propagation, we would like to refer the reader to relevant publications [5]. Since the propagation is incorporated in the InBeat tool, we use the outputs in our subsequent contributions.

Extracting Ontology Taxonomies

We designed a simplified taxonomy structure in JSON for internal propagation purposes and also for the visualization. The tool to convert any ontology in format N3 to the simplified JSON format is implemented for Node.js and is available as an open source on GitHub [11]. Example of the format is demonstrated on Listing 3.6. The *uri* attribute is an identifier of a class, *name* stands for a textual title of a class, *value* is similarity important for propagation and *children* points out to subclasses.

```
1 {
2   name: "Thing",
3   uri: "http://schema.org#Thing",
4   value: 0.5
```

[11] https://github.com/jaroslav-kuchar/n3-2-d3

```
  children: [
    {
      name: "CreativeWork",
      uri: "http://schema.org/CreativeWork",
      value: 0.1
      children: [ ... ]
    },
    ...
  ]
}
```

Listing 3.6: JSON simplified format of taxonomy

The tool is designed to be used from the command line. Listing 3.7 demonstrates usage on two examples.

```
# Conversion of NERD Ontology
node n3-2-d3.js --n3 http://nerd.eurecom.fr/ontology/nerd-v0.5.n3 --prefix
    http://nerd.eurecom.fr/ontology > nerd.json
# Conversion of schema.org
node n3-2-d3.js --n3 http://schema.rdfs.org/all.nt --prefix http://schema.
    org > schema.json
```

Listing 3.7: Usage of conversion tool

The script also supports simple visualization. Listing 3.8 demonstrates how to start a server and show visualization. The visualization also considers values that are meaningful for propagation.

```
# start server
node server.js
# open nerd file
/index.html?file=nerd.json
# or schema file
/index.html?file=schema.json
```

Listing 3.8: Usage of conversion tool

3.2.4 Discussion

Robustness of URI Alignment. We experimented on a dataset about movies and mapping of them to the DBpedia. Each movie is identified by the set of specific features (title, release date and assigned genres). The approach we use is limited only for the movie dataset. However, the method can be easily adapted to another types of data. Specific SPARQL queries needs to be designed and implemented.

Multilingualism. The right detection of titles in various languages is crucial for the URI Alignment method. We assume the presence of titles in the DBpedia either as a part of a title or a fragment of an abstract. Confidence values (especially Title Confidence) is then computed using the corresponding title from the DBpedia if available.

Augmenting using Specific Features. We do not specify a set of features than can be used for the augmenting. It is not part of our research to define the most suitable set of features. It is domain and application specific procedure. There are many existing works that focus on a selection of properties from knowledge bases that are the most important for representation of the specific types of content. We leave the selection on potential consumers.

Propagation. Although there exist other possibilities to propagate the semantics through the taxonomy, we use the basic bottom-up approach reflecting number of contributing descendants. The main purpose of our approach is to provide generalized classes. Those generalization can help to overcome issues with overspecialised descriptions.

3.2.5 Summary

In this section we focused on the semantization of content items, where the goal is to provide a semantic representation of each content item. The proposed method is based on ad-hoc queries to find *URI* identifiers to a knowledge base, extract additional features and further aggregate all available information. We call the proposed linking method URI alignment. It is designed and evaluated as an experimental domain specific method for linking movies to the DBpedia. Provided connections in form of URI references allow the extraction of additional features and to properly represent content items. Since multiple links to a knowledge base can be provided, we also propose an aggregation step to overcome issues with conflicts or overlaps of multiple features. The output of the overall semantization is a semantic representation of each content item.

3.3 Graph-based Data Enhancement

There are several approaches to enhance graph-based datasets - semantic representations, including methods that use only information within one dataset or methods that incorporates the external knowledge. As the enhancement we consider the management of links: updates, removals or insertions of links. In our proposed approach we are focused on inserting: a prediction of links within one dataset. It considers the information within one dataset and it is primarily focused on dataset with multiple type of links such as RDF datasets. Moreover, the method is designed to consider the temporal information about links as a key concept of the link prediction. We call the proposed method: *Time-Aware Link Prediction* [A.4], [A.1]. We evaluate the proposed approach on real world datasets: an RDF representation of the ProgrammableWeb directory and a subset of the DBpedia focused on movies. The results show that the proposed method outperforms other link prediction approaches.

3.3.1 Method

3.3.1.1 Definitions

Tensor. *A multi-dimensional array of numerical values* [21]. 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* [140]; 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

$$\mathcal{A}(t_0) = \mathcal{A}(t_x) * e^{-\lambda t}; t_0 > t_x, t = t_0 - t_x \tag{3.17}$$

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 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 [141, 142]. In our research we use a single *starting time point* t_x which defines an existence of the link, i.e. the link exists since t_x (see Section 3.3.4 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).

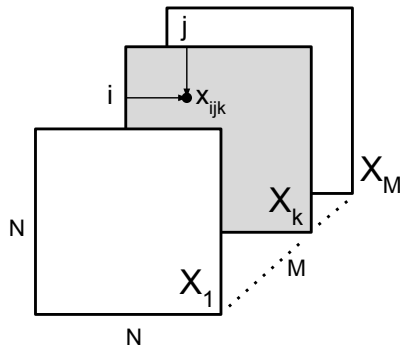


Figure 3.17: Visualisation of a tensor model $N \times N \times M$ with element x_{ijk}.

3.3.1.2 Tensor-based Model with Temporal Information

Simple graph structures can be modelled as matrices, which are 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 [6]. 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 (See Figure 3.17). 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 our research, 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 3.3.4 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 (3.17) as follows

$$x_{ijk} = y_{ijk} * e^{-\lambda t} \tag{3.18}$$

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 3.3.1.1 for additional details about the ageing function).

Example 3.3.1. ***Example of modelling data***

Consider an RDF dataset consisting of four instances of concepts ls:Mashup ($m1, m2, m3, m4$) and wl:Service ($s1, s2, s3, s4$), and three links ls:usedAPI that indicate usages of 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

Table 3.11: 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

arrow indicates the age of the link in 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 3.11 shows this information modelled as a tensor both with and without ageing (in this example we set the 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.

3.3.1.3 Learning Hidden Latent Factors

We use a tensor factorization technique to perform a structural analysis of an RDF dataset. We propose an extension of the RESCAL approach [6] which uses the time information. Each incidence matrix $\mathbf{X}_k$ of a tensor is factorized as

$$\mathbf{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^T, k = 0...M \tag{3.19}$$

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 3.18). 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

$$\min_{\widehat{\mathbf{X}_k}} \parallel \mathbf{X}_k - \widehat{\mathbf{X}_k} \parallel_F \text{ , where } \widehat{\mathbf{X}_k} = \mathbf{A}\mathbf{R}_k\mathbf{A}^T \tag{3.20}$$

The minimum optimization problem can be solved by any non-linear optimization algorithm (e.g. Stochastic Gradient Descent). Similarly to the original approach and implementation [109] we use an alternating least squares (ALS). ALS is more efficient and can be easily parallelized. ALS algorithm does not solves the problem all at once. However, only one parameter is optimized while the others are fixed. This is repeatedly altered until convergence. We use the same minimum optimization problem definition including the same loss function:

$$\min_{\mathbf{A},\mathcal{R}} f_{loss}(\mathbf{A}, \mathcal{R}) + f_{reg}(\mathbf{A}, \mathcal{R}) \tag{3.21}$$

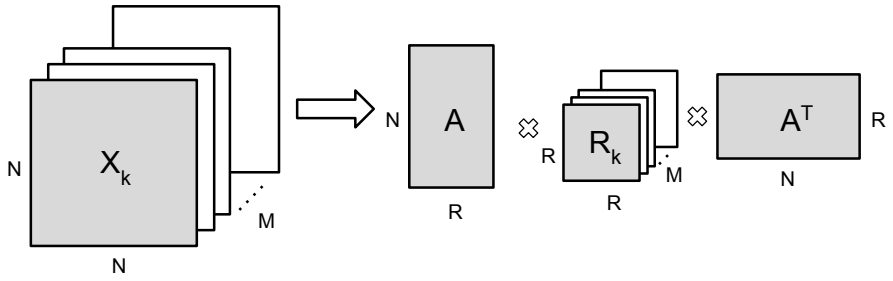


Figure 3.18: Visualization of RESCAL [6].

, where $f_{loss}(\mathbf{A}, \mathcal{R})$ is the loss function defined as:

$$f_{loss}(\mathbf{A}, \mathcal{R}) = \frac{1}{2}\left(\sum_{k=1}^{M} \parallel \mathbf{A}\mathbf{R}_k\mathbf{A}^T \parallel_F^2 \right) \tag{3.22}$$

and f_{reg} is the regularization parameter to prevent overfitting:

$$f_{reg}(\mathbf{A}, \mathcal{R}) = \delta \parallel \mathbf{A} \parallel_F^2 + \delta \sum_{k=1}^{M} \parallel \mathbf{R_k} \parallel_F^2 \tag{3.23}$$

The ALS alters optimization of $\mathbf{A}$ and $\mathbf{R_k}$. One matrix is optimized while the second is fixed. δ denotes the regularization parameter. The optimization algorithm is stopped when the stop criterion is met: either maximum number of iterations or the minimum change between two iterations is lower than predefined threshold. For initialization of $\mathbf{A}$ and $\mathcal{R}$ we use random numbers. Algorithm 4 demonstrates our ALS algorithm that we use in our link prediction method.

Optimization of $\mathbf{A}$:

$$\mathbf{A} = \left[\sum_{k=1}^{M} \mathbf{X_k}\mathbf{A}\mathbf{R_k^T} + \mathbf{X_k^T}\mathbf{A}\mathbf{R_k} \right] \left[\sum_{k=1}^{M} \mathbf{B_k} + \mathbf{C_k} + \delta\mathbf{I} \right]^{-1} \tag{3.24}$$

, where $\mathbf{B_k} = \mathbf{R_k}\mathbf{A^T}\mathbf{A}\mathbf{R_k^T}$ and $\mathbf{C_k} = \mathbf{R_k^T}\mathbf{A^T}\mathbf{A}\mathbf{R_k}$

Optimization of $\mathbf{R_k}$

$$\mathbf{R_k} = (\mathbf{Z^T}\mathbf{Z} + \delta\mathbf{I})^{-1}\mathbf{Z^T}\mathbf{vec}(\mathbf{X_k}) \tag{3.25}$$

, where $Z = Z \oplus Z$

Although there exist other tensor factorization algorithms, RESCAL [6] 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 [109].

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

Algorithm 4: Alternating Least Squares optimization algorithm (ALS).

input : Tensor $\mathcal{X}$.
Number of latent factors R.
Regularization term δ.
Maximum iteration threshold $maxIterations$.
Change criterion threshold $changeCriterion$.
output: Matrix $\mathbf{A}$ and tensor $\mathcal{R}$

```
1  begin
2      // initialization
3      A = randomizeA(rowNumber(X), R)
4      R = randomizeR(R, R, sliceNumber(X))
5      // iteratively update
6      while iteration < maxIterations and change > changeCriterion do
7          iteration = iteration + 1
8          // update A while R is fixed (See Formula 3.24)
9          A = updateA()
10         // update R while A is fixed (See Formula 3.25)
11         for k ∈ 1 ... sliceNumber(X) do
12             R_k = updateRk()
13         change = evaluateChange(A, R)
14     return(A, R)
```

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.

3.3.1.4 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 $\mathbf{k}$ from domain M from the $\mathbf{i}$-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.

As first, our method (Algorithm 5) model a tensor $\mathcal{X}$ for the input RDF dataset and the ageing constant λ. Afterwards, we compute factorization for the tensor $\mathcal{X}$ with the extended RESCAL algorithm. The factors returned from factorization we use to recon-

Algorithm 5: Time-Aware Link Prediction

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

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

```
begin
    // model tensor (see Section 3.3.1.2)
    X = toTensor(RDF, λ)
    // factorization (see Section 3.3.1.3)
    A, R = factorize(X)
    // matrix reconstruction (see Formula 3.19)
    X̂_k = A R_k A^T
    // read candidates from the i-th row
    candidates = X̂_k[i, 1...N]
    // filter existing links
    candidates = filter(candidates)
    // sort in decreasing order
    candidates = sort(candidates)
    // return Top-L values
    return candidates[0 : L]
```

Table 3.12: Example of reconstructed tensor $\mathcal{X}$ ($R = 3$).

	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

struct 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. Since, only one type of link is required and the source of links is also predefined, we read values x_{ijk} for the **i**-th row and each **j**-th column of $\widehat{\mathbf{X}_k}$. The values indicate whether a link between the **i**-th entity and entity in the **j**-th column should exist. We sort the values of the vector 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 3.3.2. ***Example of time-aware link prediction***

Consider data from Example 3.3.1 as an input RDF dataset. For simplicity, it contains only one type of the link ($k = usedAPI$). A tensor $\mathcal{X}$ in Table 3.11 corresponds to the first step of the algorithm for $\lambda = 0.01$. The second step factorizes the tensor to matrices $\mathbf{A}, \mathbf{R}_k$ and the third step provides the approximation of the tensor. Table 3.12 shows an example of the reconstructed matrix $\widehat{\mathbf{X}_k}$ ($R = 3$). 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. Given the list of candidates, we can select 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.

3.3.2 Experiments

In this section we demonstrate the time-aware link prediction method on the real-world dataset from ProgrammableWeb. Furthermore, we also performed evaluation experiments on other datasets.

The following questions we address in our experiments:

- *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.

3.3.2.1 Linked Web APIs Dataset

For evaluation purposes, we created an extended version of the *Linked Web APIs* dataset. The dataset is an RDF representation of the ProgrammableWeb[12] 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 3.19) uses several well know ontologies and vocabularies: FOAF[13] ontology (*prefix foaf*) - concept *foaf:Person* describes users and property *foaf:knows* describes a social relationship between users, WSMO-lite [143] ontology (*prefix wl*)- concept

[12]http://www.programmableweb.com/
[13]http://xmlns.com/foaf/spec/

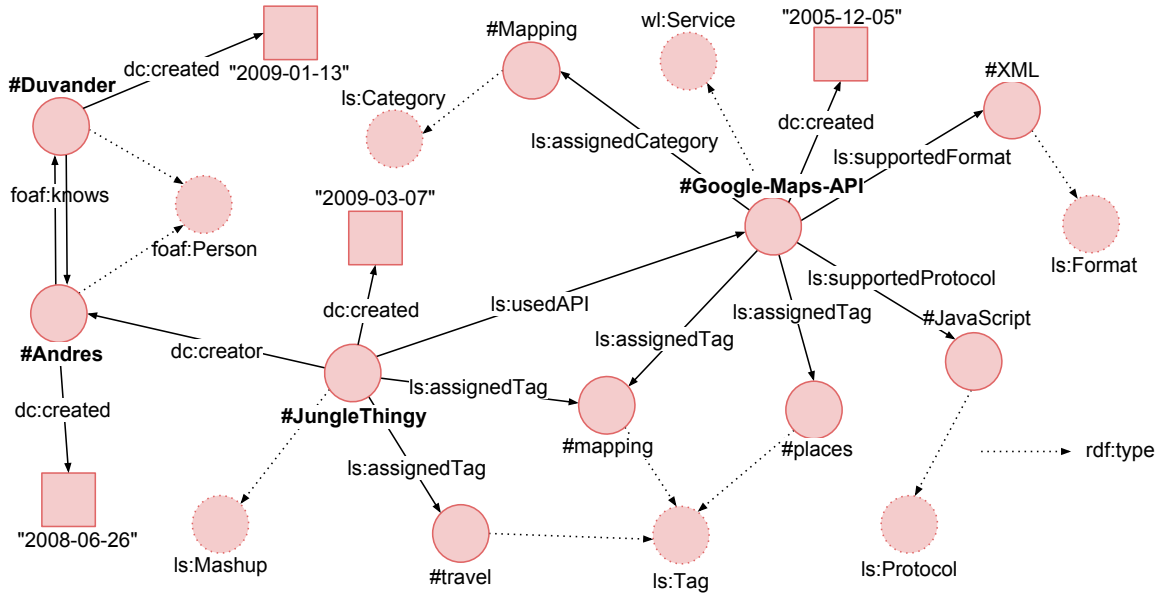


Figure 3.19: Excerpt from the extended Linked Web APIs dataset

wl:Service describes Web APIs, Dublin Core[14] 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 [70] 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*.

3.3.2.2 Experiments Settings

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 (Figure 3.20).

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

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

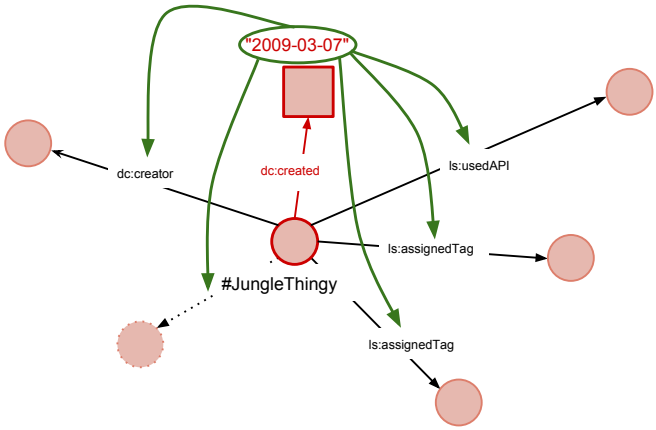
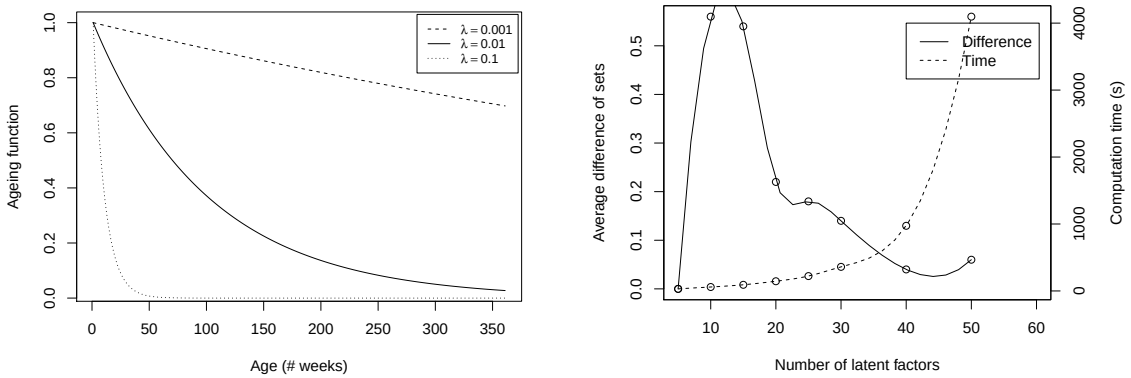


Figure 3.20: Example of the propagation of the temporal information from $dc : created$ to all related links with the same entity.

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 3.21a 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 subsequent iterations. Figure 3.21b 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 research we do not focus on the performance and scalability of the algorithm. We refer the reader to [6] for more details on the performance of the RESCAL factorization.



(a) Ageing function (b) Number of Latent factors

Figure 3.21: Experiments settings

3.3.2.3 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.

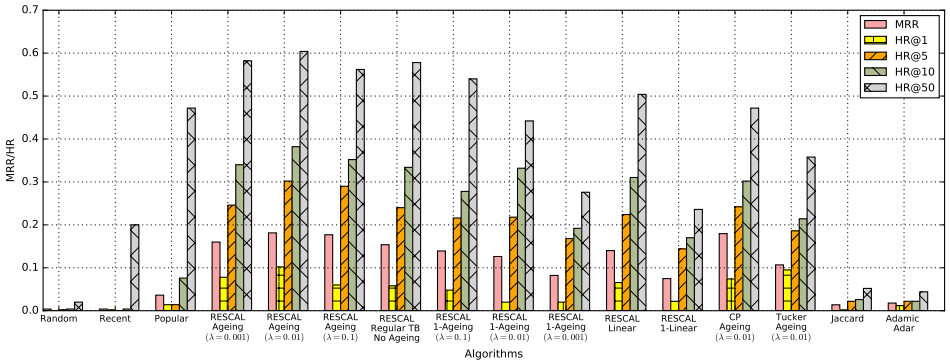


Figure 3.22: ProgrammableWeb: Mean Reciprocal Rank (MRR), HitRatio at top-k (HR@k)

- *CP and Tucker:* tensor decomposition CP (CANDECOMP/PARAFAC) and Tucker [21] using tensor model with ageing function and $\lambda = 0.01$.
- *Jaccard and Adamic Adar:* baseline graph based methods for link prediction in social networks [111] 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 3.22 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 *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 ($\lambda = 0.01$).

3.3.2.4 Significance of Time-Aware Link Prediction

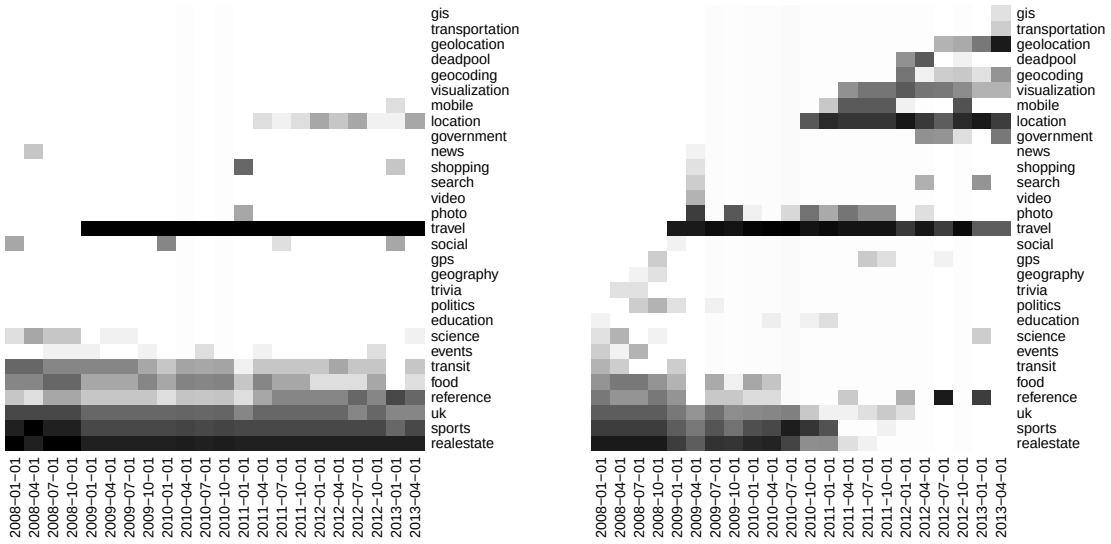
In this experiment we test how the time information influences items and their position in a list of top-L predicted links. To study the influence of time, we focused on a simple tagging task. The goal is to find a set of tags which should be assigned to a specific API (predicted links to tags can be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

Table 3.13: 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

Table 3.13 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.
In order to explore differences in both sets we run the same experiment over time (i.e., by using the 22 snapshots). Figures 3.23a,3.23b 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 3.23a depicts positions when the ageing is 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 3.23b 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



(a) Without Ageing (b) With Ageing

Figure 3.23: Visualization of positions for each snapshot

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 3.3.2.6 for more details.

3.3.2.5 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 tags and APIs or mashups and APIs. For example, to find top-10 APIs that should also have the tag *mapping* or top-10 mashups that could benefit from the Flicker API.

Table 3.14 presents the top 10 Web APIs (their names and dates, they were added to directory), which should also have the tag *mapping*. For this experiment we used the models of last snapshots with and without ageing from March 2013. In case the temporal information is not considered, the distances of dates for predicted APIs are higher. With Ageing, the predicted APIs are influenced by time and more recent APIs are predicted.

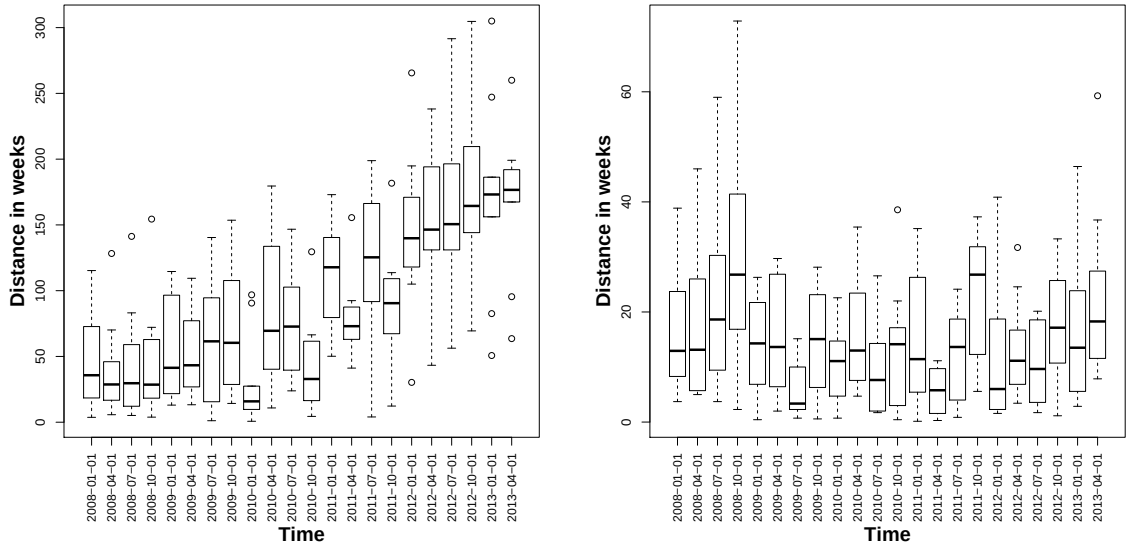
In the second experiment we performed the following task: find top-10 mashups that could benefit from the Flicker API. We run the experiment for all 22 snapshots. Figures 3.24a and 3.24b depict a distance in weeks of top 10 mashups from t_0 of every snapshot. We use a standard box plot to examine distributions of distances graphically. Figure 3.24b

Table 3.14: Top 10 APIs which should have tag *mapping* on the 1st of April 2013

	Without Ageing		With Ageing	
Position	**Name**	**Date**	**Name**	**Date**
1	Placr	2011-07-07	SetGetGo IP Geolocation	2013-12-29
2	BestParking	2011-01-06	JetSetMe	2012-11-11
3	Tube Updates	2011-03-27	Frontier Airlines Word Wheel Local	2013-02-10
4	ParkWhiz	2010-09-30	Eaupen	2012-12-29
5	Eaupen	2012-12-29	DC Location Verifier	2012-10-17
6	NAVTEQ Traffic	2011-09-11	TripCheck	2013-03-02
7	Jeppesen Journey Planner	2011-10-26	View	2013-03-01
8	View	2013-03-01	WikiSherpa	2012-07-06
9	MyTTC	2011-08-13	ThinkGeo Cygnus Track	2012-10-26
10	Pearson Eyewitness Guides	2011-09-16	weather-api.net	2012-12-19

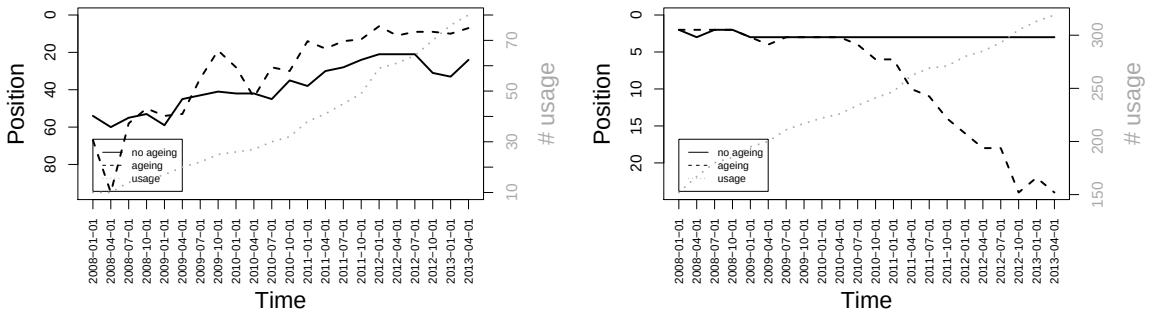
presents much lower distances than Figure 3.24a. These results support our assumption that predicted entities in top-10 lead to links between entities with time information closer to t_0 (i.e., the present time of a particular snapshot) than the link prediction that does not use time information.

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.



(a) No Ageing (b) Ageing

Figure 3.24: Distance of predicted Mashups from the ending time of snapshot



(a) Position of tag *geocoding* (b) Position of tag *realestate*

Figure 3.25: Evolution of position over time for a specific tag

3.3.2.6 Impact of Evolution of Structure

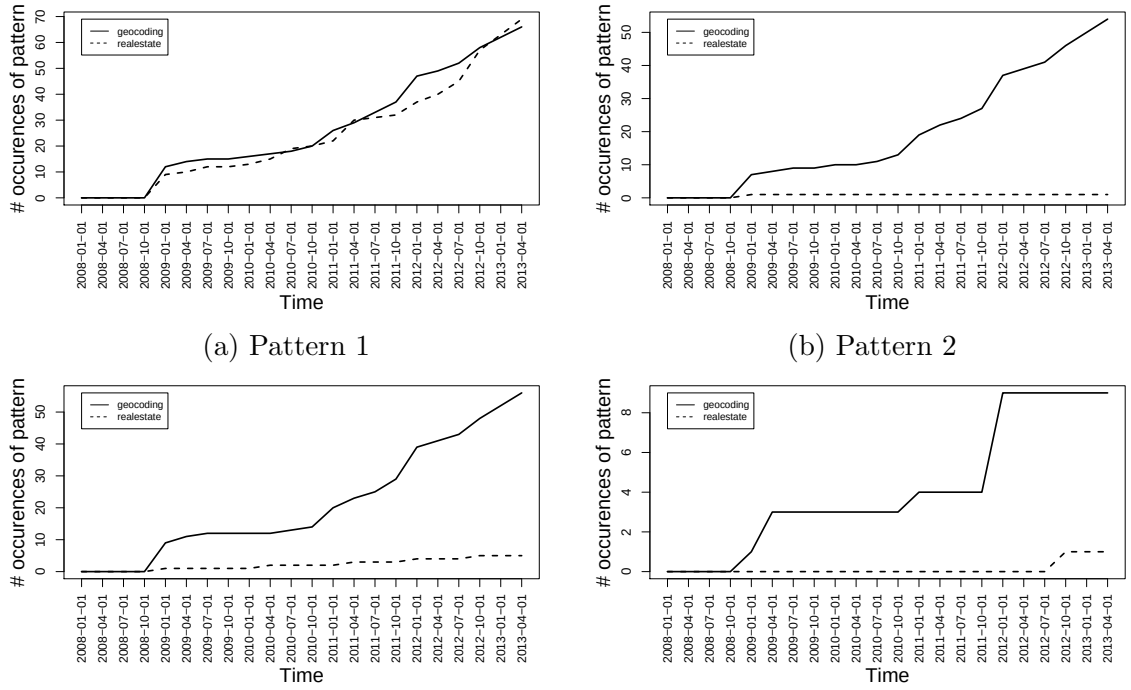
In this experiment, we demonstrate how the proposed method takes into account the evolution of the datasets' structure when predicting new links.

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 3.25a and 3.25b 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 3.25b shows a high position of the tag *realestate* when no ageing is used. This is influenced by the high number resources (APIs and mashups) 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 3.25a 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



(a) Pattern 1

(b) Pattern 2

(c) Pattern 3

(d) Pattern 4

Figure 3.26: Number of occurrences for each pattern

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 3.26a-3.26d 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

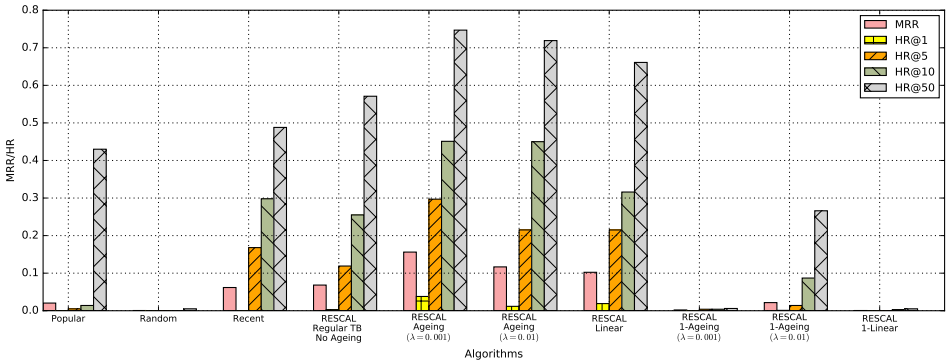


Figure 3.27: DBpedia Movies: Mean Reciprocal Rank (MRR), HitRatio at top-k (HR@k)

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 3.25b) 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 3.25a).

3.3.2.7 Evaluation on Other Datasets

For evaluation purposes of the proposed method, we prepared another graph-based RDF dataset. We focused on a subset of data from the well-known knowledge base *DBpedia* [15].
Movies Dataset. We selected a subset of movies from the DBpedia according to existing mappings [A.3] for the *MovieTweetings* dataset [14]. A movie ratings dataset extracted from tweets on Twitter. The mappings dataset contains links for over *15 000* movies. We extracted from the DBpedia following main information for each movie: assigned types and categories, actors starring in a movie, distributors, music composers, producers, writers and directors. Those information also represent eight types of links in a complete dataset. In total, there are over *66 600* unique entities and more than *280 000* links in the dataset. As temporal information we use the release date of each movie and we propagate it as a creation time to all links associated to a movie.
Evaluation settings. We set the ageing constant to $\lambda = 0.001$, since this value provides a distribution of values over the whole period of release dates since *1900s* till present 2015. The number of latent factors was experimentally set to *30*. The value *30* provides best results from the perspective of used evaluation metrics. As evaluation metrics we used both Mean Reciprocal Rank (*MRR*) and Hit Ratio at top-k (*HR@k*). On this dataset we focused on the following algorithms: *Random*, *Recent*, *Most Popular*, *Regular TB Link Prediction* and *Time-aware Link Prediction with Ageing*. *Training set* and *Test set* were created similarly to the previous evaluation: we put randomly selected 1% of newest links

[15] `http://dbpedia.org`

(from year 2015) to the testing set. The rest of the data we put to the training set. We performed repeated random sub-sampling cross-validation.

Results of Evaluation. Figure 3.27 depicts results of the evaluation on movies from the DBpedia. Both *Random* and *Most Popular* does not perform well, since they do not consider structure and temporal dimension. *Recent* takes into account time and has significantly better results than *Random* and *Most Popular*. *Regular TB Link Prediction* does not reflect creation time. It has slightly better results in terms of *MRR* but does not provide better results for *HR@k*. *1-Ageing* and *1-Linear* dos not perform well because of promoting older links. *Ageing* ($\lambda = 0.01$) and *Ageing (Linear)* outperforms the regular link prediction, but the distribution of values is not over the whole interval and does not follow forgetting curve, respectively. Our proposed method *Ageing* ($\lambda = 0.001$) outperforms all other approaches. It takes into account both structural and temporal information.

3.3.3 Implementation

The Time-Aware Link prediction method is available as an open source on GitHub [16]. The method is implemented in *R* that provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible [17].

Our implementation contains functionalities to construct a tensor with temporal aspects, RESCAL factorization algorithm, link prediction method and a running example. The Listing 3.9 demonstrates an example of usage.

```
# import algorithm
source("./lib/talp.R")

# ageing function
ageing <- function(date1, date2="2010-12-01"){
    diff <- as.numeric(difftime(date2, date1, unit="weeks"))
    if(diff>0){
        value <- 1*exp(-0.01*diff)
    } else {
        value <- 0
    }
    return(value)
}

# define link types
linkTypes <- c("knows")

# define entities
entities <- c("Person1", "Person2", "Person3")

# initialize tensor
t <- initializeTensor(entities, linkTypes)
```

[16] https://github.com/jaroslav-kuchar/Time-Aware-Link-Prediction
[17] https://www.r-project.org/

```
23
24 # add links: source, link type, target, value of link ; 1 for "strong" links
       , otherwise decreased by ageing
25 setTensorValue(t,"Person1","knows","Person2", ageing("2008-06-26"))
26 setTensorValue(t,"Person1","knows","Person3", ageing("2010-11-03"))
27
28 # print original tensor
29 printTensor(t)
30
31 # factorization, 10 latent factors, max 100 iterations and difference
       between iterations 0.01
32 factorizationOutput <- factorization(t, 10, 100, 0.01)
33
34 # predict links
35 predictedLinks <- topNtargets("Person1","knows", t, factorizationOutput)
36 print(predictedLinks)
```

Listing 3.9: Time-Aware Link Prediction in R

3.3.4 Discussion

Robustness. Although we evaluated our method on a domain-specific datasets from ProgrammableWeb and DBpedia, 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. Similarly, we have selected the movies from DBpedia since the release date can be used as the creation time for the proposed method.

Temporal Information. Due to the nature of the data from ProgrammableWeb and DBpedia we deal with a specific form of time assigned to an entity as the created time (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 (e.g. presence of termination time of a link).

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.) [141] 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 [141], 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. We have demonstrated that the proposed ageing function outperforms other formulas. However, the formula may vary in different use cases and domains.
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 3.3.2.6 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.

3.3.5 Summary

Despite many existing link prediction algorithms, there is still lack of approaches that are focused on multiple types of links and time information about existence of links. Rich graph-based datasets, including RDF, can contain links that were included in the past. Such links may loose their significance in the future. The creation time of a link thus provides meaningful information that might reflect the credibility of the link. In this section we present a method for a link prediction task that considers both structural and temporal aspects of graph-based datasets - *Time-Aware Link Prediction*. Our method incorporates the creation time of links to a tensor model and we use the tensor factorization as an underlying concept for the link prediction. We evaluated our method on the RDF representation of the ProgrammableWeb directory and a subset from the well-known knowledge-base DBpedia. Furthermore, we also performed a set of experiments to explore the details and demonstrate the capabilities of the method. The results from the evaluation and experiments show that the temporal dimension influences the results and it is an important aspect for the link prediction.

3.4 Personalized Selection of Entities

In this section we focus on a method to effectively utilize the rich semantic representation, such as RDF representations. The RDF dataset can be processed by various algorithms depending on the use case. In our research we are focused on rich representations connecting users and content. Modelling and understanding various contexts of users is important to enable personalised selection of entities in the RDF. We develop a novel selection method that provides personalized recommendations. The method has the following characteristics: 1) *social and linked* - it exploits relationships among entities, and social relationships among users such as who knows who, 2) *personalized* - it takes into account user's preferences such as users the user knows and preferences that define importances of predicates, and 3) *temporal* - it takes into account a time when the entity appeared in the RDF for the first time.

Since approaches and algorithms in this section are originally from the domain about networks and graph, we adopted the respected common terminology using graphs, nodes and edges for semantic representations, entities and links, respectively.

The proposed method calculates a maximum activation from initial nodes of the graph (defined by a user profile), to each node from a set where a node in the set represents an entity candidate. We adopt the term activation from the spreading activation method [144] and we use it as a measure of a connectivity between source nodes (initial nodes defined by a user profile) and a target node (an entity candidate). We use flow networks as an underlying concept for evaluation of the maximum activation in the graph. We evaluate it on several experiments showing that the method gives better results over traditional popularity-based recommendations. We call the proposed approach: *Maximum Activation Method* [A.13].

3.4.1 Method

3.4.1.1 Definitions

Graph model. *A rich RDF representation of users connected to the content.* Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{I})$ be a graph representing RDF. where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ is a set of edges and $\mathcal{I}$ is a capacity function. A node in $\mathcal{V}$ represents either an entity (individual or class) or a user. An edge $e \in \mathcal{E}$ represents a mutual (bidirectional) relationship between two nodes as follows: for a property in the dataset we create an *inverse property* such that when (o_1, p, o_2) is a triple where o_1, o_2 correspond to nodes in $\mathcal{V}$ and p corresponds to an edge in $\mathcal{E}$, we create a new triple (o_1, p^{-1}, o_2) where p^{-1} is an inverse property to p.

User profile. *A set of nodes the user is connected with.* Let $P = \{p_1, p_2, ..., p_n\}, p_i \in \mathcal{V}$ be a set of nodes that represent a user profile. The nodes in P may represent the user himself, nodes that the user is connected with or has any other explicit or implicit relationships with.

Target nodes. *A set of nodes as candidates for selection.* Let $W = \{w_1, w_2, ..., w_m\}$, $w_i \in \mathcal{V}$ be a set of nodes that represent a user request as entity candidates. The Maximum Activation method then calculates a maximum activation a_i for each entity candidate $w_i \in \mathcal{W}$. The higher number of the maximum activation denotes a candidate with a higher rank, that is the preferred candidate over a candidate with a lower maximum activation.

Preference function. *A function which defines an importance of an edge type for the user.* Let $S(e_i) \in \{x \in \mathbb{N} | 0 \leq x \leq 100\}$ be a user preference function that defines an importance of the edge e_i (i.e., how the user sees an importance of semantics represented by the edge). An importance $S(e_i) < 50$ indicates that the user does not prefer the edge's semantics, an importance $S(e_i) > 50$ indicates the the user prefers the edge's semantics and the importance $S(e_i) = 50$ indicates a neutral value. A user may chose an arbitrary number of edges for which he/she defines preferences. Edges for which the user does not define any preferences have a default preference 50.

Capacity function. *A function which associates a capacity of an edge with a natural number.* The function is defined as $\mathcal{I} : \mathcal{E} \rightarrow \mathbb{N}$, where $\mathcal{E}$ is a set of edges. The value of capacity for each edge defines a maximum flow that can go through the particular edge. We denote an activation that can be sent between two nodes linked with an edge e as a natural number $i(e) \in \mathbb{N}$. The activation sent through an edge cannot exceed the capacity of the edge defined by the capacity function

$$\mathcal{I}(e_{i,t}) = S(e_i) * \mathcal{A}(e_{i,t}) \tag{3.26}$$

where $S(e_i)$ is a user preference function and $A(e_{i,t})$ is the *exponential ageing function.*

Ageing function. *A function to decrease the influence of older edges.* We define the exponential ageing function as

$$\mathcal{A}(e_{i,t}) = \mathcal{A}(e_{i,t_o}) * e^{-\lambda t} \tag{3.27}$$

where $\mathcal{A}(e_{i,t})$ is an age of the edge e_i at time t, $\mathcal{A}(e_{i,t_o})$ is the initial age of the edge e_i at the time the edge appeared in the graph $\mathcal{G}$ (i.e., values of *dc:created* property) and λ is an *ageing constant*. The ageing constant allows to configure an acceleration of the ageing process. Since our method gives better results for better connected nodes in the graph, the ageing function allows to control an advantage of "older" nodes that are likely to have more links when compared to "'younger" ones.

Example 3.4.1. ***Example of a graph-model.***

Consider an RDF dataset with two users(u_1, u_2), two mashups (m_1,m_2) they like are placed in different categories (c_1,c_2). User u_1 knows u_2 for about three weeks ($u_1 \xrightarrow{t=3} u_2$).

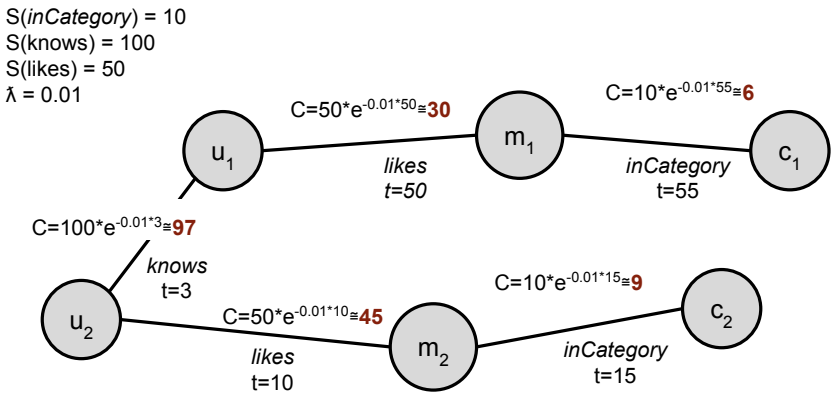


Figure 3.28: Example of a graph model for the Maximum Activation Method.

Mashup m_1 was placed to the category c_1 fifty-five weeks ago ($m_1 \xrightarrow{t=55} c_1$) and user u_1 started liking mashup m_1 fifty weeks ago ($u_1 \xrightarrow{t=50} m_1$). Further, mashup m_2 was placed to the category c_2 fifteen weeks ago ($m_2 \xrightarrow{t=15} c_2$) and user u_2 started liking mashup m_2 ten weeks ago ($u_2 \xrightarrow{t=10} m_2$). Let assume the user prefers social aspects of the graph and does not prefer associations to categories. His preference function is defined as follows: links of type inCategory are not preferred ($S(inCategory) = 10$), the knows is mandatory ($S(knows) = 100$) and the likes is neutral ($S(likes) = 50$). Figure 3.28 shows the example of computed capacities for $\lambda = 0.01$.

3.4.1.2 Maximum Activation Method

Our proposed method is based on the concept of flow networks and the problem of finding the maximum flow between source and target nodes. We use the well-known Ford-Fulkerson algorithm [7] as an underlying concept. The Algorithm 6 summarizes the overall principle of the Ford-Fulkerson algorithm *FF*. The *FF* algorithm first sets the initial activation/flow for each edge to 0 and tries to find an *improving path* on which it is possible to increase the activation by a minimum value greater then 0. If such path is found, the algorithm increases activations on every edge on the path and tries to find another path. When no more path is found, the algorithm ends. The result of *FF* is the set C (minimum cut) that contains every last edge from all paths from the source towards the target when an improving path is not possible to find.

We calculate the Maximum Activation according to the Algorithm 7. As first the graph is converted to flow network with bidirectional edges. When we construct the graph $\mathcal{G}$ from the dataset, for every predicate we create a bidirectional edge. A graph with bidirectional edges provides a richer dataset for maximum activation evaluation. A large graph with unidirectional edges may contain many dead end paths that may limit the number of improving paths that the algorithm would be able to find from the source to the target nodes. Evaluation of maximum activation on such graph would not provide many

Algorithm 6: Ford Fulkerson [7]

```
   input : Start node s
           Target node t
           Graph G
   output: Minimum cut C
           Maximum flow f
 1 begin
 2     // initialize flow
 3     for edge ∈ G do
 4         flow(edge) = 0
 5     f = 0, C = ∅
 6     // find max flow
 7     while existsImprovingPath(G) do
 8         P = improvingPath(G)
 9         f,C = improve(G,P)
10     return C, f
```

interesting results.

As we noted earlier, we interpret the maximum activation of the graph as a measure that indicates how well the source nodes are connected with the target. In general, the more improving paths exist between the source and the target, the higher maximum activation we can get. However, the value of the maximum activation is also dependent on constraints and the creation time of edges along the improving paths when the ageing function is applied. The second step of our algorithm is to compute capacity of each edge in the graph.

In lines 7–10, the algorithm creates a virtual node representing a single source node with links connecting the virtual node and all other nodes from the user profile. Any edge that connects the virtual node with any other node in the graph has a capacity set to a very large value so that the edge does not constrain the maximum activation. In lines 11–15, the algorithm finds a maximum activation for each candidate w_i from the virtual node p'. For this we use the Ford-Fulkerson algorithm to find a maximum activation from the *source node* (i.e., the virtual node) to the *target node*. In line 15, the algorithm finally calculates the maximum activation as a sum of all activations of edges in C.

The edges in C are constraining the maximum activation which means that if capacities of such edges increase, the maximum activation can be increased. Note, however, that we assign capacities based on semantics of edges thus by changing a capacity on an edge in C, we also change capacities on other edges not in C. Running the algorithm again on the graph with new capacities will lead to a different set C and different maximum activation. In other words, it does not hold that increasing a capacity on any edge in C will lead to a higher maximum activation. This also means that maximum activation that

Algorithm 7: Maximum Activation Method

input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{I})$ constructed from RDF.
A user profile $P = \{p_1, p_2, ..., p_n\}$.
A set of candidates $W = \{w_1, w_2, ..., w_m\}$.
A user preference function $S(e_i)$.
Ageing constant λ

output: A set of maximum activations $\{a_i\}$ evaluated for each $w_i \in W$.

```
1  begin
2      // construct a flow network
3      addBidirectionalEdges(G)
4      // compute capacity for each edge
5      for e_i ∈ E do
6          capcity_{e_i} = computeCapacity(S(e_i), λ)
7      // create a virtual source node p'
8      add node p' to V
9      for p_i ∈ P do
10         add edge e(p', p_i) to E, S(e) ← 100000, A(e) ← 1
11     // calculate a maximum activation a_i from
12     // a virtual node p' to every candidate w_i
13     for w_i ∈ W do
14         C ← FF(p', w_i, G)
15         a_i ← Σ_{e_i ∈ C}(I(e_i))
16     return({a_0, ...})
```

our algorithm evaluates has a global meaning while activations on individual edges do not have any meaning. Defining capacities for individual edges is the subject of our future work.

Example 3.4.2. ***Example of a Maximum Activation.***

For our example we consider that a user profile P is connected to u_1 and c_1 supposing he knows the user and the category is his favourite one. As candidates for target nodes we set both mashups $W = \{m_1, m_2\}$. Based on capacities presented in Example 3.4.1, we applied our maximum activation method. The Figure 3.29 shows results. Even if the mashup m_1 is connected by two possible paths, mashup m_2 gets more activation. It is caused by the preference for social edges (knows) and by the preference for "newer" edges as well.

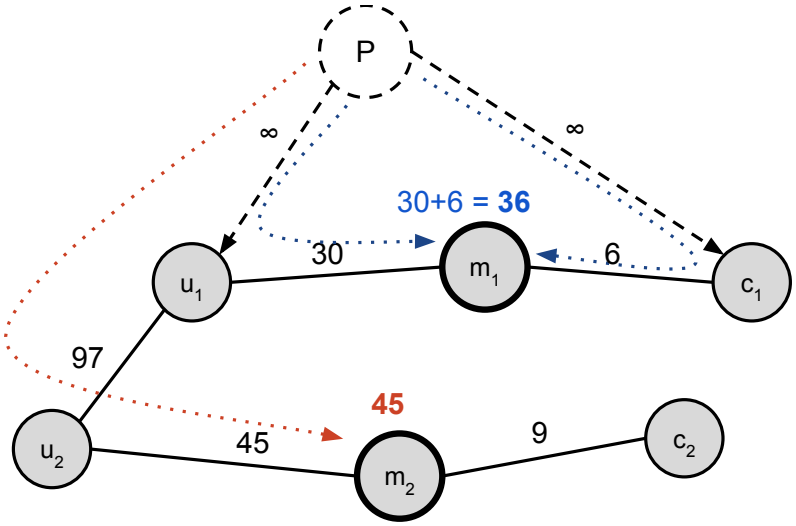


Figure 3.29: Example of the Maximum Activation Method.

3.4.2 Evaluation

In this section we present several experiments and their results[18] that use maximum activation for the Web APIs selection.

For our experiments we use the Linked Web APIs dataset. The dataset contains all user profiles for users that created at least one mashup. We also extracted profiles on all categories, tags, mashups and Web APIs. The snapshot we use covers the period from the first published API description in June 2005, till May 18th, 2012. The snapshot includes 5 988 APIs, 6 628 mashups and 2 335 user profiles. In experiments we addressed following questions:

- *What is the impact of user preference function on results of the maximum activation?*
- *How does the ageing factor influence the maximum activation?*
- *How can the popularity of an API evolve over time?*
- *How to make the process of building a mashup more personalised and contextualised?*

3.4.2.1 Setting the Ageing Constant

We experimentally set the ageing constant to a value $\lambda = 0.1$ and the age period to one week ($t = week$). Our graph contains data since June 2005, that is approximately 360 weeks. Figure 3.30 depicts an effect on ageing function for different λ. Note that the higher the constant is, the algorithm promotes the more recently added APIs and Mashups.

[18]Full results of the experiments are available at `http://goo.gl/GKIbo`

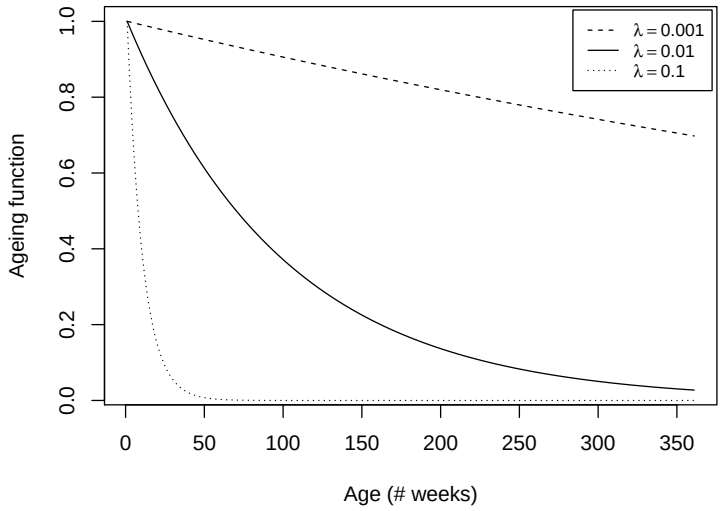


Figure 3.30: Ageing function

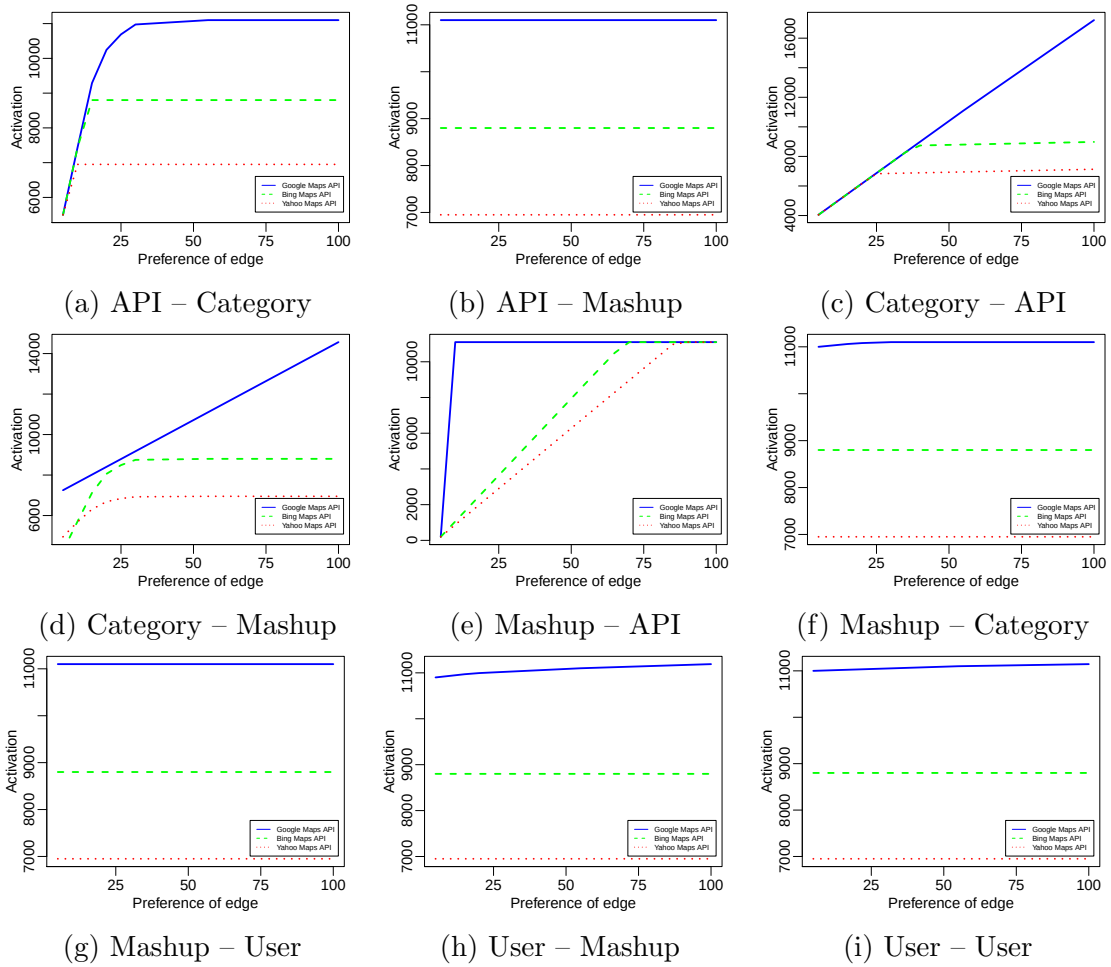
3.4.2.2 Impact of the User Preference Function

The user preference function defines an importance of the edge, that is how the user sees the importance of semantics represented by the edge. For example, the user can give a higher importance to edges representing a friendship (*foaf:knows*) than to edges between mashups and Web APIs (*ls:usedAPI*). The importance values, along with the chosen edge ageing constant λ, are used to compute the total capacity of an edge (see definition (3.26). To study the influence of an importance value on a single edge, we were gradually increasing the value from 0 to 100 by a step of 5 and fixed importance values of all other edges to 50. We run this experiment for 3 different well-known APIs, namely Google Maps, Bing Maps and Yahoo Maps.

Figure 3.31 shows the experiment results: the importance value on the edge *API–Mashup* (Fig. 3.31b) and *Mashup–User* (Fig. 3.31g) does not have influence on the maximum activation. Slight influence has the importance value on edges *Mashup–Category* (Fig. 3.31f), *User–Mashup* (Fig. 3.31h) and *User–User* (Fig. 3.31i). Fig. 3.31a further shows that different importance values have various ranges of influence: the importance value for the *API–Category* has influence in a range $0 - 5$ for Yahoo Maps API, $0 - 10$ for Bing Maps API, and $0 - 15$ for Google Maps API, while higher importance values do not have any influence as the maximum activation is limited by the capacities of other types of edges.

3.4.2.3 Impact of the Ageing Constant λ

The ageing constant λ is a configurable parameter which influences the value of assigned edge capacity. The higher the λ is, the more recent edges will be preferred – that is, the older edges will have a lower capacity. In different datasets edges can occur more



(a) API – Category (b) API – Mashup (c) Category – API

(d) Category – Mashup (e) Mashup – API (f) Mashup – Category

(g) Mashup – User (h) User – Mashup (i) User – User

Figure 3.31: Impact of Importance values

or less frequently therefore appropriate value for the λ should be set. Setting high λ in datasets where the edges occur less frequently may lead to very low edge capacities and consequently to the low activation value. In other words, the ageing constant λ makes the selection method more adaptable to different datasets.

For this experiment we chose a random user Dave Schappell[19] and we calculated the maximum activation for each API candidate in the "mapping" category. We evaluated the results in the period from 1st of June 2009 (shortly after the user registered his profile) till 1st of June 2012 with a period of age set to 1 week. We set the ageing constant λ to values 0.01 and 0.1. By setting the ageing constant we are able to accelerate the ageing process, that is we get a lower capacity on older edges. Fig.3.30 shows, setting the ageing constant to 0.1 we get higher maximum activation for edges that appeared in the last 50 weeks, and setting it to 0.01 in the last 350 weeks.

Table 3.15 shows the configuration of importance values for various types of edges for this experiment and Table 3.16 and 3.17 shows the results of this experiment for λ set to

[19]http://www.programmableweb.com/profile/daveschappell

Table 3.15: Importance Value Configuration

Edge name	Importance value	Edge name	Importance value
API–Category	50	Mashup–Category	70
API–Mashup	50	Mashup–User	50
Category–API	70	User–Mashup	90
Category–Mashup	20	User–User	90
Mashup–API	70	/	/

Table 3.16: Summarised ranking results with λ=0.01

Node ID	API name	Date created	Max-Activation $\lambda = 0.01$		PW rank
			value	rank	
2053	Google Maps API	2005-12-05	**5559**	1	1
2041	Google Earth API	2008-06-01	1080	2	5
2057	Google Maps Data API	2009-05-20	1043	3	8
2052	Google Geocoding API	2010-12-09	1028	4	11
3032	Microsoft Bing Maps API	2009-06-09	853	5	2
2060	Google Maps Flash API	2008-05-17	792	6	6
5827	Yahoo Geocoding API	2006-02-14	715	7	4
5836	Yahoo Maps API	2005-11-19	707	8	3
493	Bing Maps API	2009-06-09	662	9	10
2070	Google Places API	2010-05-20	553	10	18

Table 3.17: Summarised ranking results with λ=0.1

Node ID	API name	Date created	Max-Activation $\lambda = 0.1$		PW rank
			value	rank	
2053	Google Maps API	2005-12-05	**503**	1	1
5531	Waytag API	2012-04-27	210	2	230
4330	Scout for Apps API	2012-04-20	190	3	202
4535	Google Geocoding API	2010-12-09	184	4	11
3815	Pin Drop API	2012-03-27	135	5	191
5950	Zippopotamus API	2012-04-26	123	6	233
5825	Yahoo Geo Location API	2012-04-23	120	7	230
1836	FreeGeoIP API	2012-03-29	112	8	116
5156	Trillium Global Locator API	2012-04-18	111	9	109
1430	eCoComa Geo API	2012-05-15	108	10	108

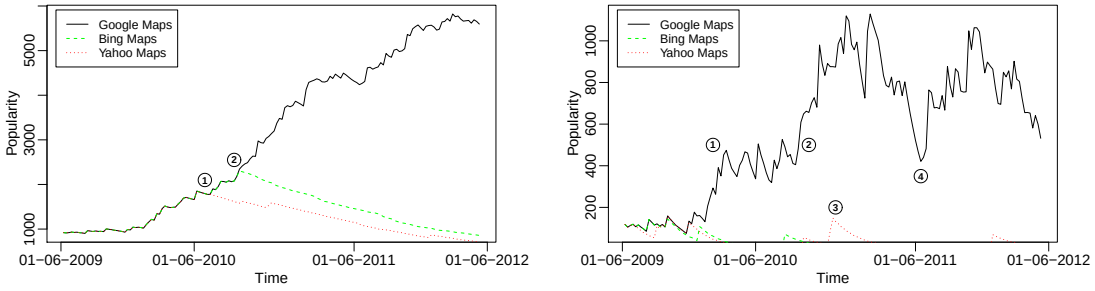
0.01 and 0.1 respectively. In these tables, the"PW rank" column shows a popularity-based ranking used by the ProgrammableWeb which is only based on a number of mashups used by an API. Google Maps API is the highest ranked API by our method (for both λ=0.01

and λ=0.1) and also is the highest ranked by the Programmable Web popularity-based method. For $\lambda = 0.01$, the method favors the recent APIs but also does not ignore APIs that were actively used in the past 350 months (approx. 7 years).

From the results in Table 3.17 it is possible to see that the ageing constant $\lambda = 0.1$ promotes newer APIs while at the same time it does not ignore the all-time popular APIs such as Google Maps API and Google Geocoding.

3.4.2.4 Popularity of APIs over Time

In this experiment we examine a popularity of 3 APIs from the "mapping" category for the user Dave Schappell in different points in time. We use the configuration in Table 3.15 and the ageing constant λ set to values 0.01 and 0.1.



(a) With ageing constant $\lambda = 0.01$

(b) With ageing constant $\lambda = 0.1$

Figure 3.32: API popularity over Time

The results show that the Google Maps API has the highest popularity in both cases for the ageing constant set at 0.01 and 0.1. From Figure 3.32a we can see that the popularity of Yahoo Maps API and Bing Maps API follows the popularity of the Google Maps API until the time marked with (1) and (2). After the times (1) and (2), a popularity of the two APIs starts to fall. Around December 2010 and January 2012 the popularity of Yahoo Maps API experienced minimal activation growth due to several new mashups that were created and used this API.

Figure 3.32b shows a popularity of the three APIs with a more strict edge ageing. After the first half year, when the popularity of the 3 APIs is nearly the same, the popularity of the Google Maps API is starting to increase until the time marked with (1) and stays at this level until the time marked with (2). Between the times (2) and (4) Google Maps API gained a popularity up to maximum activation of 1 129, however, it also started to lose some activation due to a less number of mashups that were using this API. On the other hand, popularity of the Yahoo Maps API increased around December 2010 (3) due to its more intensive usage. As we can see, in certain cases, by using the ageing function we can get better results than the PW's popularity-based ranking.

3.4.2.5 Case Study

In this section we present a case study for personalised API selection to illustrate capabilities of our maximum activation method. We have a developer who wants to improve tourists' experience in New York, USA by creating the Visitor Mashup. The Visitor Mashup should aggregate information about different events and information about restaurants in the city and in the area of New York. Information about various points of events and restaurants should be layered on the map and dynamically updated when tourists change their locations and new events and restaurants become available.

Developer starts the process of building the Visitor Mashup by identifying groups of relevant APIs. As he progresses and selects APIs, the ranking process becomes more personalised and contextualised. The process of creating the Visitor Mashup is described by following steps when in each step the developer selects one API:

- **Maps API.** Developer builds his profile adding "maps" and "location" categories to it. He assigns a high importance value to the "API–Category". Table 3.18 shows the highest ranked results: Google Maps, Microsoft Bing Maps and Yahoo Maps. The developer decides to select the Google Maps API.

Table 3.18: Summarised ranking results for Maps API

Node ID	API name	Date created	Max-Activation λ not set		Max-Activation $\lambda = 0.01$		PW rank
			value	rank	value	rank	
2053	Google Maps API	2005-12-05	**13720**	1	**6509**	1	1
3032	Bing Maps API	2009-06-09	3720	2	238	2	10
5836	Yahoo Maps API	2005-11-19	2980	3	172	3	3

- **Events API.** The developer further searches for events API by updating his profile with "events" category, adding "Google Maps API" and preserving "maps" and "location" categories. Further, he increases an importance value of the "Mashup–API". Table 3.19 shows highest ranked results: Seatwave, Eventful and Upcoming.rg. The developer selects Seatwave API.

- **Restaurant API.** The developer searches restaurants API by adding "'food", "restaurants" and "menus" categories to his profile. This time the developer decides to use his social links and to look for APIs used by his friends developers that he adds to his profile. Table 3.20 shows the highest ranked APIs SinglePlatform, Menu Mania and BooRah. The developer selects SinglePlatform API for restaurant information and recommendations.

Table 3.19: Summarised ranking results for Events API

Node ID	API name	Date created	Max-Activation λ not set		Max-Activation $\lambda = 0.01$		PW rank
			value	rank	value	rank	
4348	Seatwave API	2012-02-28	940	3	**842**	**1**	4
1578	Eventful API	2005-10-31	3930	1	710	2	1
5371	Upcoming.rg API	2005-11-19	3220	2	411	3	2

Table 3.20: Summarised ranking results for Restaurant API

Node ID	API name	Date created	Max-Activation λ not set		Max-Activation $\lambda = 0.01$		PW rank
			value	rank	value	rank	
4522	SinglePlatform API	2012-01-30	150	2	**125**	1	6
2980	Menu Mania API	2009-12-05	**220**	1	65	2	1
611	BooRah API	2008-10-31	120	3	30	3	3

3.4.3 Implementation

We implemented the method in Java as a plug-in for Gephi: an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs [20]. The implementation is available as an open source on GitHub [21].

To use the plug-in in any other project, it has to be imported together with Gephi Toolkit [22]. It is a single library that package the essential modules from the main Gephi application. Those modules are required to successfully run the implemented plug-ins. The Listing 3.10 demonstrates a basic example of usage. We start with an initialization of required objects from Gephi. Further, we create simple graph with three nodes and two edges. The important setting for computing capacities is loaded from a properties file. The settings covers user preferences for edges and setting for ageing function as well (Please refer to the example in the repository for more details). Afterwards, the preferences are used to compute capacities for each edge. Finally, the maximum activation using Ford-Fulkerson algorithm is computed.

```
1 // Initialize a project
2 ProjectController project = Lookup.getDefault()
3   .lookup(ProjectController.class);
4 project.newProject();
5 Workspace workspace = pc.getCurrentWorkspace();
6
```

[20] http://www.gephi.org
[21] https://github.com/jaroslav-kuchar/Maximum-Activation-Method
[22] http://www.gephi.org/toolkit/

```
// Initialize a graph model
GraphModel graphModel = Lookup.getDefault()
  .lookup(GraphController.class).getModel();

// Create nodes
Node nodeA = graphModel.factory().newNode("A");
nodeA.getNodeData().setLabel("Node A");
Node nodeB = graphModel.factory().newNode("B");
nodeB.getNodeData().setLabel("Node B");
Node nodeC = graphModel.factory().newNode("C");
nodeC.getNodeData().setLabel("Node C");

// Create three edges
Edge edge1 = graphModel.factory().newEdge(nodeA, nodeB, 1f, true);
Edge edge2 = graphModel.factory().newEdge(nodeB, nodeC, 1f, true);

// Load preferences
Properties preferences = new Properties();
preferences.load(new FileInputStream("preferences.properties"));

// compute capacity
CapacityFunction cf = new PreferencesCapacityFunction(preferences);
Graph graph = graphModel.getGraph()
cf.computeCapacity(graph);

// compute maximum flow and minimum cut
FordFulkerson maxflow = new FordFulkerson(G, "A", "C");
System.out.println("Max flow value = " + maxflow.value());
```

Listing 3.10: Maximum Activation Method in Java

The plugin can be also imported to the Gephi application. From the main interface you can manually add virtual node, select start and target nodes and launch the Maximum Activation Method. The example of final visualisation is on Figure 3.33. The red node above is a virtual source, the orange one denotes a target, bold lines show the paths with non-zero flow and bold red lines present minimum cut edges. Other colors of nodes represent different types of nodes.

3.4.4 Summary

Rich semantic representations allow to utilize their relations and perform an intelligent selection of resources based on existing relations. For the same reason as in the link prediction, the temporal information plays an important role in our novel approach for personalized selection of entities within rich semantic representations. The method exploits relationships among entities, and social relationships among users such as who knows who; it takes into account users preferences such as users the user knows and preferences that define importance of specific link types, and it takes into account temporal information about links. We applied our Maximum Activation Method to the domain of selection of Web APIs. Current approaches for searching and selecting Web APIs utilize rankings based

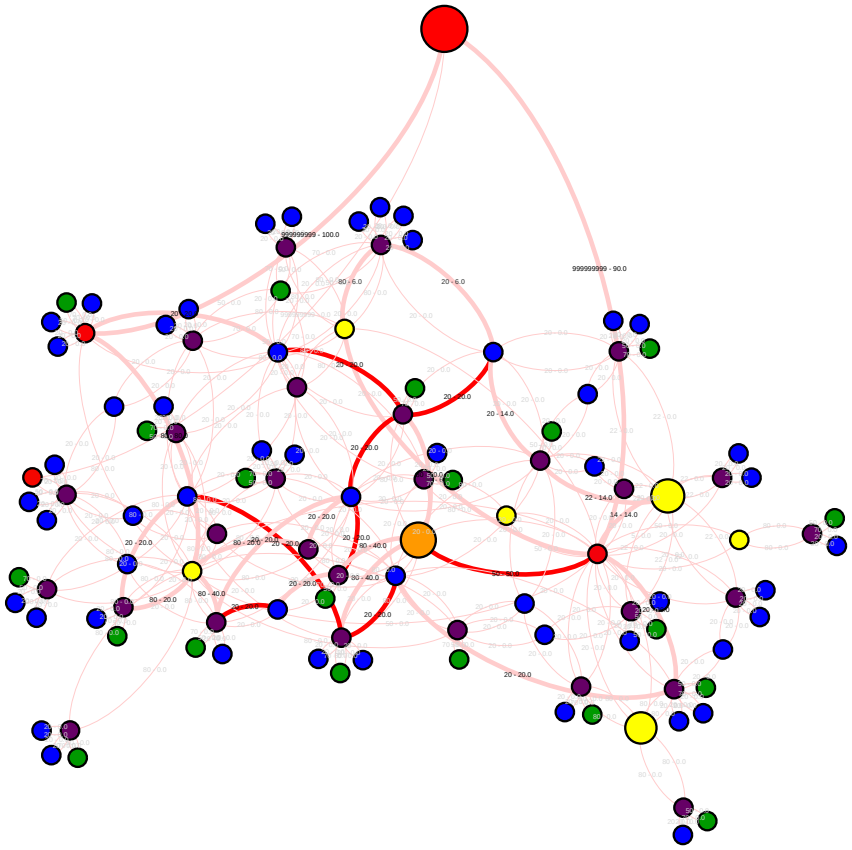


Figure 3.33: Visualisation of Maximum Activation method in Gephi - the red node above is a virtual source, the orange one denotes a target, bold lines show the paths with non-zero flow and bold red lines present minimum cut edges.

on Web APIs popularity either explicitly expressed by users or a number of Web APIs used in mashups. Such metrics works well for the large, widely-known and well-established APIs such as Google APIs, however, they impede adoption of more recent, newly created APIs. While existing popularity-based rankings use single-dimensional ranking criteria (i.e., a number of APIs used in mashups), our method uses multi-dimensional ranking criteria and with help of graph analysis methods it provides more precise results.

3.5 Rule-based Preference Learning and Recommendations

This section presents our work related to the utilization step of our methodology - how to utilize the semantics in the domain of preference learning, personalizations and recommender systems. In our research we focused on well understandable, explainable and justifiable preferences and recommendations while taking into account the semantics. Most of existing solutions use various models as an underlying concept. Those models act as black box solutions, since they use complex learning algorithms. Although there exist other learning algorithms and models, we use a rule learning and rules. Rules are considered as one of the most understandable representations for models. Humans can even add, edit or delete specific rules explicitly.

The method considers rich representations that links users and content as the input. It learns preference models using a rule learning algorithm. Preference models can be afterwards used for a personalization or even a recommendation of other content items. The advantage of the approach we propose is that it is understandable for users and also applicable for client side solutions. The client side solution preserves the privacy of users while rule based models supports understandability of models and recommendations.

We designed two approaches that are suitable for preference learning and recommendations:

- *Rule-based semantic preferences* is a method to learn user preferences in form of a rule based classifier. It consumes rich semantic representations of content items together with associated interest levels. The classfifer can be than used for labelling other objects that match the preferences. The method was mainly evaluated in [A.8, A.6, A.5]

- *Rule-based recommender system* presents a modification of the previous approach designed for recommendation of objects using a rule-based classifier. We also discuss details on aspects leading to application in real scenarios [A.7, A.2, A.18, A.19].

Methods and approaches presented in this section are mainly a joint work with Tomáš Kliegr as a part of the European LinkedTV[23] project. He mainly participated on the design and architecture of systems and he contributed to off-line evaluations with specific algorithms.

3.5.1 Definitions

Bag of Entities (BoE). *A representation of an object (content item) by a set of present semantic entities and associated types/classes.* Let the object O be a specific item users can interact with. The object is semantically annotated by as set of URIs to any knowledge

[23] `http://www.linkedtv.eu/`

base. Using those connections to knowledge bases we represent the object by a set/bag of features: entities an associated types/classes: $\{URI_1, URI_1\_class_1, \dots\}$

Example 3.5.1. ***Bag of Entities***

Consider a textual fragment from Example 3.2.3: "Rocky is a 1976 American sports drama film directed by John G. Avildsen and both written by and starring Sylvester Stallone." Three entities (prefix: dbp) were recognized using a NER tool: dbp:Rocky, dbp:John G. Avildsen and dbp:Sylvester Stallone. Types associated to each entity (prefix: dbt) are dbt:Work, dbt:Movie, . . . ; dbt:Person; dbt: Director, . . . ; dbt:Person, dbt: Actor . . . . The BoE representation for this object is a set of all entities and types: $\{dbp : Rocky, \dots, dbt : Actor\}$

Preference model. *A model valid for one specific user or the entire group of users.* We represent those models using a set of rules learned from the data related to users' interactions. Each rule is in a format, where the left-hand side (LHS) of a rule consists of entries from *BoE* and the right-hand side (RHS) corresponds to the level of preference. Those rules are usually called *class association rules (CARs)* [145].

Example 3.5.2. ***Preference model***

Assuming the user provided positive interactions to the object from previous example and also to other objects (e.g. bookmarking or positive rating). We can apply a rule mining algorithm and infer the following subset of rules as a preference model: 1) $dbt : Movie \& dbp : SylvesterStallone \rightarrow interest\_level = positive$ *or 2)* $dbp : Rocky \rightarrow interest\_level = positive$.

3.5.2 Rule-based Semantic Preferences

This section provides more detailed description of the proposed approach to construct and apply user preference models. *BoE* representations are used as an important input for a rule learning algorithm.

3.5.2.1 Learning Preferences

The proposed method is based on an association rule learning. We follow the concept of *Association Rule Classification (ARC)* when we focus on *Classification Based on Associations (CBA)* - the first association rule classifier [146]. The first stage of the approach is a rule learning and the second stage is a usage of rules. So far, there were presented many algorithms for association rule mining. We use the well-known *Apriori* algorithm [56] that learns association rules from so called transactional databases. It uses a "bottom up" approach, where it starts with frequent individuals that are extended with one item at a time. Groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found (See Algorithm 8).

One entry of the transactional database is composed from the *BoE* representing the object the user interacted with together with the information about the preference level.

Algorithm 8: Apriori algorithm

input : A transactional database TD
Minimum confidence $conf$
Minimum support $supp$
output: A set of rules R

```
1  begin
2      // initialize
3      k = 1
4      // generate candidates of length 1 and filter by support
5      FI_k = generateFrequentItemsets(k, ∅)
6      FI_k = filterFI(FI_k, supp)
7      while existsCandidates(k+1) do
8          // generate candidates (length k+1) from candidates of length k
9          candidates = generateFrequentItemsets(k + 1,FI_k)
10         // remove candidates with support less than supp
11         FI_k = filterFI(candidates, supp)
12         k = k + 1
13     // convert all frequent itemsets to association rules
14     R = convertToRules(FI)
15     // filter according to the minimum confidence
16     R = filterRules(R, conf)
17     return(R)
```

Information about the preference level can be provided explicitly by a user or using any preprocessing algorithm (See Section 3.1 for more details). Both, the *BoE* for the object and the interest level are merged to entries acting as the transactional database TD that is used in the rule learning algorithm: $TD = \{BoE_1 \cup interest\_level_1, \dots\} = \{\{URI_1, URI_1\_class_1, \dots, interest\_level_1\}, \dots\}$. Since the interest level values can be real numbers, we use a three level discretization of interest levels $= \{negative, positive, neutral\}$. The values we transform from interval $[-1; 1]$, where zero represents *neutral*, values below zero are *negative* and values above are *positive* interests.

The association rule learning algorithm generally returns all possible association rules with various combinations of attributes on LHS and RHS of rules. For the purpose of preference learning we require only rules in the format of the *Preference Model* - interest level on the right-hand side. We also use a standard setting of rule learning algorithms: setting of the minimal confidence and the minimal support. The support is used during the construction of frequent itemsets within the algorithm and the confidence to filter generated rules.

Important aspect of the preference model is a size of the model and importance of individual rules within the model. To decrease the size of the model, while preserving

the quality of the model, we can use a concept of a rule pruning. The rule pruning removes rules that can be never used for subsequent classification, usually due to their redundancy, lower significance etc. We did not focus on the pruning during experiments with preference learning. However, we performed several experiments to demonstrate the signifance of pruning in evaluation of rule based recommender systems (Section 3.5.4). Brief introduction is presented in Section 3.5.4. Details on theory and research related to the rule pruning are out of the scope of this thesis. For more details about pruning we would like to refer the readers to our relevant papers [A.19]. To prefer certain rules over other, we sort rules in the same way as CBA [146] according to the confidence (decreasingly), support (decreasingly) and length of the left hand side of each rule (increasingly - shorter is better).

Algorithm 9 demonstrates the approach to learn a rule-based semantic preference model. First, the transactional database is created as a conjunction of the BoE and the interest level. Secondly, performing *Apriori* algorithm, we learn all possible rules. Afterwards, we include only *CARs* to the final preference model. Finally, we post-process the preference model using optional pruning step and sort the rules as the last step.

Algorithm 9: Rule-based semantic preference learning

input : Relations R of user with objects including interest level:
$R_n = object_o, interest\_level_i$
Rule learning settings: $confidence$, $support$

output: A user preference model P

```
1  begin
2      TD = ∅
3      // convert relations to a transaction form
4      for object_n, interest_level_n ∈ R do
5          TD = TD ∪ {buildBoE(object_n) ∪ interest_level_n}
6      // perform rule mining
7      rules = apriori(TD, support, confidence)
8      // filter rules
9      P = toPreferences(rules)
10     // post-processing - pruning (optional), sorting, ...
11     P = postprocess(P)
12     return(P)
```

Example 3.5.3. ***Preference model learning***

Let a user $U_{example}$ interacted with three different objects o_1, o_2, o_3 and expressed following interest levels (using explicit feedback or our method from Section 3.1): $\{(o_1, 0.4)$, $(o_2, 0.7)$, $(o_3, -0.5)\}$. *The BoE representations for objects are (prefix dbp for entities and dbt for classes):* $\{dbp : Hurling, dbt : Sport\}$, $\{dbp : Swimming, dbt : Sport\}$, $\{dbp : Republic, dbt : Forms\_of\_government\}$.

The transaction database TD is constructed as follows: { $\{dbp : Hurling, dbt : Sport, positive\}$, $\{dbp : Swimming, dbt : Sport, positive\}$, $\{dbp : Republic, dbt : Forms\_of\_government, negative\}$ }

For the rule learning algorithm we experimentally set the minimum confidence 0.3 and minimum support 0.3. The selected subset of rules in the preference model sorted by confidence, support, LHS size: 1) $dbt : Sport \rightarrow positive$, support=2/3, confidence=1 2) $dbt : Sport \& dbp : Swimming \rightarrow positive$, support=1/3, confidence=1 3) $dbt : Forms\_of\_government \rightarrow negative$, support=1/3, confidence=1 N) $\emptyset \rightarrow negative$, support=1, confidence=1/3

Please note that the second rule can be conditionally (with respect to the input data) removed applying the pruning since it has lower support compared to the first one and the first more general rule already incorporates the $dbt : Sport$.

3.5.2.2 Rule-based Personalization

Our proposed method acts as a classifier that uses a sorted list of rules representing preference model of a user or the entire group. For personalization purposes we assume that the list of candidates for classification is provided. The list is further labelled using the preference model representing the user.

Algorithm 10 demonstrates the approach we use for personalization. The first part of the algorithm is responsible for preparing *BoE* representations for each candidate. The second stage of the algorithm iterates over rules of the preference model. The first highest ranked rule whose LHS matches the object is selected, and its RHS is used to label the instance. The last condition is applied in situations when no matching rule is found. It does not happen in case the rule with empty LHS is available. It depends on the setting of rule mining or the optional CBA-based rule pruning provides such rule.

Example 3.5.4. ***Labelling Candidates***

Assume we have following two candidates c_1 and c_2. Their BoE descriptions are: $\{dbp : Hockey, dbt : Sport\}$ and $\{dbp : Boeing\_747, dbt : Aircraft\}$. The first candidate c_1 is labelled as positive, because the candidate is about Sport and the first rule from our preference model is applied. For the second candidate c_2, the last default rule is applied. It does not match any previous rule and it is labelled as negative. From the perspective of personalization or recommendation: the first candidate might be interesting to the user and the second one is not relevant.

Since labelling of all available objects is a time consuming operation, the disadvantage of the method is the requirement for a short list of predefined candidates that is supposed to be labelled independently. The method make sense mostly for situations when the list is available: e.g. the client side personalization or recommendation. The server side service should provide a generic list of candidates that is labelled on the client side using the preference model and presented to the user.

Algorithm 10: Rule-based Personalization

input : A preference model of a user: P
A set of candidates: C
output: A set of labels corresponding to the set of candidates L

```
1  begin
2      temp = ∅
3      L = ∅
4      // convert the representation to a transaction form
5      for object_n ∈ C do
6          temp = temp ∪ buildBoE(object_n)
7      // use classifier (sorted list of rules)
8      for candidate ∈ temp do
9          for rule ∈ P do
10             if match(rule, candidate) then
11                 L = L ∪ RHS(rule)
12                 break
13         if no match for candidate then
14             L = L ∪ neutral
15     return(L)
```

3.5.3 Experiments with Semantic Preferences

The focus of this section is on the evaluation of the semantic Bag of Entities representation used in the proposed approach of preference learning. Our main hypothesis is that the semantic, more representative and concise BoE representation can provide comparable or even better results in personalization tasks. Since the amount of datasets suitable for the evaluation of the BoE representation for our preference learning is limited, we cast the problem as a *text categorization* task. Essentially, there are three types of documents: those for which the user interest is known to be positive, negative and neutral.

The experimental setup aims at comparing the performance of the Bag of Words (BoW) representation with the BoE representation. The comparison is performed on two versions of the rule based classifier: $brCBA$ and $termAssoc$.

brCBA. The brCBA algorithm [A.19] is a simplified version of the seminal CBA algorithm [145] and corresponds to the proposed preference learning algorithm. Since the core of the implementation was also used for learning business rules [A.19], we prefixed the original acronym with *br*. The most important difference is that unlike CBA, it includes less or no pruning steps and outputs a partial classifier, which is simply composed of (pruned) rule set output by the association rule learner. As the rule learner we use the

well known apriori implementation in *R* - *arules*[24].

termAssoc. This setup is inspired by the ARC-BC (Associative Rule-based Classifier By Category) algorithm for text categorization by term association proposed in [147]. The algorithm was selected due to its previous successful evaluation and results on the selected dataset. When learning rules for a given interest level, the system takes into consideration only content fragments annotated with a given interest level. For each interest level, the system thus generates a separate list of frequent itemsets. These frequent itemsets are converted to rules predicting the current interest level. Finally, all rules for each interest level are merged.

It should be noted that while this step is inspired by the ARC-BC algorithm [147], there are some differences. In particular, ARC-BC uses a custom Apriori implementation, which redefines the support so that one transaction (document) can increase the support count by more than 1. In contrast, our *termAssoc* relies on the "mainstream" version of the association rule learning task (with standard support definition), for which multiple performance optimizations have been proposed.

3.5.3.1 Dataset

We use the ModApte version of the Reuters-21578 Text Categorization Test Collection [25], which is one of the standard datasets for text categorization tasks. The Reuters-21578 collection contains 21,578 documents, which contain the textual content and are assigned to 135 different categories (topics). Example topics are "earn" or "wheat". One document belongs on average to 1.3 categories. We use only a subset consisting of the documents which are assigned to ten most frequently populated categories as e.g. in [147]. Our dataset thus consists of 6,399 training documents and 2,545 test documents.

3.5.3.2 Preprocessing

The preprocessing is performed in two stages. First, the *BoW* or *BoE* feature sets are created from the underlying dataset. Then, depending on the classifier used, the term (concept) vectors are pruned.

BoW. All input documents contain 58,714 of distinct terms. To decrease the dimensionality, we performed the following operations: all terms were converted to lower case, numbers were removed, punctuation was removed, stop words were removed[26], whitespace was stripped and the documents were stemmed. The final document-term matrix contained 25,604 distinct terms.

[24] http://cran.r-project.org/web/packages/arules/

[25] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[26] A list of occurring 700 English stop words was used.

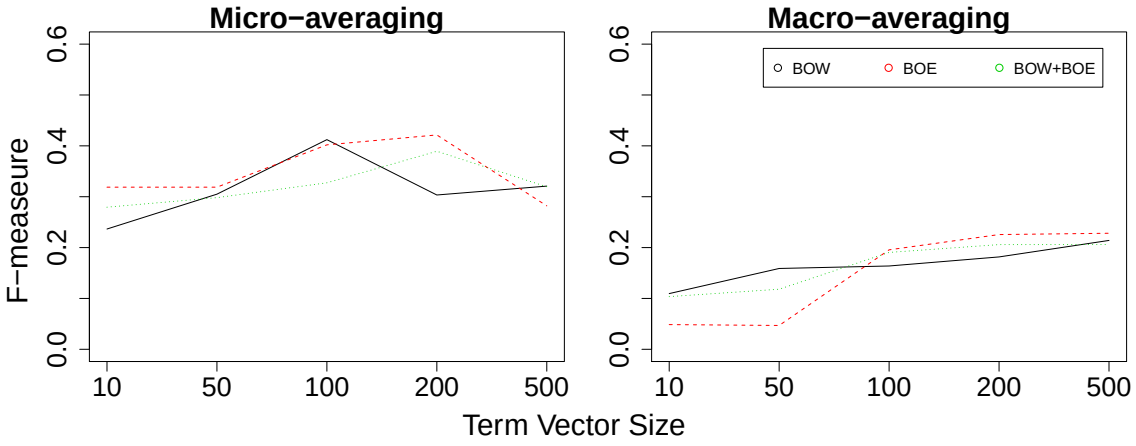


Figure 3.34: Results – brCBA

BoE. The Named Entity Recognition Tool [12] was used to annotate all documents. The service returned a list of entities (identified as DBpedia resources) and for each entity a list of the types (DBpedia Ontology concepts). The result of the preprocessing is a document-term matrix containing 12,878 unique concepts (entities and types).

Term (concept) pruning. The pruning is performed differently for *brCBA* and *termAssoc* algorithms. For *brCBA*, top N (*tvSize*) terms are selected according to TF-IDF. For *termAssoc*, term pruning is performed separately for each category using a TF score, selecting top N (*tvSize*) terms. Using TF-IDF scores with *termAssoc* degrades results in our observation, since terms with low IDF value (computed on terms within a given category) often discriminate documents in this category w.r.t. documents in other categories.

We also tried combining the *BoW* and *BoE* representations (denoted as BoW+BoE). For a given value *tvSize* parameter, 50% were top-ranked terms from BoW and 50% top-ranked concepts from BoE.

3.5.3.3 Rule learning setup

To perform experiments, we used the *minConf*=0.001 threshold, *minSupp*=0.001 for brCBA, and *minSupp*=0.2 for termAssoc. The maximum rule (frequent itemset) length was unrestricted.

The training subset of Reuters-21578 corpus is partitioned into ten folds, each fold corresponds to one category in the dataset. All documents belonging to the current category are marked as being of "positive" interest to the user; the remaining documents are marked as being of "negative" interest. In this way, we obtain ten hypothetical users, each interested in one specific topic and disinterested in everything else. The "neutral" interest that would correspond to documents without any topic is not considered since documents

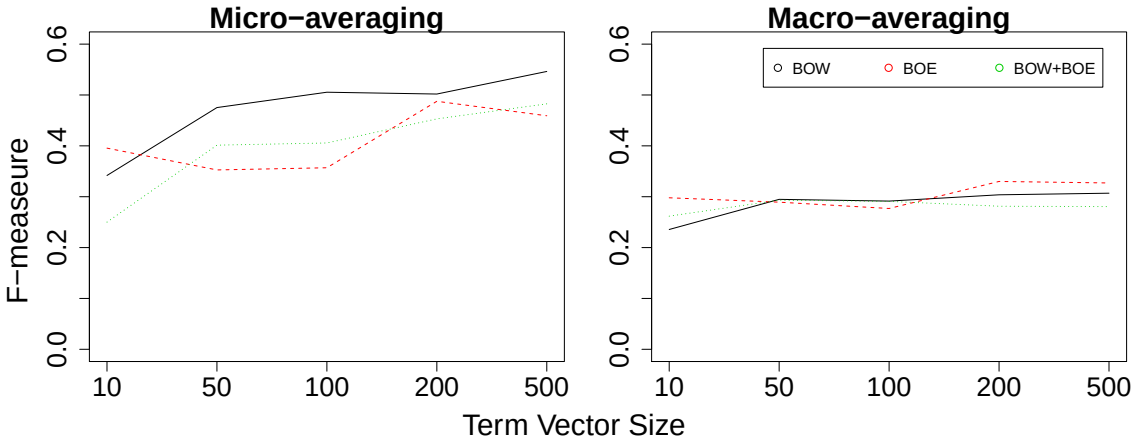


Figure 3.35: Results – termAssoc

without any topic have been removed to foster comparison of our results with previously published works.

brCBA. The rule learning is performed on all documents, outputting all rules that match the predefined minimum support (*minSup*) and minimum confidence (*minConf*) thresholds.

termAssoc. The learning is performed separately for each of the ten categories using only documents in current category. The learning has *minSup* as the only parameter. The output for a given category is a list of frequent itemsets. These frequent itemsets are converted to rules predicting the current category.

3.5.3.4 Results

Results are reported in terms of micro-average and macro-average F-measure (refer to [148] for details). The important conclusion is that BoE is suitable for preference learning/text categorization tasks. Even with significantly lower dimensionality of a source vector space model (25,604 for BoW vs 12,878 for BoE), we can achieve comparable results. BoE can significantly reduce the amount of data needed for representation of content items while results remain at the same level.

The results, depicted on Fig. 3.34- 3.35, indicate that for the smallest term vector size, BoE representation yields better overall F-Measure than the BoW representation. Also, the best overall results are provided by the termAssoc algorithm.

Surprisingly, the fusion of BoW and BoE into one term vector is dominated by the performance of the BoW/BoE alone.

It should be noted that significantly better results than we have achieved on Reuters-21578 are reported e.g. in [148], also the relative improvement provided by the BoE rep-

resentation is only marginal, pointing at the need to perform more research into generating the BoE feature set.

3.5.4 Context-Aware Item Recommender Modification

In the previous section we proposed the rule-based semantic preference models acting as classifiers for labelling of unseen objects. We mainly focused on the evaluation of BoE as the semantic representation of objects. In this section we focus on the evaluation of rule-based classifiers in the domain of recommendations based on implicitly performed usage data.

To better fit the recommender scenario, we performed following modifications of the previously described approach. The transactional database is composed from a sequence of contextual features related to the relation between a user and an object (e.g. a location, time of the day etc.), object identifier and optionally BoE of the object. One entry of the Transactional Database TD is thus in the following format: $\{contextual\_features \cup BoW_{object\_id} \cup object\_id\}$. Rule learning is modified to return rules in format where the LHS contains $\{contextual\_features \cup BoW_{object\_id}\}$ and the RHS contains the $\{object\_id\}$. Classification step is not limited to return only one RHS of the best matching rule, but it is intended to return top-L unique RHS of matching rules. The goal is to provide a list of *object_id*s based on a set of contextual features. The list is then used as the set of recommended objects for users with specific contextual features. Such association is internally represented as a rule based classifier.

Example 3.5.5. ***Item Recommendation***

Let two users (u_1, u_2) interacted with three objects (o_1, o_2, o_3). The first user, who interacted in the morning with objects o_1 and o_2, is from $location_1$. The second user interacted in the evening with o_2 and o_3, he is from the same location.

The transaction database TD is constructed as follows: $\{$ $\{morning, location_1, o_1\}$, $\{morning, location_1, o_2\}$, $\{evening, location_1, o_3\}$, $\{evening, location_1, o_2\}$ $\}$

The example of the extracted rules subset representing the learned model sorted by confidence, support, LHS size: 1) $location_1 \rightarrow o_2$, support=1, confidence=1/2 2) $location_1$ & $morning \rightarrow o_1$, support=1/2, confidence=1/2 N) $\emptyset \rightarrow object_3$, support=1, confidence=1/4

For another user from the same location in the afternoon, we can use the model. The rules that match the user are 1) and the last default one N). The algorithm recommends objects extracted from RHS of matching rules: o_2, o_3.

3.5.5 Evaluation of the Rule-based Recommender System

In this section, we evaluate the proposed algorithm in the context of our participation in the International News Recommender Systems Challenge[27] and CLEF-NEWSREEL: News

[27] `https://sites.google.com/site/newsrec2013/challenge`

Recommendation Evaluation Lab[28] (further only Challenge), where we achieved 2nd and 3rd place in the overall evaluation. They both are focused on recommending news articles in real-time.

3.5.5.1 On-line task: Setup and Results

The Challenge is focused on recommending news articles in real-time. The crucial criteria of the Challenge are placed on the quality of computed recommendations together with scalability of participating solutions (peak load up to 100 messages per second) and response time limitation. Recommendations had to be provided in real-time (within 100 ms), and the winning criterion was set to the total number of successful recommendations, rather than the prediction accuracy (clickthrough rate). The successful recommendation is identified when a user selects an article from the list of recommended items. There are practical problems with real time processing of recommendations that are not incurred when there is "unlimited time" to provide the recommendation. It is necessary to balance the architecture and technologies with the complexity of the involved algorithms.

Task Definition

This section describes a simplified definition of the news recommender task.
Inputs: The main inputs are users' interactions and news item descriptions.

- $interaction(type, userId, itemId, context)$
  where $type = \{impression|click\}$ and $context$ describes the features of the user (e.g. browser version, geolocation, etc.) and special features related to items and their presentation (e.g. keywords, position).

- $item(itemId, domain, description)$
  where $domain$ is the identifier of items from the same group (e.g. news portal) and $desciption$ provides more detailed information about items (e.g. title, text, time of last update).

Outputs: Set of recommended items for the specific user who is reading the item within a given context.

- $(userId, itemId, context) \rightarrow \{item_x, item_y, ...\}$

Algorithms

In this section we describe a set of algorithms we used in the Challenge. First two algorithms are baselines to compare the quality of the proposed approach.

[28] `http://www.clef-newsreel.org/`

Top Interacted. This algorithm is based on the daily popularity of news items. To avoid excessive effect of high short-time popularity of one item the interactions are aggregated on a daily basis. This approach deals with an evolution of popularity over time and decrease an influence of peaks appeared at specific days. We implemented the algorithm using simple incremental updates in a collection represented as a triple $(Date, ItemId, count)$. The result is the list of items sorted by the number of interactions.

Most Recent. Since we are in the highly dynamic news domain, the recency of an article plays an important role. Our baseline recency-based algorithm uses a simple heuristic based on the newest news item within the same group as the group of the item the user is reading at the time of the request. The results is the ordered list of items sorted by creation time.

Table 3.21: Training dataset for rule based recommender. Values are anonymized by the organizer of the challenge.

Context							Class
browser	isp	os	geo	weekday	lang	zip	item
312613	281	431229	19051	26887	49021	62015	127563250
457399	45	952253	18851	26887	48985	65537	45360072

Rule Based. For each $interaction(type, userId, itemId, context)$ stored in our database, we prepared one entry in the training dataset as described in Table 3.21. Interactions are described only by the contextual features that are provided by the platform (e.g. Location, Browser, ...) and by an identifier of item the user interacted with.

The training dataset was used to learn association rules. The contextual features could appear only in the rule body (LHS) and the identifier of the item only on the right side of rule (RHS). We used association mining algorithm *apriori* implemented in *R - arules*[29]. Example of a rule:

$$\text{isp} = \text{“281”} \ \wedge \ \text{os} = \text{“431229”} \ \rightarrow \ \text{item} = \text{“1124541”}$$

Additional mining setup is as follows. We used latest N thousands interactions as the training dataset from our database. We experimentally set N to five thousands. The apriori algorithm is experimentally constrained with minimal support of five interactions, and minimum confidence of 0.2. We regularly replaced the current set of rules every 30 minutes with rules learned on the continuously updated latest N interactions.

All discovered rules are imported into our simple rule engine acting similarly to the labelling in 3.5.2.2. The engine finds all rules that match contextual features of a recommendation request. The RHS of each matching rule represents a recommended item. The output is a list of unique item identifiers from the right side of the matching rules.

[29]http://cran.r-project.org/web/packages/arules/

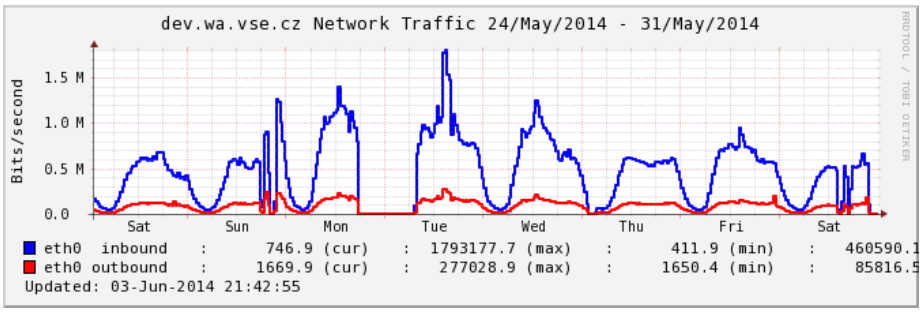


Figure 3.36: Network Traffic on the server - evaluation period (2014-05-25 – 2014-05-31).

Table 3.22: Leaderboard with cumulative number of clicks and average click-through rate per team in the Challenge - last evaluation period (2014-05-25 – 2014-05-31). Source: http://orp.plista.com

team	requests	clicks ↓	CTR
labor	285533	5614	1.97%
abc	206330	3653	1.77%
inbeat	**268611**	**3451**	**1.28%**
insight	508851	2012	0.4%
ba214	158593	1828	1.15%
uned	370510	1215	0.33%
riemannzeta	99920	1156	1.16%
plista GmbH	9112	137	1.5%

3.5.5.2 Performance

In this section, we present the performance of recommender in the Challenge.

The metrics used in the Challenge to select the winning recommender systems was the *cumulative number of clicks* (number of successful recommendations) over the three different evaluation periods. The additional metrics provided by the organizers include daily reports on *number of impressions* and *click-through rate*.

Sum of the number of impressions with the number of clicks can be interpreted as the performance of the systems – the ability to process large number of interaction on the server.

Figure 3.36 shows the network traffic on our server infrastructure within the last evaluation week of the Challenge. During this period, our solution handled thousands of recommendation requests. The peaks in the graph correspond to the higher number of interaction in daytime. Note that the gap between Monday and Tuesday is caused by the maintenance break of our infrastructure. The server load was kept mostly under ten percent even in peak periods. Implementation was run on a single virtual machine assigned four Core i7@3.20GHz cores and 8GB of RAM.

Table 3.22 presents the results for the last evaluation period. The table is sorted by

the cumulative number of clicks. InBeat (our) team is on the third position. The table provides only results that are aggregated per team participating in the Challenge. There are no specific results for each recommendation algorithm. In click-through rate, the second metric, InBeat is on the fourth position.

Since the CTR reported in Figure 3.22 is the average for all algorithms, we also report the numbers for the individual algorithms:

- *Top Interacted* has 1.4% CTR,
- *Most Recent* has 0.8% CTR,
- *Rule Based* has 1.5% CTR.

The most successful algorithm is *Rule Based*, which we explain by the fact that it takes into account both popularity and contextual features. *Most Recent* is influenced only by temporal aspects and *Top Interacted* takes into account only the popularity.

3.5.5.3 Off-line task: Setup and Results

Since the on-line evaluations point out the quality of the rule based algorithm, the objective of our experimental off-line evaluation is to investigate the performance of Association Rule Classification (ARC) algorithms in the recommender problem casted as a standard classification task. Secondly, we compare the results with related mainstream classification algorithms.

Data and task

We used the data published within the off-line task of CLEF-NEWSREEL'14 Challenge. The entire dataset consisted of 84 million records collected across multiple news portals [149]. We selected the website with the smallest amount of data (26,875 records) denoting the resulting dataset as CLEF#26875.

The dataset consists of instances described by a fixed number of attributes. In our evaluation we process the data with standard machine learning algorithms that require data in tabular form.

The task is to predict the class label (item viewed). The CLEF#26875 off-line dataset has 1,704 distinct items (target class values). This is an unusually high number in comparison with other datasets typically used for evaluation of machine learning algorithms, such as the most frequently cited datasets from the UCI repository.[30] This distributional characteristic has an impact both on execution time and accuracy of the evaluated algorithms. The second notable feature of the dataset is that all its attributes are nominal. This is a favourable property for ARC algorithms in general, since they typically require that numerical attributes are discretized prior mining. The discretization algorithm and its parameters may have substantial impact on both accuracy and execution time.

[30] `https://archive.ics.uci.edu/ml/datasets.html`

The problem is cast as a standard machine learning classification task, where each row corresponds to a separate training instance. We also provide comparison with related mainstream machine learning algorithms that create rule or tree-based models (decision trees are convertible to rules).

Algorithms

The main focus of our evaluation is the Classification Based on Associations (*CBA*) ARC algorithm [145] and its two candidate successors – *CMAR* [150] and *CPAR* [151]. We compare the results with related symbolic machine learning algorithms, namely rule induction (*FOIL*, *CPAR*) and decision tree algorithms (*ID3*, *CHAID*).

The primary difference between ARC algorithms and rule induction is that the former class of algorithms first generates all association rules in the training data, and then performs pruning, while the rule learning algorithms add rules to the model one-by-one. The *CPAR* algorithm has some features of both ARC and rule induction algorithm, we list it under rule induction.

Association Rule Classifiers

In 1998, Liu et al. introduced *CBA*, the first association rule classifier according to [152]. The first step in *CBA* is association rule learning with a modified *apriori* algorithm. The learning is constrained to produce rules that have an item corresponding to a class label value in the RHS.

In the second step, the resulting rules are subject to several pruning algorithms:

1. Pessimistic pruning (optional). This pruning method attempts to simplify discovered rules by removing individual conditions from the rule LHS. The rule is pruned if the pessimistic error rate [153] of the original rule is higher than that of the pruned rule.

2. Data coverage pruning[31]. This method removes rules preserving the following two conditions: i) each training case is covered by the rule with the highest precedence over other rules covering the case and ii) every rule in the classifier correctly classifies at least one training case.

3. Default rule pruning[32]. Rules pruned with data coverage pruning are ordered and all rules after the first rule with the lowest total error are replaced by a rule with empty antecedent predicting the majority class in the remaining data.

The gist of the *CBA* algorithm are the latter two pruning methods. The final ordered rule set is used as the classifier. Rules are sorted according to confidence, support and antecedent length. *CBA* performs single rule classification: for a given unlabelled instance,

[31]We adopt the name for this method from [152].

[32]This pruning type is omitted from the review [152], but we are of the opinion that "default rule pruning" could be perceived as a separate step from data coverage pruning.

the first highest ranked rule whose LHS matches the instance is selected, and its RHS is used to label the instance.

The *CMAR* algorithm is based on similar principles as *CBA*, but uses the newer FP-Growth [154] algorithm for association rule generation. In addition to data coverage pruning, *CMAR* performs also pruning based on chi-square test. The rule is pruned if the correlation between the rule's LHS and and the rule's RHS is not statistically significant. The data coverage pruning in *CMAR* is slightly different from *CBA* as it requires at least δ rules to cover an instance before the instance is removed from training data (in *CBA*, $\delta = 1$).

In our benchmarks, we used the LUCS-KDD implementations of the ARC algorithms[33]. According to the implementations' author the software matches the description in the original papers introducing the respective algorithms, apart from that in the first rule generation step, the Apriori-TFP algorithm [155] is used instead of the modified *apriori* algorithm (*CBA*) or *FP-Growth* (*CMAR*).

It should be also noted that the LUCS-KDD implementation of *CBA* does not include pessimistic pruning. In evaluations on 20 UCI datasets reported in [145] *CBA* with pessimistic pruning had exactly the same accuracy as *CBA* without pessimistic pruning, but order of magnitude smaller number of rules in the classifier.

For part of the experiments with *CBA*, we used our own implementation of *CBA*. While this is not as efficient as the LUCS-KDD implementation, this allows us to test the effect of the individual pruning stages in *CBA* on accuracy and rule count of the resulting classifier. For rule generation phase, our implementation uses the *apriori* algorithm from the arules package followed by a filtering step which retains only rules that have one of the class labels in the consequent. For the rule generation phase we implemented both version M1 and M2 of *CBA* [145]. The most simplified form of the classifier has a learning phase roughly corresponding to the execution of the *apriori* algorithm.

Rule learning (baseline)

As a second set of baseline algorithms, we selected the First-Order Induction Learner (*FOIL*) [156] and the Classification based on Predictive Association Rules (*CPAR*) algorithm. It was shown that *FOIL* is prone to overfitting the training data as the size of the theory learned by FOIL can grow with the number of training examples [157]. For this reason, we tried to include Repeated Incremental Pruning to Produce Error Reduction (*RIPPER*) [158] algorithm, which effectively addresses the overfitting problem [159]. We did not include *RIPPER*, because on the CLEF#26875 data the RapidMiner 5 implementation[34] of the algorithm did not finish within a 12 hour time limit.

Finally, *CPAR* was designed to combine advantages of rule learning algorithms with association rule classifiers. The algorithm tests more rules than traditional rule-based classifiers which is claimed to ensure it does not miss important rules.

We used again the LUCS-KDD implementation of *FOIL* and *CPAR*.

[33] `http://cgi.csc.liv.ac.uk/~frans/KDD/Software/`

[34] `http://sourceforge.net/projects/rapidminer/`

Decision trees

Decision tree induction algorithms produce models that to an extent resemble those produced by ARC algorithms. Each path from the root of the tree to the leaf in a decision tree corresponds to a classification rule.

Out of the multiple proposed decision tree algorithms, we included those implemented in the RapidMiner 5 open source data mining suite: *ID3*, RapidMiner's "*Decision Tree*" and *CHAID*.

ID3 [160] is a frequently used baseline decision tree algorithm. Since all input attributes in CLEF#26875 are nominal, the algorithm can be used directly on input data without any preprocessing.

The RapidMiner's *Decision Tree* operator was found to be the most accurate decision tree classifier in [161], which evaluated decision tree learning algorithms in three common data mining suites: SPSS-Clementine, RapidMiner and Weka. This implementation supports prepruning and postpruning methods.

The RapidMiner's *CHAID* implementation uses the chi-square test as a goodness criterion, otherwise it is the same as *Decision Tree*.

Experimental evaluation

The algorithms described in the previous subsections were executed with parameters set according to Table 3.23.

Table 3.23: Algorithm parameters used in the off-line evaluation.

method	**parameters**
CBA	support = 2 records (0.008%), confidence = 2.0%, max size of antecedent = 6, max number of CARS = 80000, max number of frequent sets = 1,000,000
CMAR	support = 2 records (0.008%), confidence = 2.0%, max size of antecedent = 6, min cover (δ) = 1
CPAR	*default values*: K value = 5, min. best gain = 0.7, total weight factor = 0.05, decay factor = 1/3, gain similarity ratio = 0.99
Decision Tree, CHAID	*default values*: criterion = gain ratio (Decision Tree), Chi-square test (CHAID), minimal size for split = 4, minimal leaf size = 2, minimal gain = 0.1, maximum depth = 20, confidence = 0.25, no prepruning, postpruning enabled
ID3	*default values*: criterion = gain ratio, minimal size for split = 4, minimal leaf size = 2, minimal gain = 0.1
FOIL	max number of attributes per rule = 6

The support and confidence parameters of *CBA* and *CMAR* had to be changed from the default values (of 20% and 80% respectively), since otherwise no rules were generated

(no class item in the data had at least 20% support). The maximum number of frequent sets for *CBA* and *CMAR* was increased to 1,000,000 since for support threshold lower then approximately 0.01%, the default limit of 500,000 prevented further improvements of the classifier. For *DecisionTree*, we initially obtained very low accuracy of 2%. This was caused by the prepruning step, which is enabled in RapidMiner by default. The resulting tree was composed of only one leaf class, which is the most frequent class label in the training data. The (post)pruning feature had a small but positive impact on accuracy and model size, therefore we left it enabled. For *CPAR* the default parameters produced acceptable results. Additional parameter tuning could have improved the performance of the algorithm.

The data were preprocessed to the form shown at Table 3.21 and randomly split to a training dataset (90%) and test dataset (10%). The experiments were run on Intel core i5 3320M CPU@2.6 GHz with 16 GB of RAM.

Table 3.24: Model benchmark on CLEF#26875 dataset (single 90/10 split). Model size refers to the number of rules for rule models and number of leaves for decision trees. Time is measured in seconds.

	time			
algorithm	**train**	**test**	**accuracy**	**model size**
DecisionTree	273	4	23.0	13496
ID3	290	4	22.8	13579
CHAID	284	3	**25.4**	13224
FOIL	815	1.5	24.7	18047
CPAR	**87**	1.23	4.6	18907
CBA	279	**0.25**	21.2	**3681**
CMAR	205	1.781	16.9	22516

Table 3.25: Effect of support threshold - CBA (ten-fold shuffled cross-validation). Time is measured in seconds.

metric	**0.10%**	**0.09%**	**0.08%**	**0.07%**	**0.06%**	**0.05%**	**0.04%**	**0.03%**	**0.02%**	**0.01%**
accuracy	6.68	6.88	7.07	7.64	8.1	8.65	9.48	10.4	13.47	17.55
train time	1.8	2.3	3	4.56	5.6	8.7	14.6	30.5	172	477
test time	0.02	0.03	0.03	0.04	0.03	0.04	0.05	0.05	0.1	0.19
rule count	148	178	193	228	270	317	452	576	1100	2303

The results depicted in Table 3.24 indicate that the overall best accuracy was obtained by the *CHAID* decision tree algorithm. *CBA* obtained accuracy close to the decision tree classifiers, however, with smaller training times and - for the on-line setting most significantly - shorter testing times. There are several factors contributing to the fast testing: a) the fact that *CBA* performs single rule classification, b) small number of rules in the classifier (compared to models created by other algorithms). The difference in test

times between decision trees and the rule learning algorithms might be to a large extent caused by implementation-specific issues. Our impression is that additional optimization for the evaluation of the decision tree models could lead to substantially shorter test times.

Trading speed for accuracy

Speed of training can be important in on-line recommender setting. Fast training also typically entails simpler models that are faster to apply. The accuracy/execution time balance can be controlled by the minimum leaf size and/or maximum depth parameters for decision trees and by the minimum support parameter for ARC classifiers.

Table 3.26: Effect of support threshold - CMAR (ten-fold shuffled cross-validation). Time is measured in seconds.

metric	**0.10%**	**0.09%**	**0.08%**	**0.07%**	**0.06%**	**0.05%**	**0.04%**	**0.03%**	**0.02%**	**0.01%**
accuracy	4.82	5.12	5.28	5.78	6.12	6.59	7.48	8	10.23	13.84
train time	0.744	0.89	1	1.39	1.75	2.13	3.83	6.5	36.34	178.92
test time	0.11	0.115	0.14	0.144	0.18	0.2042	0.32	0.46	1.05	2.26
rule count	834	999	1177	1557	1863	2251	3581	5116	11450	20561

Table 3.27: Effect of minimum leaf size - ID3 (ten-fold shuffled cross-validation, *based on one 90/10 split). Time is measured in seconds.

metric	**100**	**90**	**80**	**70**	**60**	**50**	**40**	**30**	**20**	**10**
accuracy	13.67	13.89	14.1	14.4	14.7	14.9	15.3	16.2	17	18.7
train time	8.58	8.58	9.04	9.57	10.57	12.17	13.93	18.09	25.4	80.66
test time	2.36	1.41	1.36	1.35	1.34	1.43	1.29	1.3	1.28	3.9
number of leaves*	3278	3362	3427	3522	3708	3959	4167	4817	5596	7389

Tables 3.25 and 3.26 show the impact of varying the support threshold on the accuracy and execution time of the *CBA* and *CMAR* classifiers. To obtain more reliable estimates especially at higher support thresholds, we performed ten-fold cross-validation. Table 3.27 shows the impact of minimum leaf size on the ID3 results.

The comparison between *ID3* and *CBA* at 13% accuracy level shows that *ID3* has much shorter training time (8.58s vs 172s), but it also produces more complex models (3278 leaf nodes vs 1100 rules for *CBA*). The more compact model size contributes to fast test times for *CBA*.

Optimizing CBA

In the field of decision tree induction, one of the mainstream pruning techniques is reduced error pruning, which uses different sets of data for learning the classifier and for pruning. Our experiments with *CBA* on CLEF#26875 showed that dividing available training data

into a training set and a holdout set for pruning (validation data) does not have a positive effect on classifier accuracy. We tried multiple ratios of training set/holdout set size without obtaining a notable increase in accuracy.

An interesting finding follows from results presented in Table 3.28: if only part of the data used for the rule learning phase (i.e. *apriori* in *CBA*) is used for the pruning phase (i.e. data coverage and default pruning in *CBA*), the impact on accuracy is small. The training time can be reduced substantially as smaller amount of data is processed.

Table 3.28: Effect of pruning data set size. 100% of training data were used for rule generation, only x% used for pruning. For this experiment, we used our implementation of CBA M1.

metric	**1%**	**2%**	**5%**	**10%**	**20%**	**30%**	**50%**	**75%**
rule count	38	48	78	96	125	138	151	166
accuracy [%]	4.5	5.6	6.6	6.8	7.1	7	6.9	6.9

Table 3.29: Impact of pruning steps in CBA. Minimum support set to 0.1% and minimum confidence set to 2%.

algorithm	**accuracy**	**rules**
no pruning, direct use of association rules	6.4	1735
data coverage pruning	6.9	497
data coverage, default rule pruning	7	175

The results of the experiments with omission of individual pruning steps from *CBA* (Table 3.29) indicate that both data coverage pruning and default rule pruning not only reduce the size of the rule set, but also slightly improve the accuracy of the model. Interestingly, the absolute difference in accuracy between direct use of association rules (as in the on-line challenge) and *CBA* is very small. However, the order of magnitude decrease in the number of rules in the classifier justifies the use of *CBA* in on-line setting which puts emphasis on fast prediction times.

Summary

Taking into account the success of rule based algorithm in on-line challenges, we evaluated the proposed rule based recommender system using off-line setting in order to elaborate on influence of different settings. The experiments performed on the off-line dataset indicate that the *CBA* association rule classifier can further improve the results in terms of accuracy and especially speed, as it significantly reduces the size of the rule set. Since pruning steps can significantly reduce the size of models, they are also efficient in aspects of time requirements for recommendations.

We further investigated the options for optimizing the pruning workflow in the *CBA* algorithm. The results indicate that the primary effect of the *CBA* pruning is the reduction of the number of rules in the model and that the impact on classifier accuracy is small. However, the potential saving in training time resulting from omission of these pruning steps might be offset by the increase of prediction time due to increased model size. Experiments showed that a viable direction of training time optimization might be using only part of the available training data for pruning.

Our benchmark on the off-line dataset was methodologically limited with respect to the typical setting for evaluation of recommender algorithms a) by ignoring the temporal dimension associated with the instances in the dataset and b) by providing results in terms of accuracy. Since recommender systems are frequently used as rankers other evaluation metric than accuracy could be more suitable. Future work could thus aim at addressing these limitations.

3.5.6 Details on Implementation

The method is available in two separate implementations. Two modules of *InBeat*: *Preference Learning* and *Recommender System* are implemented mainly in Node.js and available as an open source on GitHub [35].

The second implementation is a module for *R*. It provides methods for building a CBA classifier and is also available as an open source on GitHub [36]. The module is also available in official R repository[37].

3.5.6.1 InBeat - Preference Learning

The *Preference Learning* module of *InBeat* builds a recommendation model for each user. The current version implements association rule learning, but this can be also substituted by any standard learner accepting tabular data.

InBeat was implemented as a service that exposes the RESTfull API for any client. The implementation of the service is in Node.js. Underlying rule learning algorithms are implemented either in pure JavaScript or the baseline *arules* implementation from *R* is consumed. The details on the exposed API and several examples are in the on-line documentation.

3.5.6.2 InBeat - Recommender System

The *Recommender System* module of *InBeat* executes the model created in the *Preference learning* module providing a list of candidate recommendations associated with a confidence value. The current implementation uses one rule classification following the CBA

[35] `https://github.com/KIZI/InBeat`

[36] `https://github.com/jaroslav-kuchar/rCBA`

[37] `https://cran.r-project.org/web/packages/rCBA/index.html`

algorithm [145], allowing for easy explanation. This module is implemented in pure JavaScript without any external library. The details on the exposed API and several examples are in the on-line documentation.

3.5.6.3 rCBA

We implemented the module *rCBA* for widely-used and well-known *R* language. The package is partially a wrapper around our Java implementation of the CBA classifier.

Listing 3.11 demonstrates the usage of the module in *R*. The first part performs the rule mining and in the second part the CBA classifier is created using post-processing and pruning. Afterwards, the build classifier can be used for labelling of new instances.

```
# load libraries
library("arules")
library("rCBA")

# read data
train <- read.csv("./train.csv",header=TRUE) # read data

# train
txns <- as(train,"transactions") # convert
rules <- apriori(txns, parameter = list(confidence = 0.1, support= 0.1,
    minlen=1, maxlen=5)) # rule mining
rules <- subset( rules, subset = rhs %pin% "y=") # filter
rulesFrame <- as(rules,"data.frame") # convert
print(nrow(rulesFrame))

# pruning
prunedRulesFrame <- pruning(trainData, rulesFrame, method="m2cba") # m2cba(
    default)|m1cba|dcbrcba
print(nrow(prunedRulesFrame))
```

Listing 3.11: rCBA usage example.

3.5.7 Discussion

Client side recommendation. An important performance factor when devising client-side recommender systems is the size of the feature set. We performed experimental validation of the Bag of Entities (BoE) representation on a standard dataset. Our hypothesis was that the BoE representation provides better accuracy at a given term vector size than the standardly used Bag of Words (BoW). Experimental evidence obtained on Reuters-21578 text categorization collection suggests that the BoE representation can yield indeed slightly better results (F-Measure) with very small term vector size, although the increase is not as large as we have hoped for.

Rule-based Recommender System. Although there exists many recommender systems and suitable algorithms. The on-line track of the challenge required the competing systems to balance the architecture and technologies with the complexity of the involved algorithms.

The practical experience that we obtained with our recommender system underpin the choice of association rules as a fast on-line recommender algorithm. The experiments performed on the off-line dataset indicate that the *CBA* association rule classifier can further improve the results in terms of accuracy and especially speed, as it significantly reduces the size of the rule set.

Effects of pruning. We further investigated the options for optimizing the pruning workflow in the CBA algorithm. The results indicate that the primary effect of the CBA pruning is the reduction of the number of rules in the model and that the impact on classifier accuracy is small. However, the potential saving in training time resulting from omission of these pruning steps might be offset by the increase of prediction time due to increased model size. Experiments showed that a viable direction of training time optimization might be using only part of the available training data for pruning. Further decrease in the number of rules could be attained by applying pessimistic pruning, an optional step in CBA, which was not covered in our evaluation.

3.5.8 Summary

The contribution of this section focuses on preference learning and recommendations. We designed a semantic preference model that uses rules as an underlying concept. Both, the learning stage and application of the model use well-know algorithms. The main benefit of the proposed approach is in the possibility of learning well understandable, explainable and justifiable preferences and recommendations while taking into account the semantics. Most of existing solutions use various models as an underlying concept. Those models act as black box solutions, since they use complex learning algorithms. Since rules are considered as the most expressive form of encoding preferences, our approach is build on top of rule representations and rule learning algorithms. Second advantage of the proposed method is the possibility to use such model in client side modelling. The semantic annotation (Bag of Entities representation) reduces the amount of information that is needed for modelling, while preserving the quality of results of modelling. The model can be built directly on the client side, no information is send to any server. It can thus preserve the user privacy.

The rule-based modelling can be also used for recommendation. In our experimental evaluations and during participation in several challenges we have proved the suitability of the model in such scenarios, especially with focus on news recommendation tasks. We have also presented ideas for modifications of presented methods that can improve the time complexity of modelling.

Chapter 4

Conclusions

In this chapter we conclude the dissertation thesis by summing up main achieved results and contributions. We also present an outlook on further research directions for feature work.

4.1 Summary

In this dissertation thesis, we present theory, algorithms, evaluations and proof-of-concept implementations in the domain of the Web Usage Mining. We contributed with several novel approaches for building and utilizing rich semantic representations connecting users and content items. The overall methodology presents phases from data acquisition over semantization and enhancement to utilization of rich semantic representations.

Although there is an abundance of proprietary approaches and algorithms for the data acquisition in the Web Usage Mining, they are domain specific and the outputs are typically unsuitable for direct processing by mainstream machine learning algorithms and tools. One important reason is that interactions performed by individual users tend to be of irregular length. Modern rich interfaces or interactive web applications also allow to perform multiple interactions per one content item. Our method for the aggregation of multiple user interactions into one unified relation between a user and a content item addresses such issues. It provides unified outputs that are processable by conventional algorithms and tools.

To overcome the so-called "Semantic gap" between individual content items we adopted the semantization as a key concept to link each content item to an existing knowledge base. Such connections in form of URI references allow an extraction of additional features and to properly represent content items. Additional features are important to interconnect all information. We present an experimental domain specific linking method of movie titles to the DBpedia knowledge base using a set of ad-hoc queries. Since multiple links to a knowledge base can be provided, we also propose an aggregation step to overcome issues with conflicts or overlaps of multiple features. The output of the overall semantization is a semantic representation of each content item.

Most of existing knowledge bases or repositories, that are used for the linking, incorporate humans in its creation procedure. Thus, fragments of links within a representation can be missing. In order to get rich semantic representations, an automatic management of the knowledge base using link predictions helps to overcome issues with missing links. Despite many existing link prediction algorithms, there is still lack of approaches that are focused on multiple types of links (as they are in semantic representations) and time information about existence of links. Links that were included in the past may loose their significance in the future. We designed a novel approach for a link prediction incorporating temporal information. The evaluations and experiments show that the temporal dimension influences the results and it is an important aspect for the link prediction.

Rich semantic representations allow to utilize their relations and perform an intelligent selection of resources based on existing relations. For the same reason as in the link prediction, the temporal information plays an important role in our novel approach for personalized selection of entities within rich semantic representations. The method exploits relationships among entities, and social relationships among users such as who knows who; it takes into account users preferences such as users the user knows and preferences that define importance of specific link types, and it takes into account temporal information about links. We evaluate it on several experiments showing that the method gives better results over traditional popularity-based recommendations.

We also focused on preference learning algorithms and semantic-aware recommendations as the utilization of available rich semantic representations. Our research is focused on well understandable, explainable and justifiable preferences and recommendations while taking into account the semantics. Most of existing solutions use various models as an underlying concept. Those models act as black box solutions, since they use complex learning algorithms. Since rules are considered as the most expressive form of encoding preferences, our approach is build on top of rule representations and rule learning algorithms. In our experimental evaluations and during participation in several challenges we have proved the suitability of the method in several scenarios, especially in news recommendation tasks.

4.2 Contributions of the Thesis

The main contributions of this thesis are as follows:

- Method for an aggregation of semantically enriched user interactions.
- Algorithm for linking content to a public knowledge base and a method for semantic aggregation.
- Link prediction method that allows enhancement of semantic representations with respect to temporal information.
- Method for selection of the most relevant target among a predefined set of candidates.

- Preference learning and recommendation technique profiting from semantic annotations.

4.3 Future Work

The author of the dissertation thesis suggests to explore the following:

- Although we addressed issues of modern web interfaces that provide multiple interactions, our solutions are limited in usage of semantics. Either hand-written rules or proposed learning algorithms do not fully consider semantic information about individual interactions or content items. We suggest to explore the possibilities of involving methods that utilize ontologies, relations among interactions and build on top of a semantic reasoning.

- We focused our method for the linking of content items to a knowledge base on a specific domain. We are limited by movies and associated characteristics in the design of the method. As a future work we propose to focus on possibilities for a generalization of the method and evaluations in other domains as well.

- The proposed approach for the link prediction is build on top of the factorization of a third-order tensor, where the temporal dimension is modelled as a value within the tensor. We propose to consider the modification of the method in terms of using tensors of order four, where the fourth dimension would stand for the temporal dimension. Such modification can reveal more existing patterns in data and improve the quality of predictions.

- We limited our automatic management of semantic representations only to the link prediction. As continuation of our research we suggest to focus on an evaluation of existing links too - how existing links can be removed or even updated.

- Since we focused only on global preferences in our selection method, where specific values are assigned for all links of the same kind within semantic representations, we would like to explore the pros and cons of preferences for individual links. Applying of other graph-based metrics in our personalized selection algorithm is also another possibility to study their influence on results.

- The proposed method for the preference learning is limited by the repetitive applying of the rule learning to keep preferences updated. In order to overcome issues with repetitive updates of models, we propose to focus on streaming based rule learning algorithms. Another limitation is that the conventional rule learning algorithms are build on mining frequent itemsets and input data has to be properly preprocessed. As another possibility we see the improvement in using algorithms for mining frequent subgraphs and naturally use semantic representations. However, those algorithms suffer from incomparable complexity.

- Last but not least, we see the directions for our future research in an evaluation of the overall methodology in terms of studying the influence of individual steps on results of subsequent phases within our methodology- e.g. how the link prediction or removal influences the selection of entities or the preference learning.

Bibliography

[1] Maryam, J.; Farzad, S.; Shahram, J. Extracting Users' Navigational Behavior from Web Log Data: a Survey. *Journal of Computer Sciences and Applications*, volume 1, no. 3, 2013: pp. 39–45, ISSN 2328-725X, doi:10.12691/jcsa-1-3-3. Available from: http://pubs.sciepub.com/jcsa/1/3/3

[2] Gauch, S.; Speretta, M.; Chandramouli, A.; et al. User Profiles for Personalized Information Access. In *The Adaptive Web*, *Lecture Notes in Computer Science*, volume 4321, edited by P. Brusilovsky; A. Kobsa; W. Nejdl, Springer Berlin / Heidelberg, 2007, pp. 54–89.

[3] Chang, C.-H.; Kayed, M.; Girgis, M. R.; et al. A Survey of Web Information Extraction Systems. *IEEE Trans. on Knowl. and Data Eng.*, volume 18, no. 10, Oct. 2006: pp. 1411–1428, ISSN 1041-4347, doi:10.1109/TKDE.2006.152. Available from: http://dx.doi.org/10.1109/TKDE.2006.152

[4] Dědek, J. Web Information Extraction Systems for Web Semantization. In *ITAT*, *CEUR Workshop Proceedings*, volume 584, edited by P. Vojtáš, CEUR-WS.org, 2009, pp. 1–6.

[5] Stein, D.; Dorothea, T.; Pantelis, I.; et al. Deliverable D4.7: Evaluation and final results. 2015.

[6] Nickel, M.; Tresp, V.; Kriegel, H.-P. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, edited by L. Getoor; T. Scheffer, ICML '11, New York, NY, USA: ACM, June 2011, ISBN 978-1-4503-0619-5, pp. 809–816.

[7] Ford, L. R.; Fulkerson, D. R. Maximal flow through a network. *Canadian Journal of Mathematics*, volume 8, 1956: pp. 399–404.

[8] Toffler, A. *The Third Wave*. Morrow, 1980, ISBN 9780688035976. Available from: https://books.google.cz/books?id=ViRmAAAAIAAJ

[9] Zhou, B.; Hui, S.; Fong, A. Web Usage Mining for Semantic Web Personalization. 2004. Available from: http://www.win.tue.nl/persweb/program.html

[10] Stumme, G.; Berendt, B.; Hotho, A. Usage Mining for and on the Semantic Web. In *Proc. NSF Workshop on Next Generation Data Mining*, Baltimore, November 2002, pp. 77–86. Available from: http://www.kde.cs.uni-kassel.de/stumme/papers/2002/NSF-NGDM02.pdf

[11] Anand, S. S.; Kearney, P.; Shapcott, M. Generating semantically enriched user profiles for Web personalization. *ACM Trans. Internet Techn.*, volume 7, no. 4, 2007.

[12] Dojchinovski, M.; Kliegr, T. Entityclassifier.eu: Real-time Classification of Entities in Text with Wikipedia. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, ECMLPKDD'13, Springer-Verlag, 2013, pp. 1–1.

[13] Herlocker, J. L.; Konstan, J. A.; Borchers, A.; et al. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, New York, NY, USA: ACM, 1999, ISBN 1-58113-096-1, pp. 230–237, doi:10.1145/312624.312682. Available from: http://doi.acm.org/10.1145/312624.312682

[14] Dooms, S.; De Pessemier, T.; Martens, L. MovieTweetings: a Movie Rating Dataset Collected From Twitter. In *Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys 2013*, 2013.

[15] Paulheim, H.; Fümkranz, J. Unsupervised Generation of Data Mining Features from Linked Open Data. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, New York, NY, USA: ACM, 2012, ISBN 978-1-4503-0915-8.

[16] Di Noia, T.; Mirizzi, R.; Ostuni, V. C.; et al. Linked Open Data to Support Content-based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, New York, NY, USA: ACM, 2012, ISBN 978-1-4503-1112-0, pp. 1–8, doi:10.1145/2362499.2362501. Available from: http://doi.acm.org/10.1145/2362499.2362501

[17] Rowe, M. SemanticSVD++: Incorporating Semantic Taste Evolution for Predicting Ratings. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '14, Washington, DC, USA: IEEE Computer Society, 2014, ISBN 978-1-4799-4143-8-01, pp. 213–220, doi:10.1109/WI-IAT.2014.36. Available from: http://dx.doi.org/10.1109/WI-IAT.2014.36

[18] Sieg, A.; Mobasher, B.; Burke, R. D. Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. *IEEE Intelligent Informatics Bulletin*, volume 8, no. 1, 2007: pp. 7–18.

[19] Kearney, P.; An, S. S.; Shapcott, M. M.: Employing a domain ontology to gain insights into user behaviour. In *In: Proceedings of the 3rd Workshop on Intelligent Techniques for Web Personalization, at IJCAI 2005*, 2005.

[20] Lu, L.; Zhou, T. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, volume 390, no. 6, 2011: pp. 1150 – 1170, ISSN 0378-4371, doi:http://dx.doi.org/10.1016/j.physa.2010.11.027. Available from: `http://www.sciencedirect.com/science/article/pii/S037843711000991X`

[21] Kolda, T. G.; Bader, B. W. Tensor Decompositions and Applications. *SIAM Rev.*, volume 51, no. 3, Aug. 2009: pp. 455–500, ISSN 0036-1445, doi:10.1137/07070111X. Available from: `http://dx.doi.org/10.1137/07070111X`

[22] Spiegel, S.; Clausen, J.; Albayrak, S.; et al. Link prediction on evolving data using tensor factorization. In *Proceedings of the 15th international conference on New Frontiers in Applied Data Mining*, PAKDD'11, Berlin, Heidelberg: Springer-Verlag, 2012, ISBN 978-3-642-28319-2, pp. 100–110, doi:10.1007/978-3-642-28320-8_9. Available from: `http://dx.doi.org/10.1007/978-3-642-28320-8_9`

[23] Acar, E.; Dunlavy, D. M.; Kolda, T. G. Link Prediction on Evolving Data Using Matrix and Tensor Factorizations. In *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, ICDMW '09, Washington, DC, USA: IEEE Computer Society, 2009, ISBN 978-0-7695-3902-7, pp. 262–269, doi:10.1109/ICDMW.2009.54. Available from: `http://dx.doi.org/10.1109/ICDMW.2009.54`

[24] Dunlavy, D. M.; Kolda, T. G.; Acar, E. Temporal Link Prediction Using Matrix and Tensor Factorizations. *ACM Trans. Knowl. Discov. Data*, volume 5, no. 2, Feb. 2011: pp. 10:1–10:27, ISSN 1556-4681, doi:10.1145/1921632.1921636. Available from: `http://doi.acm.org/10.1145/1921632.1921636`

[25] Ermis, B.; Acar, E.; Cemgil, A. T. Link Prediction via Generalized Coupled Tensor Factorisation. *CoRR*, volume abs/1208.6231, 2012.

[26] Anderson, J. R. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, volume 22, 1983: pp. 261–295.

[27] Choudhury, S.; Breslin, J.; Passant, A. Enrichment and ranking of the youtube tag space and integration with the linked data cloud. *The Semantic Web-ISWC 2009*, 2009: pp. 747–762.

[28] Freitas, A.; Oliveira, J.; O'Riain, S.; et al. Querying linked data using semantic relatedness: a vocabulary independent approach. *Natural Language Processing and Information Systems*, 2011: pp. 40–51.

[29] Dix, A.; Katifori, A.; Lepouras, G.; et al. Spreading activation over ontology-based resources: from personal context to web scale reasoning. *International Journal of Semantic Computing*, volume 4, no. 1, 2010: p. 59.

[30] Crestani, F. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, volume 11, 1997: pp. 453–482.

[31] Dembczynski, K.; Kotlowski; W., R., Slowinski; et al. Learning of Rule Ensembles for Multiple Attribute Ranking Problems. In *Preference Learning*, edited by J. Fürnkranz; E. Hüllermeier, Springer-Verlag, 2010, pp. 217–247. Available from: http://www.springer.com/computer/ai/book/978-3-642-14124-9

[32] Wang, R.-Q.; Kong, F.-S. Semantic-Enhanced Personalized Recommender System. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 7, Aug 2007, pp. 4069–4074, doi:10.1109/ICMLC.2007.4370858.

[33] Lops, P.; Gemmis, M.; Semeraro, G. *Recommender Systems Handbook*, chapter Content-based Recommender Systems: State of the Art and Trends. Boston, MA: Springer US, 2011, ISBN 978-0-387-85820-3, pp. 73–105, doi:10.1007/978-0-387-85820-3_3. Available from: http://dx.doi.org/10.1007/978-0-387-85820-3_3

[34] Cantador, I.; Castells, P. Extracting Multilayered Communities of Interest from Semantic User Profiles: Application to Group Modeling and Hybrid Recommendations. *Computers in Human Behavior 27, special issue on Social and Humanistic Computing for the Knowledge Society*, July 2011: pp. 1321–1336. Available from: http://ir.ii.uam.es/pubs/chb11.pdf

[35] Cantador, I.; Bellogín, A.; Castells, P. *Adaptive Hypermedia and Adaptive Web-Based Systems: 5th International Conference, AH 2008, Hannover, Germany, July 29 - August 1, 2008. Proceedings*, chapter News@hand: A Semantic Web Approach to Recommending News. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ISBN 978-3-540-70987-9, pp. 279–283, doi:10.1007/978-3-540-70987-9_34. Available from: http://dx.doi.org/10.1007/978-3-540-70987-9_34

[36] Nguyen, A.-T.; Denos, N.; Berrut, C. Improving New User Recommendations with Rule-based Induction on Cold User Data. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-730–8, pp. 121–128, doi:10.1145/1297231.1297251. Available from: http://doi.acm.org/10.1145/1297231.1297251

[37] Rainsberger, J.; Tin, R.; Tong, T. Rule-based personalization framework for integrating recommendation systems. Mar. 8 2005, uS Patent 6,865,565. Available from: https://www.google.com/patents/US6865565

[38] Sandvig, J. J.; Mobasher, B.; Burke, R. Robustness of Collaborative Recommendation Based on Association Rule Mining. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-730–8, pp. 105–112, doi:10.1145/1297231.1297249. Available from: http://doi.acm.org/10.1145/1297231.1297249

[39] Abel, F.; Bittencourt, I. I.; Henze, N.; et al. *Adaptive Hypermedia and Adaptive Web-Based Systems: 5th International Conference, AH 2008, Hannover, Germany, July 29 - August 1, 2008. Proceedings*, chapter A Rule-Based Recommender System for Online Discussion Forums. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ISBN 978-3-540-70987-9, pp. 12–21, doi:10.1007/978-3-540-70987-9_4. Available from: http://dx.doi.org/10.1007/978-3-540-70987-9_4

[40] Kadziński, M.; Słowiński, R.; Greco, S. Robustness analysis for decision under uncertainty with rule-based preference model. *Information Sciences*, volume 328, 2016: pp. 321 – 339, ISSN 0020-0255, doi:http://dx.doi.org/10.1016/j.ins.2015.07.062. Available from: http://www.sciencedirect.com/science/article/pii/S0020025515006325

[41] Vitvar, T. Semantic Web. September 2010, fTS, Czech Technical University in Prague, Habilitation thesis.

[42] Pan, J. Z. Resource Description Framework. In *Handbook on Ontologies*, Springer Publishing Company, Incorporated, 2009, pp. 71–90, doi:10.1007/978-3-540-92673-3_3. Available from: http://dx.doi.org/10.1007/978-3-540-92673-3_3

[43] Antoniou, G.; van Harmelen, F. *A Semantic Web Primer, 2nd Edition*. The MIT Press, second edition, March 2008, ISBN 0262012421, 288 pp.

[44] Heath, T.; Bizer, C. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, first edition, 2011, ISBN 9781608454303. Available from: http://linkeddatabook.com/

[45] Domingue, J.; Fensel, D.; Hendler, J. A. (editors). *Handbook of Semantic Web Technologies*. Berlin: Springer, 2011, ISBN 978-3-540-92912-3, doi:10.1007/978-3-540-92913-0.

[46] Lehmann, J.; Isele, R.; Jakob, M.; et al. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal*, volume 6, no. 2, 2015: pp. 167–195, doi:10.3233/SW-140134. Available from: http://dx.doi.org/10.3233/SW-140134

[47] Kosala, R.; Blockeel, H. Web Mining Research: A Survey. *SIGKDD Explor. Newsl.*, volume 2, no. 1, June 2000: pp. 1–15, ISSN 1931-0145, doi:10.1145/360402.360406. Available from: http://doi.acm.org/10.1145/360402.360406

[48] Scime, A. *Web Mining: Applications and Techniques*. Idea Group Pub., 2005, ISBN 9781591404149. Available from: https://books.google.cz/books?id=TDhPMs3adw0C

[49] Bakariya, B.; Mohbey, K.; Thakur, G. An Inclusive Survey on Data Preprocessing Methods Used in Web Usage Mining. In *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, *Advances in Intelligent Systems and Computing*, volume 202, edited by J. C. Bansal; P. Singh; K. Deep; M. Pant; A. Nagar, Springer India, 2013, ISBN 978-81-322-1040-5, pp. 407–416, doi:10.1007/978-81-322-1041-2_35. Available from: http://dx.doi.org/10.1007/978-81-322-1041-2_35

[50] Srivastava, J.; Cooley, R.; Deshpande, M.; et al. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, volume 1, no. 2, 2000: pp. 12–23, doi:10.1145/846183.846188. Available from: http://doi.acm.org/10.1145/846183.846188

[51] V. V. R. Maheswara Rao, V. V. K.; Raju, K. V. S. V. N. Study of Visitor Behavior by Web Usage Mining. In *Information Processing and Management*, 2010, pp. 181–187.

[52] Chitraa, V.; Davamani, A. S. A Survey on Preprocessing Methods for Web Usage Data. *CoRR*, volume abs/1004.1257, 2010. Available from: http://arxiv.org/abs/1004.1257

[53] Velásquez, J. D.; Jain, L. C. *Advanced Techniques in Web Intelligence-1*. Springer, 2010, ISBN 978-3-642-14460-8, 284 pp.

[54] Mobasher, B. Web Usage Mining. In *Web Data Mining*, Data-Centric Systems and Applications, Springer Berlin Heidelberg, 2007, ISBN 978-3-540-37881-5, pp. 449–483, doi:10.1007/978-3-540-37882-2_12. Available from: http://dx.doi.org/10.1007/978-3-540-37882-2_12

[55] Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer, first edition, January 2009, ISBN 3540378812.

[56] Agrawal, R.; Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, ISBN 1-55860-153-8, pp. 487–499. Available from: http://dl.acm.org/citation.cfm?id=645920.672836

[57] Mobasher, B. Data Mining for Web Personalization. In *The Adaptive Web*, 2007, pp. 90–135.

[58] Pabarskaite, Z.; Raudys, A. A process of knowledge discovery from web log data: Systematization and critical review. *Journal of Intelligent Information Systems*, volume 28, no. 1, 2007: pp. 79–104, ISSN 0925-9902, doi:10.1007/s10844-006-0004-1. Available from: http://dx.doi.org/10.1007/s10844-006-0004-1

[59] Zeng, Y.; Wang, Y.; Huang, Z.; et al. *Active Media Technology: 6th International Conference, AMT 2010, Toronto, Canada, August 28-30, 2010. Proceedings*, chapter User Interests: Definition, Vocabulary, and Utilization in Unifying Search and Reasoning. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-15470-6, pp. 98–107, doi:10.1007/978-3-642-15470-6_11. Available from: http://dx.doi.org/10.1007/978-3-642-15470-6_11

[60] Hofgesang, P. I.; Patist, J. P. On Modelling and Synthetically Generating Web Usage Data. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '08, Washington, DC, USA: IEEE Computer Society, 2008, ISBN 978-0-7695-3496-1, pp. 98–102, doi:10.1109/WIIAT.2008.384. Available from: http://dx.doi.org/10.1109/WIIAT.2008.384

[61] Hofgesang, P. I. *Modelling Web Usage in a Changing Environment*. Dissertation thesis, Vrije Universiteit Amsterdam, 2009. Available from: http://hdl.handle.net/1871/13413

[62] Hofgesang, P. I. *Web Mining Applications in E-commerce and E-services*, chapter Online Mining of Web Usage Data: An Overview. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ISBN 978-3-540-88081-3, pp. 1–24, doi:10.1007/978-3-540-88081-3_1. Available from: http://dx.doi.org/10.1007/978-3-540-88081-3_1

[63] Dědek, J.; Eckhardt, A.; Vojtáš, P. Web Semantization - Design and Principles. In *Advances in Intelligent Web Mastering - 2*, *Advances in Intelligent and Soft Computing*, volume 67, edited by V. Snášel; P. Szczepaniak; A. Abraham; J. Kacprzyk, Springer Berlin Heidelberg, 2010, ISBN 978-3-642-10686-6, pp. 3–18, doi:10.1007/978-3-642-10687-3_1. Available from: http://dx.doi.org/10.1007/978-3-642-10687-3_1

[64] Maimon, O.; Rokach, L. *Data Mining and Knowledge Discovery Handbook*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005, ISBN 0387244352, 9780387244358.

[65] Kushmerick, N.; Weld, D. S.; Doorenbos, R. Wrapper Induction for Information Extraction. In *Proc. IJCAI-97*, 1997. Available from: http://citeseer.nj.nec.com/kushmerick97wrapper.html

[66] Labský, M.; Nekvasil, M.; Svátek, V. Towards web information extraction using extraction ontologies and (indirectly) domain ontologies. In *Proceedings of the 4th International Conference on Knowledge Capture (K-CAP 2007), October 28-31, 2007,*

Whistler, BC, Canada, 2007, pp. 201–202, doi:10.1145/1298406.1298454. Available from: http://doi.acm.org/10.1145/1298406.1298454

[67] Nixon, L.; Troncy, R. Survey of Semantic Media Annotation Tools for the Web: Towards New Media Applications with Linked Media. In *The Semantic Web: ESWC 2014 Satellite Events*, *Lecture Notes in Computer Science*, volume 8798, edited by V. Presutti; E. Blomqvist; R. Troncy; H. Sack; I. Papadakis; A. Tordai, Springer International Publishing, 2014, ISBN 978-3-319-11954-0, pp. 100–114, doi:10.1007/978-3-319-11955-7_9. Available from: http://dx.doi.org/10.1007/978-3-319-11955-7_9

[68] Pilz, A.; Paaß, G. Named Entity Resolution Using Automatically Extracted Semantic Information. In *LWA*, volume TUD-CS-2009-0157/TUD-KE-2009-04, edited by M. Hartmann; F. Janssen, FG Telekooperation/FG Knowledge Engineering, Technische Universität Darmstadt, Germany, 2009, pp. KDML:84–91.

[69] Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Linguisticae Investigationes*, volume 30, no. 1, January 2007: pp. 3–26, publisher: John Benjamins Publishing Company. Available from: http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002

[70] Kopecky, J.; Vitvar, T.; Bournez, C.; et al. SAWSDL: Semantic Annotations for WSDL and XML Schema. *Internet Computing, IEEE*, volume 11, no. 6, nov.-dec. 2007: pp. 60 –67, ISSN 1089-7801, doi:10.1109/MIC.2007.134.

[71] Sherif, M. A.; Ngomo, A.-C. N.; Lehmann, J. *The Semantic Web. Latest Advances and New Domains: 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 – June 4, 2015. Proceedings*, chapter Automating RDF Dataset Transformation and Enrichment. Cham: Springer International Publishing, 2015, ISBN 978-3-319-18818-8, pp. 371–387, doi:10.1007/978-3-319-18818-8_23.

[72] Gagnon, M.; Barrière, C.; Charton, E. Full Syntactic Parsing for Enrichment of RDF Dataset. In *Proceedings of the First International Conference on Linked Data for Information Extraction - Volume 1057*, LD4IE'13, Aachen, Germany, Germany: CEUR-WS.org, 2013, pp. 14–25. Available from: http://dl.acm.org/citation.cfm?id=2874472.2874475

[73] Kalfoglou, Y.; Schorlemmer, M. Ontology Mapping: The State of the Art. *Knowl. Eng. Rev.*, volume 18, no. 1, Jan. 2003: pp. 1–31, ISSN 0269-8889, doi:10.1017/S0269888903000651. Available from: http://dx.doi.org/10.1017/S0269888903000651

[74] Zhang, J.; Yu, P. S. *Link Prediction across Heterogeneous Social Networks: A Survey*. Dissertation thesis, University of Illinois at Chicago, 2014.

[75] Hasan, M.; Zaki, M. A Survey of Link Prediction in Social Networks. In *Social Network Data Analytics*, edited by C. C. Aggarwal, Springer US, 2011, ISBN 978-1-4419-8461-6, pp. 243–275, doi:10.1007/978-1-4419-8462-3_9. Available from: http://dx.doi.org/10.1007/978-1-4419-8462-3_9

[76] Davis, D.; Lichtenwalter, R.; Chawla, N. V. Multi-relational Link Prediction in Heterogeneous Information Networks. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '11, Washington, DC, USA: IEEE Computer Society, 2011, ISBN 978-0-7695-4375-8, pp. 281–288, doi:10.1109/ASONAM.2011.107. Available from: http://dx.doi.org/10.1109/ASONAM.2011.107

[77] Bizer, C.; Volz, J.; Kobilarov, G.; et al. Silk - A Link Discovery Framework for the Web of Data. In *18th International World Wide Web Conference*, April 2009. Available from: http://www2009.eprints.org/227/

[78] Fürnkranz, J.; Hüllermeier, E. Preference Learning: An Introduction. In *Preference Learning*, edited by J. Fürnkranz; E. Hüllermeier, Springer Berlin Heidelberg, 2011, ISBN 978-3-642-14124-9, pp. 1–17, doi:10.1007/978-3-642-14125-6_1. Available from: http://dx.doi.org/10.1007/978-3-642-14125-6_1

[79] Gemmis, M.; Iaquinta, L.; Lops, P.; et al. Learning Preference Models in Recommender Systems. In *Preference Learning*, edited by J. Fürnkranz; E. Hüllermeier, Springer Berlin Heidelberg, 2011, ISBN 978-3-642-14124-9, pp. 387–407, doi:10.1007/978-3-642-14125-6_18. Available from: http://dx.doi.org/10.1007/978-3-642-14125-6_18

[80] Ricci, F.; Rokach, L.; Shapira, B. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, edited by F. Ricci; L. Rokach; B. Shapira; P. B. Kantor, Springer US, 2011, ISBN 978-0-387-85819-7, pp. 1–35, doi:10.1007/978-0-387-85820-3_1. Available from: http://dx.doi.org/10.1007/978-0-387-85820-3_1

[81] Salton, G.; Wong, A.; Yang, C. S. A Vector Space Model for Automatic Indexing. *Commun. ACM*, volume 18, no. 11, Nov. 1975: pp. 613–620, ISSN 0001-0782, doi:10.1145/361219.361220. Available from: http://doi.acm.org/10.1145/361219.361220

[82] Terveen, L.; Hill, W. Human-computer collaboration in recommender systems. *HCI in the New Millennium*, 2001.

[83] Newell, C.; Miller, L. Design and Evaluation of a Client-side Recommender System. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, New York, NY, USA: ACM, 2013, ISBN 978-1-4503-2409-0, pp. 473–474, doi:10.1145/2507157.2508220. Available from: http://doi.acm.org/10.1145/2507157.2508220

[84] Wischenbart, M.; Firmenich, S.; Rossi, G.; et al. Recommender Systems for the People - Enhancing Personalization in Web Augmentation. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2015, co-located with ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 19, 2015.*, 2015, pp. 53–60. Available from: `http://ceur-ws.org/Vol-1438/paper10.pdf`

[85] d'Aquin, M.; Elahi, S.; Motta, E. Semantic monitoring of personal web activity to support the management of trust and privacy. In *SPOT 2010: 2nd Workshop on Trust and Privacy on the Social and Semantic Web*, 2010, co-located with ESWC2010, the 7th European Extended Semantic Web Conference held 30 May-3 June 2010 at Heraklion (Greece). Available from: `http://oro.open.ac.uk/24325/`

[86] Fujimoto, H.; Etoh, M.; Kinno, A.; et al. Web User Profiling on Proxy Logs and Its Evaluation in Personalization. In *Web Technologies and Applications*, *Lecture Notes in Computer Science*, volume 6612, edited by X. Du; W. Fan; J. Wang; Z. Peng; M. Sharaf, Springer Berlin / Heidelberg, 2011, pp. 107–118.

[87] Xu, G.; Zhang, Y.; Li, L.; et al. Web Mining and Recommendation Systems. In *Web Mining and Social Networking*, *Web Information Systems Engineering and Internet Technologies*, volume 6, edited by Y. Zhang, Springer US, 2011, ISBN 978-1-4419-7735-9, pp. 169–188.

[88] Karthikeyan, S.; Hakkeem, M. Extracting Web User Profiles Using H-UNC Clustering. *Asian Journal of Information Technology*, volume 10, 2011: pp. 78–83, ISSN 1682-3915.

[89] Grace, L. K. J.; Maheswari, V.; Nagamalai, D. Web Log Data Analysis and Mining. In *Advanced Computing*, *Communications in Computer and Information Science*, volume 133, edited by N. Meghanathan; B. K. Kaushik; D. Nagamalai, Springer Berlin Heidelberg, 2011, ISBN 978-3-642-17881-8, pp. 459–469.

[90] Tomczak, J.; Swiatek, J. Personalisation in Service-Oriented Systems Using Markov Chain Model and Bayesian Inference. In *Technological Innovation for Sustainability*, *IFIP Advances in Information and Communication Technology*, volume 349, edited by L. Camarinha-Matos, Springer Boston, 2011, pp. 91–98.

[91] Alice Marques, O. B. Discovering Student Web Usage Profiles Using Markov Chains. In *ECEL 2010 special issue*, volume 9, edited by C. V. de Carvalho, EJEL, 2011, pp. 91–98.

[92] Sankaradass, V.; Arputharaj, K. An Intelligent Recommendation System for Web User Personalization with Fuzzy Temporal Association Rules. *European Journal of Scientific Research*, volume 51, 2011: pp. 88–96, ISSN 1450-216X.

[93] Amatriain, X.; Jaimes, A.; Oliver, N.; et al. Data Mining Methods for Recommender Systems. In *Recommender Systems Handbook*, edited by F. Ricci; L. Rokach; B. Shapira; P. B. Kantor, Springer US, 2011, ISBN 978-0-387-85820-3, pp. 39–71.

[94] Rao, V. V. R. M.; Kumari, V. V.; Raju, K. V. S. V. N. *Information Processing and Management: International Conference on Recent Trends in Business Administration and Information Processing, BAIP 2010, Trivandrum, Kerala, India, March 26-27, 2010. Proceedings*, chapter Study of Visitor Behavior by Web Usage Mining. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-12214-9, pp. 181–187, doi:10.1007/978-3-642-12214-9_31. Available from: http://dx.doi.org/10.1007/978-3-642-12214-9_31

[95] Gad, W.; Kamel, M. Incremental clustering algorithm based on phrase-semantic similarity histogram. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, volume 4, July 2010, pp. 2088–2093, doi:10.1109/ICMLC.2010.5580499.

[96] Aghabozorgi, S. R.; Wah, T. Y. Recommender Systems: Incremental Clustering on Web Log Data. In *Proceedings of the 2Nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, New York, NY, USA: ACM, 2009, ISBN 978-1-60558-710-3, pp. 812–818, doi:10.1145/1655925.1656073. Available from: http://doi.acm.org/10.1145/1655925.1656073

[97] Leung, K. W.-T.; Lee, D. L. *Database Systems for Advanced Applications: 15th International Conference, DASFAA 2010, Tsukuba, Japan, April 1-4, 2010, Proceedings, Part I*, chapter Dynamic Agglomerative-Divisive Clustering of Clickthrough Data for Collaborative Web Search. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-12026-8, pp. 635–642, doi:10.1007/978-3-642-12026-8_48. Available from: http://dx.doi.org/10.1007/978-3-642-12026-8_48

[98] Plumbaum, T.; Stelter, T.; Korth, A. *User Modeling, Adaptation, and Personalization: 17th International Conference, UMAP 2009, formerly UM and AH, Trento, Italy, June 22-26, 2009. Proceedings*, chapter Semantic Web Usage Mining: Using Semantics to Understand User Intentions. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ISBN 978-3-642-02247-0, pp. 391–396, doi:10.1007/978-3-642-02247-0_42. Available from: http://dx.doi.org/10.1007/978-3-642-02247-0_42

[99] Kliegr, T. Towards Linked Hypernyms Dataset 2.0: complementing DBpedia with hypernym discovery and statistical type inferrence. In *LREC 2014*, ELRA, 2014.

[100] Mendes, P. N.; Jakob, M.; García-Silva, A.; et al. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, New York, NY, USA: ACM, 2011, ISBN 978-1-4503-0621-8, pp. 1–8, doi:10.1145/2063518.2063519. Available from: http://doi.acm.org/10.1145/2063518.2063519

[101] Rizzo, G.; Troncy, R. NERD: evaluating named entity recognition tools in the web of data. In *ISWC 2011, Workshop on Web Scale Knowledge Extraction (WEKEX'11), October 23-27, 2011, Bonn, Germany*, Bonn, ALLEMAGNE, 10 2011. Available from: http://www.eurecom.fr/publication/3517

[102] Uren, V.; Cimiano, P.; Iria, J.; et al. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, volume 4, no. 1, 2006: pp. 14 – 28, ISSN 1570-8268, doi:http://dx.doi.org/10.1016/j.websem.2005.10.002. Available from: http://www.sciencedirect.com/science/article/pii/S1570826805000338

[103] Friedman, N.; Getoor, L.; Koller, D.; et al. Learning probabilistic relational models. In *In IJCAI*, Springer-Verlag, 1999, pp. 1300–1309.

[104] Khosravi, H.; Bina, B. A survey on statistical relational learning. In *Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence*, AI'10, Berlin, Heidelberg: Springer-Verlag, 2010, ISBN 3-642-13058-5, 978-3-642-13058-8, pp. 256–268, doi:10.1007/978-3-642-13059-5_25. Available from: http://dx.doi.org/10.1007/978-3-642-13059-5_25

[105] Gao, S.; Denoyer, L.; Gallinari, P. Probabilistic Latent Tensor Factorization Model for Link Pattern Prediction in Multi-relational Networks. *CoRR*, volume abs/1204.2588, 2012.

[106] London, B.; Rekatsinas, T.; Huang, B.; et al. Multi-relational Learning Using Weighted Tensor Decomposition with Modular Loss. *CoRR*, volume abs/1303.1733, 2013.

[107] Taskar, B.; fai Wong, M.; Abbeel, P.; et al. Link Prediction in Relational Data. In *in Neural Information Processing Systems*, 2003.

[108] Raymond, R.; Kashima, H. Fast and Scalable Algorithms for Semi-supervised Link Prediction on Static and Dynamic Graphs. In *Machine Learning and Knowledge Discovery in Databases*, *Lecture Notes in Computer Science*, volume 6323, edited by J. Balcázar; F. Bonchi; A. Gionis; M. Sebag, Springer Berlin Heidelberg, 2010, ISBN 978-3-642-15938-1, pp. 131–147, doi:10.1007/978-3-642-15939-8_9. Available from: http://dx.doi.org/10.1007/978-3-642-15939-8_9

[109] Nickel, M.; Tresp, V.; Kriegel, H.-P. Factorizing YAGO: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, New York, NY, USA: ACM, 2012, ISBN 978-1-4503-1229-5, pp. 271–280, doi:10.1145/2187836.2187874. Available from: http://doi.acm.org/10.1145/2187836.2187874

[110] Ngomo, A.-C. N.; Auer, S. 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, AAAI Press, 2011, ISBN 978-1-57735-515-1, pp. 2312–2317, doi:10.5591/978-1-57735-516-8/IJCAI11-385. Available from: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-385

[111] Oyama, S.; Hayashi, K.; Kashima, H. Cross-Temporal Link Prediction. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, Washington, DC, USA: IEEE Computer Society, 2011, ISBN 978-0-7695-4408-3, pp. 1188–1193, doi:10.1109/ICDM.2011.45. Available from: http://dx.doi.org/10.1109/ICDM.2011.45

[112] Li, D.; Xu, Z.; Li, S.; et al. Link Prediction in Social Networks Based on Hypergraph. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, ISBN 978-1-4503-2038-2, pp. 41–42. Available from: http://dl.acm.org/citation.cfm?id=2487788.2487802

[113] Symeonidis, P.; Perentis, C. Link Prediction in Multi-modal Social Networks. In *Machine Learning and Knowledge Discovery in Databases*, *Lecture Notes in Computer Science*, volume 8726, edited by T. Calders; F. Esposito; E. Hüllermeier; R. Meo, Springer Berlin Heidelberg, 2014, ISBN 978-3-662-44844-1, pp. 147–162, doi:10.1007/978-3-662-44845-8_10. Available from: http://dx.doi.org/10.1007/978-3-662-44845-8_10

[114] Al-Sharawneh, J.; Williams, M.-A. A social network approach in Semantic Web Services Selection using Follow the Leader behavior. In *Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. 13th*, sept. 2009, pp. 310 –319, doi:10.1109/EDOCW.2009.5331986.

[115] Torres, R.; Tapia, B.; Astudillo, H. Improving Web API Discovery by Leveraging Social Information. In *Web Services (ICWS), 2011 IEEE International Conference on*, july 2011, pp. 744 –745, doi:10.1109/ICWS.2011.96.

[116] Wang, S.; Zhu, X.; Zhang, H. Web Service Selection in Trustworthy Collaboration Network. In *e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on*, oct. 2011, pp. 153 –160, doi:10.1109/ICEBE.2011.71.

[117] Shafiq, M.; Alhajj, R.; Rokne, J. On the Social Aspects of Personalized Ranking for Web Services. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, sept. 2011, pp. 86 –93, doi:10.1109/HPCC.2011.21.

[118] Godse, M.; Bellur, U.; Sonar, R. Automating QoS Based Service Selection. In *Web Services (ICWS), 2010 IEEE International Conference on*, july 2010, pp. 534 –541, doi:10.1109/ICWS.2010.58.

[119] Yau, S.; Yin, Y. QoS-Based Service Ranking and Selection for Service-Based Systems. In *Services Computing (SCC), 2011 IEEE International Conference on*, july 2011, pp. 56 –63, doi:10.1109/SCC.2011.114.

[120] Lécué, F. Combining Collaborative Filtering and Semantic Content-Based Approaches to Recommend Web Services. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, sept. 2010, pp. 200 –205, doi:10.1109/ICSC.2010.37.

[121] Zheng, Z.; Ma, H.; Lyu, M.; et al. WSRec: A Collaborative Filtering Based Web Service Recommender System. In *Web Services, 2009. ICWS 2009. IEEE International Conference on*, july 2009, pp. 437 –444, doi:10.1109/ICWS.2009.30.

[122] Jiang, Y.; Liu, J.; Tang, M.; et al. An Effective Web Service Recommendation Method Based on Personalized Collaborative Filtering. In *Web Services (ICWS), 2011 IEEE International Conference on*, july 2011, pp. 211 –218, doi:10.1109/ICWS.2011.38.

[123] Zhang, Q.; Ding, C.; Chi, C.-H. Collaborative Filtering Based Service Ranking Using Invocation Histories. In *Web Services (ICWS), 2011 IEEE International Conference on*, july 2011, pp. 195 –202, doi:10.1109/ICWS.2011.61.

[124] Xu, S.; Jiang, H.; Lau, F. C. Personalized Online Document, Image and Video Recommendation via Commodity Eye-tracking. In *RecSys '08*, New York, NY, USA: ACM, 2008, ISBN 978-1-60558-093-7, pp. 83–90, doi:10.1145/1454008.1454023. Available from: http://doi.acm.org/10.1145/1454008.1454023

[125] Pazzani, M. J.; Billsus, D. Content-Based Recommendation Systems. In *The Adaptive Web*, LNCS, Springer, 2007, ISBN 978-3-540-72078-2, pp. 325–341, doi:10.1007/978-3-540-72079-9_10. Available from: http://dx.doi.org/10.1007/978-3-540-72079-9_10

[126] Pradeep, A.; Knight, R.; Gurumoorthy, R. Methods and apparatus for providing personalized media in video. June 11 2013, uS Patent 8,464,288. Available from: https://www.google.com/patents/US8464288

[127] Orwell, G. *1984*. 1st World Library - Literary Society, 2004, ISBN 9781595404329. Available from: http://books.google.cz/books?id=w-rb62wiFAwC

[128] Heitmann, B.; Hayes, C. Using linked data to build open, collaborative recommender systems. *Artificial Intelligence*, 2010: pp. 76–81. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.174.2755http://www.aaai.org/ocs/index.php/SSS/SSS10/paper/viewPDFInterstitial/1067/1452

[129] Mabroukeh, N. R.; Ezeife, C. I. Using Domain Ontology for Semantic Web Usage Mining and Next Page Prediction. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, New York, NY, USA: ACM, 2009, ISBN 978-1-60558-512-3, pp. 1677–1680, doi:10.1145/1645953.1646202. Available from: http://doi.acm.org/10.1145/1645953.1646202

[130] Wei, L.; Lei, S. *Active Media Technology: 5th International Conference, AMT 2009, Beijing, China, October 22-24, 2009. Proceedings*, chapter Integrated Recommender Systems Based on Ontology and Usage Mining. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ISBN 978-3-642-04875-3, pp. 114–125, doi:10.1007/978-3-642-04875-3_16. Available from: http://dx.doi.org/10.1007/978-3-642-04875-3_16

[131] Hofgesang, P. I. Relevance of Time Spent on Web Pages. In *In Proc. of WebKDD 2006: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006*, 2006.

[132] Nagy, I.; Gaspar-Papanek, C. User Behaviour Analysis Based on Time Spent on Web Pages. In *Web Mining Applications in E-commerce and E-services*, *Studies in Computational Intelligence*, volume 172, edited by I.-H. Ting; H.-J. Wu, Springer Berlin Heidelberg, 2009, ISBN 978-3-540-88080-6, pp. 117–136, doi:10.1007/978-3-540-88081-3_7. Available from: http://dx.doi.org/10.1007/978-3-540-88081-3_7

[133] Billard, L.; Diday, E. Symbolic Regression Analysis. In *Classification, Clustering, and Data Analysis*, edited by K. Jajuga; A. Sokołowski; H.-H. Bock, Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, 2002, ISBN 978-3-540-43691-1, pp. 281–288, doi:10.1007/978-3-642-56181-8_31. Available from: http://dx.doi.org/10.1007/978-3-642-56181-8_31

[134] Loveard, T.; Ciesielski, V. Representing classification problems in genetic programming. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 2, 2001, pp. 1070–1077 vol. 2, doi:10.1109/CEC.2001.934310.

[135] Badran, K.; Rockett, P. Integrating Categorical Variables with Multiobjective Genetic Programming for Classifier Construction. In *Genetic Programming*, *Lecture Notes in Computer Science*, volume 4971, edited by M. O'Neill; L. Vanneschi; S. Gustafson; A. Esparcia Alcázar; I. De Falco; A. Della Cioppa; E. Tarantino, Springer Berlin Heidelberg, 2008, ISBN 978-3-540-78670-2, pp. 301–311, doi:10.1007/978-3-540-78671-9_26. Available from: http://dx.doi.org/10.1007/978-3-540-78671-9_26

[136] Gupta, S.; Kim, J.; Grauman, K.; et al. Watch, Listen & Learn: Co-training on Captioned Images and Videos. In *Machine Learning and Knowledge Discovery in Databases*, *Lecture Notes in Computer Science*, volume 5211, edited by W. Daelemans; B. Goethals; K. Morik, Springer Berlin Heidelberg, 2008, ISBN

978-3-540-87478-2, pp. 457–472, doi:10.1007/978-3-540-87479-9_48. Available from: http://dx.doi.org/10.1007/978-3-540-87479-9_48

[137] Celma, O. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer Publishing Company, Incorporated, first edition, 2010, ISBN 3642132863, 9783642132865.

[138] Goldberg, K.; Roeder, T.; Gupta, D.; et al. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Inf. Retr.*, volume 4, no. 2, July 2001: pp. 133–151, ISSN 1386-4564.

[139] Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; et al. Improving Recommendation Lists Through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, New York, NY, USA: ACM, 2005, ISBN 1-59593-046-9, pp. 22–32, doi:10.1145/1060745.1060754. Available from: http://doi.acm.org/10.1145/1060745.1060754

[140] Ebbinghaus, H. *Memory: A contribution to experimental psychology*. Teachers college, Columbia university, 1913.

[141] Rula, A.; Palmonari, M.; Harth, A.; et al. On the Diversity and Availability of Temporal Information in Linked Open Data. In *The Semantic Web - ISWC 2012*, *Lecture Notes in Computer Science*, volume 7649, edited by P. Cudré-Mauroux; J. Heflin; E. Sirin; T. Tudorache; J. Euzenat; M. Hauswirth; J. Parreira; J. Hendler; G. Schreiber; A. Bernstein; E. Blomqvist, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-35175-4, pp. 492–507, doi:10.1007/978-3-642-35176-1_31. Available from: http://dx.doi.org/10.1007/978-3-642-35176-1_31

[142] Gutiérrez-Basulto, V.; Klarman, S. Towards a Unifying Approach to Representing and Querying Temporal Data in Description Logics. In *Web Reasoning and Rule Systems*, *Lecture Notes in Computer Science*, volume 7497, edited by M. Krötzsch; U. Straccia, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-33202-9, pp. 90–105, doi:10.1007/978-3-642-33203-6_8. Available from: http://dx.doi.org/10.1007/978-3-642-33203-6_8

[143] Vitvar, T.; Kopecký, J.; Viskova, J.; et al. WSMO-Lite Annotations for Web Services. In *ESWC*, 2008, pp. 674–689.

[144] Akim, N. M.; Dix, A.; Katifori, A.; et al. Spreading Activation for Web Scale Reasoning: Promise and Problems. In *Proceedings of the ACM WebSci'11*, 2011.

[145] Liu, B.; Hsu, W.; Ma, Y. Integrating Classification and Association Rule Mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, edited by P.-S. G. Agrawal R., Stolorz P., 1998, pp. 80–86.

[146] Vanhoof, K.; Depaire, B. Structure of association rule classifiers: a review. In *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Nov 2010, pp. 9–12, doi:10.1109/ISKE.2010.5680784.

[147] Antonie, M.-L.; Zaiane, O. Text document categorization by term association. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 19–26, doi:10.1109/ICDM.2002.1183881.

[148] Bekkerman, R.; El-Yaniv, R.; Tishby, N.; et al. On Feature Distributional Clustering for Text Categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, New York, NY, USA: ACM, 2001, ISBN 1-58113-331-6, pp. 146–153, doi:10.1145/383952.383976. Available from: `http://doi.acm.org/10.1145/383952.383976`

[149] Kille, B.; Hopfgartner, F.; Brodt, T.; et al. The plista Dataset. In *Proceedings of the International Workshop and Challenge on News Recommender Systems*, NRS'13, ACM, 10 2013, pp. 14–22.

[150] Li, W.; Han, J.; Pei, J. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In *The 2001 IEEE International Conference on Data Mining (ICDM'01)*, edited by N. Cercone; T. Y. Lin; X. Wu, IEEE Computer Society, 2001, ISBN 0-7695-1119-8, pp. 369–376.

[151] Yin, X.; Han, J. CPAR: Classification based on Predictive Association Rules. In *Proceedings of the SIAM International Conference on Data Mining*, San Franciso: SIAM Press, 2003, pp. 369–376.

[152] Vanhoof, K.; Depaire, B. Structure of association rule classifiers: a review. In *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Nov 2010, pp. 9–12, doi:10.1109/ISKE.2010.5680784.

[153] Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993, ISBN 1-55860-238-0.

[154] Han, J.; Pei, J.; Yin, Y. Mining Frequent Patterns Without Candidate Generation. *SIGMOD Rec.*, volume 29, no. 2, May 2000: pp. 1–12, ISSN 0163-5808, doi:10.1145/335191.335372. Available from: `http://doi.acm.org/10.1145/335191.335372`

[155] Coenen, F.; Leng, P.; Ahmed, S. Data structure for association rule mining: T-trees and P-trees. *IEEE Transactions on Knowledge and Data Engineering*, volume 16, no. 6, June 2004: pp. 774–778, ISSN 1041-4347, doi:10.1109/TKDE.2004.8.

[156] Quinlan, J. R. Learning Logical Definitions from Relations. *Machine Learning*, volume 5, no. 3, Sept. 1990: pp. 239–266, ISSN 0885-6125, doi:10.1023/A:1022699322624. Available from: `http://dx.doi.org/10.1023/A:1022699322624`

[157] Fürnkranz, J. FOSSIL: A Robust Relational Learner. In *Proceedings of the European Conference on Machine Learning on Machine Learning (ECML-94)*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994, ISBN 3-540-57868-4, pp. 122–137. Available from: http://dl.acm.org/citation.cfm?id=188408.188422

[158] Cohen, W. W. Fast Effective Rule Induction. In *Proceedings of the 12th International Conference on Machine Learning (ML'95)*, Lake Tahoe, CA: Morgan Kaufmann, 1995, pp. 115–123.

[159] Fürnkranz, J.; Gamberger, D.; Lavrač, N. *Foundations of Rule Learning*. Springer-Verlag, 2012, ISBN 978-3-540-75196-0, doi:10.1007/978-3-540-75197-7. Available from: http://www.springer.com/978-3-540-75196-0

[160] Quinlan, J. Induction of decision trees. *Machine Learning*, volume 1, no. 1, 1986: pp. 81–106, ISSN 0885-6125, doi:10.1007/BF00116251. Available from: http://dx.doi.org/10.1007/BF00116251

[161] Moghimipour, I.; Ebrahimpour, M. Comparing Decision Tree Method Over Three Data Mining Software. *International Journal of Statistics and Probability*, volume 3, no. 3, 2014, ISSN 1927-7040. Available from: http://www.ccsenet.org/journal/index.php/ijsp/article/view/37872

Reviewed Publications of the Author Relevant to the Thesis

Conference Papers

[A.1] J. Kuchař, M. Dojchinovski, T. Vitvar. *Exploiting Temporal Dimension in Tensor-based Link Prediction*. Web Information Systems and Technologies, V. Monfort et al. (Eds.): WEBIST 2015, Revised Selected Papers. Lecture Notes in Business Information Processing (LNBIP 246) published by Springer, 2016. ISBN: 978-3-319-30996-5.

[A.2] T. Kliegr, J. Kuchař. *Benchmark of Rule-based Classifiers in the News Recommendation Task*. Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France. Volume 9283, p. 130-141. Published by Springer, 2015. ISBN: 978-3-319-24026-8.

[A.3] J. Kuchař. *Augmenting a Feature Set of Movies Using Linked Open Data*. Proceedings of the RuleML 2015 Challenge, the Special Track on Rule-based Recommender Systems for the Web of Data, the Special Industry Track and the RuleML 2015 Doctoral Consortium hosted by the 9th International Web Rule Symposium (RuleML 2015), Berlin, Germany. Published by CEUR Workshop Proceedings, 2015. ISSN: 1613-0073.

The paper has been cited in:

- M. Kopecky, L. Peska, P. Vojtáš , M. Vomlelova. *Monotonization of User Preferences*, 11th International Conference FQAS 2015, Cracow, Poland. Published by Springer, 2015. ISBN: 978-3-319-26153-9.
- P. Vojtáš , M. Kopecky, M. Vomlelova. *Understanding Transparent and Complicated Users as Instances of Preference Learning for Recommender Systems*, 10th International Doctoral Workshop, MEMICS 2015, Telč, Czech Republic. Published by Springer, 2015. ISBN: 978-3-319-29816-0.

[A.4] J. Kuchař, M. Dojchinovski, T. Vitvar. *Time-Aware Link Prediction in RDF Graphs.* WEBIST 2015 - Proceedings of the 11th International Conference on Web Information Systems and Technologies, Lisbon, Portugal, p. 390-401. Published by SciTePress May 2015, ISBN: 978-989-758-106-9.

[A.5] J. Kuchař, T. Kliegr. *Bag-of-Entities text representation for client-side (video) recommender systems.* First Workshop on Recommender Systems for Television and online Video (RecSysTV), ACM RecSys 2014 Foster City, Silicon Valley, USA, October 2014.

[A.6] J. Kuchař, T. Kliegr. *Doporučování multimediálního obsahu s využitím senzoru Microsoft Kinect.* Znalosti 2014. 13th Annual Conference, Jasná pod Chopkom, Nízké Tatry, Slovakia, Published by Oeconomica, 2014, ISBN: 978-80-245-2054-4.

[A.7] J. Kuchař, T. Kliegr. *InBeat: News Recommender System as a Service @ CLEF-NEWSREEL'14.* CLEF-NEWSREEL, CLEF 2014 Conference on Multilingual and Multimodal Information Access Evaluation, Sheffield, UK. Published by CEUR Workshop Proceedings, 2014. ISSN: 1613-0073.

The paper has been cited in:

- D. Doychev, R. Rafter, A. Lawlor, B. Smyth. *News Recommenders: Real-Time, Real-Life Experiences*, 23rd International Conference, UMAP 2015, Dublin, Ireland. Published by Springer, 2015. ISBN: 978-3-319-20266-2

[A.8] T. Kliegr., J. Kuchař. *Orwellian Eye: Video Recommendation with Microsoft Kinect.* Conference on Prestigious Applications of Intelligent Systems - PAIS 2014, Co-located with the 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, Published by IOS Press, 2014, ISBN: 978-1-61499-418-3.

[A.9] J. Leroy, F. Rocca, M. Mancas, R. Madhkour, F. Grisard, T. Kliegr, J. Kuchař, J. Vit, I. Pirner, P. Zimmermann. *KINterestTV - Towards Non-invasive Measure of User Interest While Watching TV.* Innovative and Creative Developments in Multimodal Interaction Systems . Eds. Y. Rybarczyk, T. Cardoso, J. Rosas, and L. Camarinha-Matos. Published by Springer, 2014, ISBN: 978-3-642-55142-0.

[A.10] J. Kuchař, T. Kliegr. *GAIN: web service for user tracking and preference learning - a SMART TV use case.* Proceedings of the 7th ACM Recommender Systems Conference (RecSys 2013), Hong Kong, China. ACM, New York, NY, USA. Published by ACM, 2013, ISBN: 978-1-4503-2409-0.

The paper has been cited in:

- F. Rocca, P. Deken, F. Grisard, M. Mancas, B. Gosselin. *Real-Time Marker-Less Implicit Behavior Tracking for User Profiling in a TV Context* , 28th Annual Conference on Computer Animation and Social Agents (CASA 2015), 2015, Singapore, ISBN: 978-981-09-4946-4

[A.11] D. Tsatsou, M. Mancas, J. Kuchař, L. Nixon, M. Vacura, J. Leroy, F. Rocca, V. Mezaris. *When TV meets the Web: towards personalised digital media*. Semantic Multimedia Analysis and Processing, Evangelos Spyrou, Dimitrios Iakovidis, Phivos Mylonas (Eds.). Published by Crc Pr I Llc, 2014, ISBN: 978-1-4665-7549-3.

[A.12] J. Kuchař, T. Kliegr. *GAIN: Analysis of Implicit Feedback on Semantically Annotated Content*. 7th Workshop on Intelligent and Knowledge Oriented Technologies, Smolenice, Slovakia, Published by STU Bratislava, 2012, ISBN: 978-80-227-3812-5.

The paper has been cited in:

- J. Leroy, F. Rocca, M. Mancas, B. Gosselin. *Second screen interaction: an approach to infer tv watcher's interest using 3d head pose estimation*, WWW '13 Companion Proceedings of the 22nd international conference on World Wide Web companion, ACM, 2013, ISBN: 978-1-4503-2038-2.
- J. Leroy, F. Rocca, M. Mancas, B. Gosselin. *3D Head Pose Estimation for TV Setups*, 5th International ICST Conference, INTETAIN 2013, Mons, Belgium, Springer International Publishing, 2013, ISBN: 978-3-319-03891-9.

[A.13] M. Dojchinovski, J. Kuchař, T. Vitvar, M. Zaremba. *Personalized Graph-based Selection of Web APIs*. The 11th International Semantic Web Conference (ISWC 2012), Boston, USA. Published by Springer, p. 34-48, Volume 7649, 2012, ISBN: 978-3-642-35175-4.

The paper has been cited in:

- E. Wittern, V. Muthusamy, J. A. Laredo, M. Vukovic. *API Harmony: Graph-based search and selection of APIs in the cloud*, IBM Journal of Research and Development, Vol. 60, Issue 2-3, 2016. ISSN: 0018-8646.
- T. Liang, L. Chen, H. Ying, Z. Zheng, J. Wu. *Crowdsourcing based API Search via Leveraging Twitter Lists Information*, IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA. ISBN 978-1-4673-8493-3.
- K. Kim, J. Altmann. *Evolution of the software-as-a-Service innovation system through collective intelligence*, International Journal of Cooperative Information Systems, Volume 22, Issue 03, 2013. ISSN: 0218-8430.
- D. Bianchini, V. De Antonellis, M. Melchiori. *Exploiting Social Tagging in Web API Search*, On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Lecture Notes in Computer Science Volume 8185, Springer, Graz, Austria, 2013, p. 764-771, ISBN: 978-3-642-41029-1.
- E. Wittern, J. Laredo, M. Vukovic, V. Muthusamy, A. Slominski. *A Graph-based Data Model for API Ecosystem Insights*, IEEE International Conference on Web Services (ICWS), Anchorage, AK, p. 41 - 48, IEEE,2014, ISBN: 978-1-4799-5053-9.

- K. Kim, WR. Lee, J. Altmann. *SNA-based innovation trend analysis in software service networks*, Electronic Markets - The International Journal on Networked Business, Springer Berlin Heidelberg, 2014, ISSN: 1019-6781.
- D. Bianchini, V. De Antonellis, M. Melchiori. *Advanced Web API search patterns adding collective knowledge to public repository facets*, IIWAS '13 Proceedings of International Conference on Information Integration and Web-based Applications & Services, ACM New York, Vienna, Austria, 2013, ISBN: 978-1-4503-2113-6.
- J. Houghton , M. Siegel, M. Vukovic. *Towards a Model for Resource Allocation in API Value Networks*, Service-Oriented Computing - ICSOC 2014 Workshops, Paris, France, 2015, Springer, ISBN: 978-3-319-22884-6.
- P. Zhao, X. Ye. *An Artificial Neural Network for Predicting Service Rating in the Presence of Rating Manipulation*, SCC 2015- IEEE 12th International Conference on Services Computing, New York City, NY, USA, 2015, ISBN: 978-1-4673-7280-0.
- I. Alvertis, M. Petychakis, F. Lampathaki, D. Askounis, T. Kastrinogiannis. *A community-based, Graph API framework to integrate and orchestrate cloud-based services*, IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), 2014, Doha, Qatar.
- D. Bianchini, V. De Antonellis, M. Melchiori. *Composite Patterns for Web API Search in Agile Web Application Development*, 24th International Conference, DEXA 2013, Prague, Czech Republic, Springer, 2013, ISBN:978-3-642-40172-5.
- Xin Le,Fan Yu-shun. *Multi-patterns integration model for business services*, CIMS Journal - Computer Integrated Manufacturing Systems, 2015
- Xin Le, Fan Yu-shun. *Service composition analysis with collaboration*, China Academic Journal Electronic Publishing House, 2015, ISSN: 10000-0054.
- M. Vukovic. *The Role of Crowdsourcing and Semantic Web for Consumable APIs*, Crowdsourcing and the Semantic Web, Dagstuhl Reports, Vol. 4, Issue 7, 2014.

Journal Papers

[A.14] J. Kuchař, I. Jelínek. *Learning Semantic Web Usage Profiles by Using Genetic Algorithms.* International Journal on Information Technologies and Security (IJITS), No 4 (vol.3) Sofia, Bulgaria. 2011, p. 3-20, ISSN 1313-8251.

[A.15] J. Kuchař, I. Jelínek. *Scoring Pageview Based on Learning Weight Function.* International Journal on Information Technologies and Security (IJITS), No 4 (vol.2) Sofia, Bulgaria. 2010, p. 19-28, ISSN 1313-8251.

Other Publications

[A.16] J. Kuchař. *Web Usage Mining.* Ph.D. Minimum Thesis, Faculty of Electrical Engineering, Prague, Czech Republic, 2011.

[A.17] J. Kuchař, I. Jelínek. *Dynamical online modeling of web user behaviour in adaptive web.* Czech Technical University Workshop 2011, Prague, Czech Republic, 2011.

Remaining Publications of the Author

Conference Papers

[A.18] S. Vojíř, V. Zeman, J. Kuchař, T. Kliegr. *EasyMiner/R: Web Interface for Rule Learning and Classification in R.* Proceedings of the RuleML 2015 Challenge, the Special Track on Rule-based Recommender Systems for the Web of Data, the Special Industry Track and the RuleML 2015 Doctoral Consortium hosted by the 9th International Web Rule Symposium (RuleML 2015), Berlin, Germany. Published by CEUR Workshop Proceedings, 2015. ISSN: 1613-0073.

[A.19] T. Kliegr., J. Kuchař, D. Sottara, S. Vojíř. *Learning Business Rules with Association Rule Classifiers.* 8th International Symposium, RuleML 2014, Co-located with the 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, Published by Springer, 2014, ISBN: 978-3-319-09869-2.

The paper has been cited in:

- M. Morzy, A. Lawrynowicz , M. Zozulinski. *Using Substitutive Itemset Mining Framework for Finding Synonymous Properties in Linked Data*, 9th International Symposium, RuleML 2015, Berlin, Germany, 2015, ISBN 978-3-319-21541-9.

[A.20] D. Tsatsou, L. Nixon, M. Mancas, M. Vacura, R. Klein, J. Leroy, J. Kuchař, T. Kliegr, M. Kober, M. Loli, V. Mezaris. *Contextualised user profiling in networked media environments.* 2nd International Workshop on Augmented User Modeling in conjunction with 20th Conference on User Modeling, Adaptation and Personalization (UMAP 2012). Montreal, Canada. Published by CEUR Workshop Proceedings. 2012, ISSN: 1613-0073.

Other Publications

[A.21] J. Kuchař,T. Vitvar. *Integrace pomocí webových služeb.* Pražská technika. 2012, č. 6, s. 14-15. ISSN: 1213-5348