RDF-OWL based recommender for videos and books of future electronic libraries

Thomas Maketa, Alain April
ETS University
1100 Notre-Dame west
Montreal, Quebec, Canada
thomas.maketa-lutete.1@ens.etsmtl.ca
alain.april@etsmtl.ca

Emanuel Berndl, Harald Kosch
University of Passau
Innstrae 43, 94032
Passau, Germany
berndl.emanuel@uni-passau.de
harald.kosch@uni-passau.de

***Abstract*—This paper proposes an architecture of components as well as proposed techniques/technologies for a future electronic library semantic based recommender system, this system will be using RDF-OWL technology to be used for multimedia recommendation of videos and books. To analyze videos, a first item enrichment approach aims at using video dialogue translated into text and process them through semantic enrichment methods to extract concepts to allow consistent recommendation between videos and books. This paper presents how linked data concepts could be used to enrich textual and video media information using linked data content from open repositories.**

***Keywords*—recommender system; electronic library; linked data; video enrichment; open-source**

I. Introduction

Handling large amount of data and making sense of them have been studied since the late sixties, well before personal computers (PC), Internet, social networks, Internet of Things and Big Data. Already in 1965 the term "Information overload"(IO) was coined by [1] and [2] to describe the inability to take decision when handling amount of data exceeding one's cognitive processing capability [3].

The rise of PCs, Internet and their corollary applications made IO's concern increasingly acute. In 1996, with the growing popularity of Internet, Richard Wurman, a precursor of the current Big Data movement, had already identified that "*a tsunami of data will be crashing onto the beaches of the civilized world, a tidal wave of unrelated, growing data formed in bits and bytes, coming in an unorganized, uncontrolled manner*"[4]. We are now at this point in time, where this tsunami of data is becoming a reality and researchers are developing technologies to manage it and make sense of it.

One trend of research addresses the issue through developing applications recommending relevant items from large public multimedia digital libraries of items containing books and videos. Recommender systems (RS) help user of digital libraries to get pertinent item proposition based on their profile, taste, past browsing and past library borrowing history. While RSs have been around for a long time [5, 6], they are getting increased attention ever since the growing amount of available online multimedia data has made the need to filter out relevant items from a digital library search a necessity. Users of public electronic libraries search for a very wide spectrum of subjects, for example: cooking recipes, movie to watch, local events happening in their area and RSs can provide them just in time recommendations while doing the heavy data crunching in the background. With the increased complexity of RSs' use cases utilizing multimedia data, there is also increased intelligence and accuracy expected from them. Most of this additional intelligence and accuracy required from RS lies in better understanding of natural language and contextualization of information. Current semantic web and linked data research directly address these issues[7-10]. They have the potential to greatly improve the relevance and accuracy of RSs.

The objective of this paper is to present the architecture and concepts of a future fully semantic RS. This paper is composed of 3 sections, first, a research topic overview, second, the proposed architecture for a fully semantic RS for video and books and finally, a brief conclusion and future directions.

II. Research Topic Overview

A. Recommender Systems

RSs, are information retrieval systems that assist users in identifying objects of interest from a collection of items based on the user preferences or previous search history. Usually RSs generate a list of recommended objects ranked by their relevance [11, 12].

The recommendation process involves typically 5 elements: 1) a user who is searching for information; 2) a collection of items where the search is done (digital library); 3) the user preferences history, that can be expressed explicitly by the user, for instance by the user filling a form, or implicitly by observing the user previous searches and item selections; 4) a recommender algorithm that recommends items from the collection; and 5) an initial user search query that contains words or sentences. In modern electronic library systems users can either query directly or click on pre-recommended items already proposed by the recommender system.

In this paper we will repeatedly use the word "instance" to indistinctly designate an item of the collection or a user preference history (i.e. respectively element 2 and 3 of the recommendation process presented above). This is helpful as these two words are often used in combination throughout this text and because both can be considered as instances of a general class of element used to be recommended.

RSs algorithms typically use similarity measures between instances for establishing a ranked list of items to be recommended. To assess similarities between instances that might come from digital libraries of very different objects, for instance an ecommerce website (that recommends movie, electronics, furniture, etc…), RSs algorithms use an internal representation that maps instances into mathematical construct (i.e. vectors or graphs), that then can be compared, measured, and paired. RS internal representation must be general enough to consistently map collections of heterogeneous objects, while being precise enough to address the specificity of any user query. Many similarity measures can be calculated depending of the RS internal representation. The choice of the internal representation and the similarity measures greatly influence the performances of an RS [13] [14].

RSs internal representation vary for semantic and non-semantic RS. Most non semantic RS typically use a vector space model (VSM) approach while semantic RS typically use either VSM or RDF graph based approaches [15] [16] [7] [17] [8] [14] [18] [19] [20]. The VSM approach maps each instance to a vector in a features vector space. That vector can then be manipulated using vector algebra and it can be measured using vector similarity measurements. A feature vector space is chosen for the type of items to be recommended and the choice of feature will affect the quality of the recommendation as perceived by the electronic library user.

When using a VSM approach, the main parameters to determine are the vector terms weightings scheme and the feature vector similarity metric [11]. The most commonly used term weighting scheme is the Term Frequency-Inverse Document Frequency, (TF-IDF), that is based on the empirical observations that in a corpus (i.e. a collection of documents) terms occurring frequently in one document (TF =term-frequency), but rarely in the rest of the corpus (IDF = inverse-document-frequency), are more likely to be relevant to the topic of the document [11]. In this context a very popular similarity measure used is the cosine similarity.

The main challenge reported when using a VSM approach to RS is the "lack of intelligence" of VSM [11]. Due to their polysemy and synonymy characteristics, words found in books and extracted from video dialogue might have different meaning based on their context. VSM key words based search queries consider only the syntactic characteristic form the word and not its semantic meaning. For instance, a search for "white house" will search for the words "white" and "house" in isolation or in combination but will not consider element related the words "US presidency residence" that can be one important semantic meaning. Semantic approaches have been designed to address this issue [11]. Non semantic RS are essentially based on a VSM internal representation, while Semantic RS typically will either use VSM or RDF graphs combined or separately . Still, semantic VSM is "semanticized" which means that there is a semantic extraction processes, addressing word sense disambiguation, which is applied to instances to build a vector. Some examples of this approach are the Explicit Semantic Analysis (ESA) representation [10], and the three dimensional matrix representation of RDF [21]. Wikify, while not being a formal representation scheme, can also be cited here as it maps text into keyword using semantic information to address word sense disambiguation [22]. It should also be noted that for each of the provided example the semantics extraction strategies are very different which is a testimony of the potential of this approach.

Resulting RDF graphs representation suppose that instances are translated into RDF and stored in an RDF store (figure 1). To compute similarity measures between instances, graphs representation of instances are extracted using each instances' Universal Resource Identifier [23] as the initial nodes. The graph is extracted using a predefined depth. These graphs are then compared using RDF based similarity measures [6, 11, 16, 24].

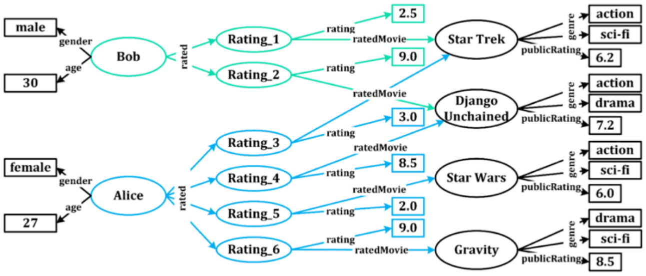

Figure 1: RDF graph example [25]

Similarities measures for RDF graphs might either use proprieties of the RDF graph structure or use its semantic information [14]. For instance a proximity measure between graphs using the number of common nodes between two graphs and not explicitly using semantic information carried by the graph will find instances that are more interconnected more similar [14]. Alternatively, semantic similarity measures can be derived by studying the two types of properties available in semantic graph: 1) non-taxonomical (i.e. object and datatype properties often defined using OWL format); and 2) taxonomical (i.e. those involving classes) [26]. Based on these two types of available RDF properties, one can design and experiment similarity measures combining both: comparing nodes from the same class between two graphs and exploit class hierarchy information or/and the properties these nodes have, can themselves be compared using measures adapted to the type of considered property, e.g. in using a measure to compare dates of book publication date. Then a global similarity score can be computed to aggregate the various measures and assess the similarities between two nodes [14, 26]. This is the approach used for the popular SemMF similarity measure [13] and used by [27, 28] to design their RDF Graph similarity measurement.

B. CB Recommender systems components

Lops et al. [11] splits the internal architecture of a content-based recommender algorithm using three logical components: 1) a content analyzer; 2) a user profile learner; and 3) an item filtering component on top of these elements. For semantic RS, an RDF storage system, for persisting its data, should also be added as a fourth component.

The Content analyzer is the component containing the information extraction algorithms that translate instances into the RS internal representation. The Profile learner is the component containing the supervised learning algorithm that build user profile. The profile learner use past user preferences to predict its appreciation of new items. Next, the filtering algorithm component suggests relevant items by matching the profile representation against that of items to be recommended. This component is in charge of producing the recommendation list of ranked items computed using some similarity measures [11].

Some limitations of RS are important to summarize here. RSs experience some limitations: cold start, sparsity, overspecialization and domain-dependency. Each one of these limitations can be mitigated but it is difficult to eliminate their effects entirely [28].

Cold start happens when a user preference history is absent and a recommendation is required. One of the consequences of missing historical data is that the RS will be taking more time to recommend relevant items.

Sparsity comes from the fact that user preference knowledge is sparse and incomplete and that little information is available because a user may have chosen not to disclose, or does not know, what he likes.

Recommendation diversity is also important for RS. Very similar items, like news on the same subject with the same information but from different source, should not be recommended twice. Recommendation list should include some novelty: items different from each other but still pertaining to the searched subject. RS should filter these items out and only recommend items that the user has not seen before.

Over specialization and domain dependency come from RS difficulties to propose items beyond those already rated by users. One possible solution to address this problem is the introduction of some randomness in the item filtering [11].

C. *Serendipity to find novel insight*

Another important and interesting characteristic of RS to include in this discussion is “serendipity”, that we call the “Eureka” factor. Serendipity “is the case of acquiring unsearched but still useful items or pieces of information” [29]. Serendipity allows RSs to recommend to user totally novel items that users did not even know existed. Due to over specialization, content-based RSs have no inherent method for generating serendipitous recommendations [11]. This is explained by the Gup’s theory[30]. Among approaches proposed to attain operationally induced serendipity [31] proposes: • Role of chance or blind luck, implemented via a random information node generator; • Pasteur principle (“ chance favors the prepared mind”), implemented via a user profile; • Anomalies and exceptions, partially implemented via poor similarity measures;• Reasoning by analogy, whose implementation is currently unknown.

This research topic overview of RSs has introduced characteristics and challenges that must be addressed by the proposed fully semantic RS for video and books. The next section will describe how this can be achieved.

III. Proposed fully semantic RS for video and books

This section introduces the design of the proposed fully semantic multimedia recommender system (SRS) for video and books that could be useful in future public electronic libraries. SRS should allow recommending videos and books indistinctly. Internally the SRS should not require adaptations when processing audio documents as well as it will translate them into the same RDF-OWL ontology that is used by the video processing component to allow consistent recommendations.

The strategy proposed for analyzing video is to first use the textual representation of the video (i.e. through its dialogue extraction) to enrich the content of the electronic library video items. It is obvious that, during this initial step, some visual video information that contributes to a user appreciation of a video, such as acting, landscape scenery or image quality for instance will not be captured. This will be done as a second research activity. This paper aims at describing the textual representation of the video as a first enrichment to be made available to recommendation algorithms.

One challenge of this approach is the translation of a book and textual video representation into RDF format. RDF is based on statements that are said to be “true” and that resemble statements like: “*this video is a science fiction movie*”, or “*this book’s theme is related to the Korean war*”. Extracting quality statements from a book and the textual representation of a video is the main challenge as it necessitates a good semantic understanding of the content. The proposed SRS architecture needed to achieve this is depicted in figure 2. The next sections describe each component of the architecture.

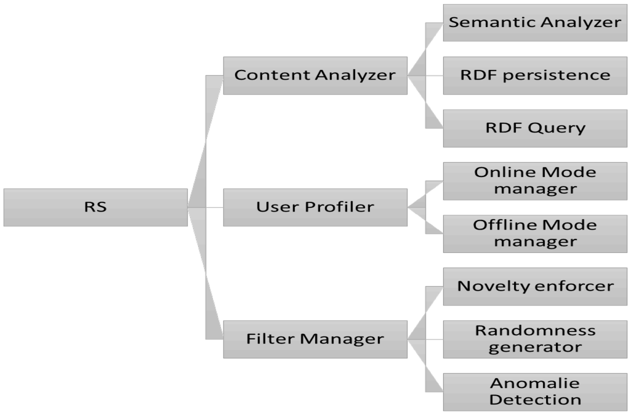

Figure 2: SRS Architecture

A. *Content Analyzer*

The content analyzer (CA) is the first module of the proposed SRS architecture (see figure 2). Its internal structure is further described by figures 3. The CA translates text from books and video's textual representation into RDF triples. The translation approach to semantic RDF representation of textual information was first experimented successfully in a previous research by (Oulaidi, 2016) from our research team. Oulaidi experimented an automated translation of text to RDF using the following 3 steps: 1) natural linguistic analysis - generation of normalized RDF triples using the WordNet thesaurus and the open source Natural Language Toolkit (NLTK); 2) translation of the triples to English, (i.e. because users queries can origin from any language), using the Google translate API and open source library TextBlob; and 3) topic modeling to identify the closest semantic RDF concepts using Latent Dirichlet Allocation (LDA) algorithms from the popular Mallet open source software (mallet.cs.umass.edu).

RDF generated by the CA should adhere to the web annotation data model (WADM)[32]. This is successfully done in the Media in Context (Mico) project [33]. Using WADM RDF allows easy integration with external applications, especially integration with Mico. It also allows the entire ontology structure to be easily extensible.

The ontology used to describe videos and books will be an extension of the media in context (mico) ontology as the current Mico ontology model does not have a vocabulary to model movie and books content.

The following ontology extension is proposed:

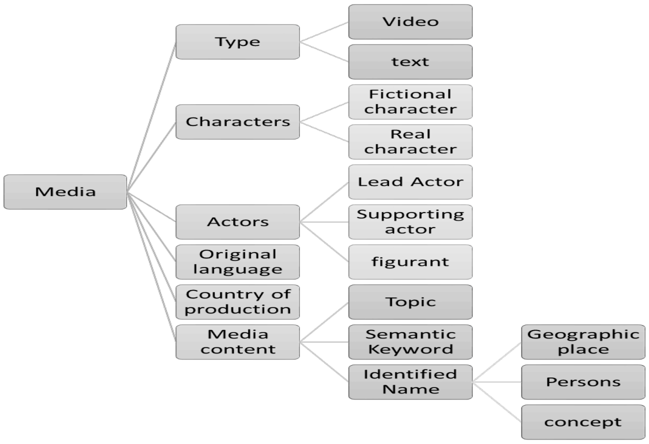

Figure 3 Ontology model

The figure 3 details a simplified ontology based on the ontology for media resource [34], each node can be considered as a class of objects and the entire graph depicts the hierarchy between them. The CA itself will use only the media content class and its sub classes containing information describing the content of the media.

The media content class possesses the following sub classes:

1. Topic (genre): the topic of the text as defined by a semantic topic modeler.
2. Semantic Keywords: The key words semantically extracted from the text.
3. Identified names: containing the names of geographic places, person or concepts included in the text.

It will be based on that ontology models that the RDF triples will be generated.

The content analyzer (CA) functional architecture is depicted in figure 4 bellow.

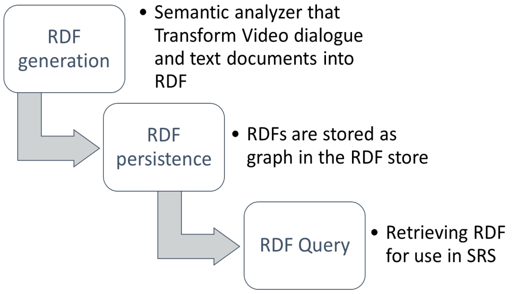
RDF generation
• Semantic analyzer that Transform Video dialogue and text documents into RDF
RDF persistence
• RDFs are stored as graph in the RDF store
RDF Query
• Retrieving RDF for use in SRS

Figure 4: Content Analyzer architecture

As we can see a semantic analyzer (SA) is a complex component. It is the most complex CA component of figure 3, it will be the component in charge of extracting the text topic, the semantic keywords and the identified names and turn them into RDF conform to the ontology proposed in the above section.

To better understand SA structure, it is further described in figure 4.

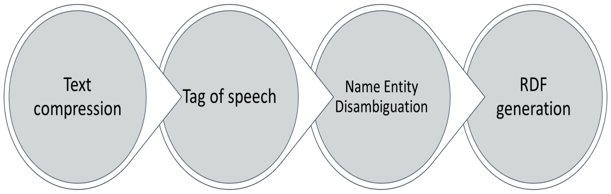

Figure 5: Semantic Analyzer Architecture

The proposed SA will be composed of four modules:

1. The text compression module (first module of figure 4) will extract key concepts from text and combine them into sentences (or keywords).

   This compression reduces the text to a smaller representation that still keeps the meaning of the entire text. This step reduces the processing time and contributes to high performance.

   It is at that step that the overall topic of the document will be detected.

   It should be noted that:

   a. Two approaches will be used for text compression: the first is based on Explicit Semantic Analysis (ESA) [10], that breaks inputs into semantic keywords that translate best the text's concepts. The second approach will use a neural network [35] that will break the text into sentences containing the key text concepts;

b. The ESA approach does not need the tag of speech step at this time and will go directly to the name entity disambiguation step.

c. These two approaches will be executed in parallel and results compared to confirm whether using neural network brings any performance improvement over using ESA.

2. The tag of speech module (second module of figure 4) will determine the components and role of each sentences' words. Google algorithm [35] used in the Google Syntax net library will be used by this module; This step is key to detecting the names and associate to them disambiguation based on the phrases they are used in;
3. The name entity disambiguation module (third module of figure 4) will ensure the disambiguation of names of persons, concepts, geographical areas. It will attempt to enrich the data, and link them to linked data open datasets such as DBpedia. This component will implement Chang et al. [36] approach that is based on the Wikify concept; and finally
4. The RDF generation module (the fourth module of figure 4) will create the resulting RDF. It will use Anno4j java library(https://github.com/anno4j/anno4j) to produce annotations conform to WADM. Anno4j will also facilitate RDF persistence into RDF store and provides the path-based query language LDPath, which is more convenient to users than building up complicated SPARQL queries.

Finally, for the CA RDF persistence technology we will assess how the open source Marmotta triple store (marmotta.apache.org) will perform. CA query and retrieval from the triple store will be experimented using SPARQL protocol and query language.

B. User Profiler

Now that the future content analyzer has been described, the user profiler (UP), which is the second component of the SRS architecture of Figure 2 is presented next. UP interacts with the user and monitor his interaction. It transforms user queries and action's history into RDF graphs. UP overall architecture is given in figures 2 and 5. UP has an online mode, to handle online session user queries and an offline mode to analyze user browsing action.

An example of the type of RDF graph structure resulting from the CA and UP persisting in the same RDF storage can be seen in figure 1. The SRS internal RDF graph structure will not exactly look like that, but it will have a similar structure.

The user profiler shares the same semantic model as the context analyzer. Based on [37] the user profiler will use semantic inference to improve profiling accuracy and integrates external linked dataset to address the cold start problem and make serendipity possible.

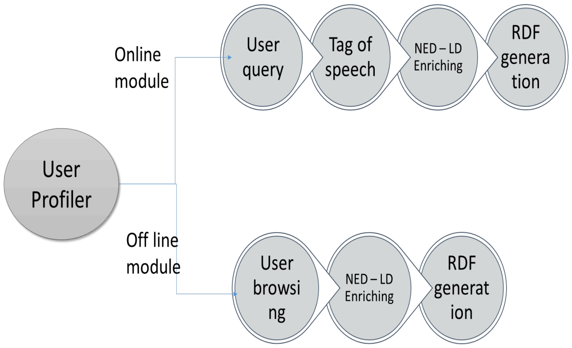

Figure 5: User profiler architecture

Then, the semF similarity measure will be used for comparison and ranking te similitude between books/video representation and user search/profile. Internally UP uses a deep learning algorithm to handle inference, however the system is built in such a way that changing the inference algorithm is possible without too much impact.

The user profiler will try to differentiate between short time and long term user interest to provide the most personalized experience. This approach is inspired by [28, 38].

C. Filter Manager

Now that the user profiler has been described, the filter manager (FM), which is the third and final component of the SRS architecture of Figure 2 is described. To avoid over specialization and promote diversity, the filtering algorithms will be built to promote novelty. Semantically similar items will be filtered out and some level of randomness will be incorporated to address overspecialization. An anomaly detection mechanism, similar to [29] ,will also be introduced as an additional measure to ensure serendipity.

IV. Conclusion

This paper has presented a literature review and challenges associated with semantic capabilities for future recommender systems of electronic libraries. Many semantic capabilities required have been discussed. The main contribution of this paper consists in a proposed semantic based recommender system architecture, and its components details. This proposed architecture is separated into three separate components: 1) a content analyzer; 2) a user profiler; and 2) a filter manager. Promising technologies and techniques have been presented. The next stage of the project is to design the SRS and conduct experimentations.

In the next stage, this SRS proposal will be evaluated in 2 ways: 1) using available data from the Netflix prize; and 2) using video data from an electronic library located in Quebec, Canada.

References

1. Gross, B.M., *The Managing of Organizations: The Administrative Struggle, Volume 1*. 1964, Free Press of Glencoe.
2. Toffler, A., *Future Shock*. 1970, Random House.
3. Cheri Speier1, J.S.V.a.V. *The Influence of Task Interruption on Individual Decision Making: An Information Overload Perspective*. in *Decision Sciences* 1999 Spring
4. Wurman, R.S., *Information Architect*. 1996: Graphis Inc.
5. Montaner, M., B. López, and J.L. De La Rosa, *A taxonomy of recommender agents on the internet.* Artificial intelligence review, 2003. **19**(4): p. 285-330.
6. Delgado-López., E.P.J.M.M.-d.-C.J.A., *Semantic Recommender Systems. Analysis of the state of the topic.* 2008.
7. Kumar, S.M., K. Anusha, and K.S. Sree, *Semantic Web-based Recommendation: Experimental Results and Test Cases.* 2015.
8. Ristoski, P., E.L. Mencía, and H. Paulheim, *A hybrid multi-strategy recommender system using linked open data*, in *Semantic Web Evaluation Challenge*. 2014, Springer. p. 150-156.
9. Vasileios, E. and G. Antoniou, *A real-time semantics-aware activity recognition system.* 2012.
10. Egozi, O., S. Markovitch, and E. Gabrilovich, *Concept-based information retrieval using explicit semantic analysis.* ACM Transactions on Information Systems (TOIS), 2011. **29**(2): p. 8.
11. Lops, P., M. De Gemmis, and G. Semeraro, *Content-based recommender systems: State of the art and trends*, in *Recommender systems handbook*. 2011, Springer. p. 73-105.
12. Anand Rajaraman, J.L., Jeffrey D. Ullman, *Mining massive dataset*. 2014: Stanford University.
13. Oldakowski, R. and C. Bizer. *SemMF: A framework for calculating semantic similarity of objects represented as RDF graphs*. in *Poster at the 4th International Semantic Web Conference (ISWC 2005)*. 2005.
14. Harispe, S., et al. *Semantic measures based on RDF projections: application to content-based recommendation systems*. in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*. 2013. Springer.
15. Piao, G. and J.G. Breslin. *Measuring Semantic Distance for Linked Open Data-enabled Recommender Systems*. in *The 31st ACM/SIGAPP Symposium on Applied Computing*. 2016.
16. Phuong T. Nguyen, P.T., Tommaso Di Noia, Eugenio Di Sciascio, *Content-based recommendations via DBpedia and Freebase: a case study in the music domain* 2015.
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*. 2014. IEEE Computer Society.
18. Di Noia, T., et al. *Exploiting the web of data in model-based recommender systems*. in *Proceedings of the sixth ACM conference on Recommender systems*. 2012. ACM.
19. Ricci, F., L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*. 2011: Springer.
20. Vivek K. Singh, M.G., Ramesh Jain, *Social pixels: genesis and evaluation.* ACM Multimedia, 2010: p. 481-490.
21. Di Noia, T., et al. *Linked open data to support content-based recommender systems*. in *Proceedings of the 8th International Conference on Semantic Systems*. 2012. ACM.
22. Mihalcea, R. and A. Csomai. *Wikify!: linking documents to encyclopedic knowledge*. in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007. ACM.
23. Group, W.C.W., *RDF 1.1 Primer*. 2014.
24. Nguyen, P., et al. *An evaluation of SimRank and Personalized PageRank to build a recommender system for the Web of Data*. in *Proceedings of the 24th International Conference on World Wide Web Companion*. 2015. International World Wide Web Conferences Steering Committee.
25. Victor Anthony Arrascue Ayala, M.P.-Z., Thomas Hornung, Alexander Schätzle, Georg Lausen, University of Freiburg, *Recommender Systems and SPARQL: More than a Shotgun Wedding?* 2014.
26. Euzenat, J. and P. Shvaiko, *Ontology matching*. Vol. 333. 2007: Springer.
27. Passant, A., B. Heitmann, and C. Hayes, *Using linked data to build recommender systems.* RecSys '9, New-York, NY USA, 2009.
28. Codina, V. and L. Ceccaroni, *A recommendation system for the semantic web*, in *Distributed Computing and Artificial Intelligence*. 2010, Springer. p. 45-52.
29. Iaquinta, L., et al. *Introducing serendipity in a content-based recommender system*. in *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*. 2008. IEEE.
30. Gup, T., *Technology and the End of Serendipity.* The Chronicle of Higher Education, 1997. **44**(52).
31. McCay - Peet, L. and E.G. Toms, *Investigating serendipity: How it unfolds and what may influence it.* Journal of the Association for Information Science and Technology, 2015. **66**(7): p. 1463-1476.
32. W3C, *Open Annotation Data Model.* 2013.
33. Kai Schlegel, E.B., Andreas Eisenkolb, *MICO Metadata Model, Media in Context Vocabulary.* 2015.
34. W3C, *Ontology for Media Resources 1.0.* 2012.
35. Andor, D., et al., *Globally normalized transition-based neural networks.* arXiv preprint arXiv:1603.06042, 2016.
36. Chang, A.X., et al., *Evaluating the word-expert approach for Named-Entity Disambiguation.* arXiv preprint arXiv:1603.04767, 2016.
37. Middleton, S.E., N.R. Shadbolt, and D.C. De Roure, *Ontological user profiling in recommender systems.* ACM Transactions on Information Systems (TOIS), 2004. **22**(1): p. 54-88.
38. Billsus, D. and M.J. Pazzani, *User modeling for adaptive news access.* User modeling and user-adapted interaction, 2000. **10**(2-3): p. 147-180.