RENDER
FP7-ICT-2009-5
Contract no.: 257790
www.render-project.eu

RENDER

Deliverable D3.3.1

Prototype of diversity-aware ranking

Editor:	Andreas Thalhammer, UIBK
Author(s):	Andreas Thalhammer, UIBK; Andreea Gagiu, UIBK; Simon Hangl, UIBK; Ioan Toma, UIBK; Maurice Grinberg, Ontotext
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	29.02.2012
Actual Delivery Date:	31.05.2012
Suggested Readers:	Researchers and practitioners in the Linked Data and NLP field
Version:	0.2
Keywords:	Ranking, Diversity, Result Diversification, Sentiment, Topic

Disclaimer

This document contains material, which is the copyright of certain RENDER consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU):
All RENDER consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP):
All RENDER consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE):
The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement. However, all RENDER consortium parties have agreed to make this document available to <group> / <purpose>.

In case of Consortium confidential (CO):
The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the RENDER consortium as a whole, nor a certain party of the RENDER consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

Full Project Title:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP3
Document Title:	D3.3.1 - Prototype of diversity-aware ranking
Editor (Name, Affiliation)	Andreas Thalhammer, UIBK
Work package Leader (Name, affiliation)	Ioan Toma, UIBK
Estimation of PM spent on the deliverable:	10

Executive Summary

This deliverable will present the details of the approach that was taken in order to produce the first prototype of diversity-aware ranking. As such, we focus on the output of the opinion and fact mining toolkits, i.e. structured data in KDO format. In the prototype we introduce the dimensions of topic coverage and sentiment.
In addition to that, we implemented the prototype as a RESTful service that enables to rank data on any RDF store which contains data in KDO compliant format.

The deliverable is structured as follows:

Section 2 will give detailed information about the state-of-the-art analysis that was conducted in order not to reinvent the wheel as well as to reconsider the most important definitions. At this point, it is worth to mention, that most state-of-the-art publications on search result diversification focus on a rather limited interpretation of diversity.

In Section 3 we will question this viewpoint and introduce the algorithm for the first prototype of diversity-aware ranking. This also includes a detailed description of the according APIs.

Finally, Section 4 summarizes all approaches that have been considered for the prototypical implementation as case studies. However, as some of these approaches look promising, we can be confident that we will incorporate these technologies in the next version of this deliverable.

List of authors

Organisation	Author
UIBK	Andreas Thalhammer
UIBK	Andreea Gagiu
UIBK	Simon Hangl
UIBK	Ioan Toma
Ontotext	Maurice Grinberg

Table of Contents

List of Figures

List of Tables

Abbreviations

HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
KDO	Knowledge Diversity Ontology
NER	Named Entity Recognition
REST	Representational State Transfer
RDF	Resource Description Framework
SIOC	Semantically-Interlinked Online Communities
URL	Uniform Resource Locator
XML	Extensible Markup Language

1 Introduction

This deliverable will present the details of the approach that was taken in order to produce the first prototype of diversity-aware ranking. As such, we focus on the output of the opinion and fact mining toolkits, i.e. structured data in KDO format. In the prototype we introduce the dimensions of topic coverage and sentiment.
In addition to that, we implemented the prototype as a RESTful service that enables to rank data on any RDF store which contains data in KDO compliant format.

The deliverable is structured as follows:

Section 2 will give detailed information about the state-of-the-art analysis that was conducted in order not to reinvent the wheel as well as to reconsider the most important definitions. At this point, it is worth to mention, that most state-of-the-art publications on search result diversification focus on a rather limited interpretation of diversity.

In Section 3 we will question this viewpoint and introduce the algorithm for the first prototype of diversity-aware ranking. This also includes a detailed description of the according APIs.

Finally, Section 4 summarizes all approaches that have been considered for the prototypical implementation as case studies. However, as some of these approaches look promising, we can be confident that we will incorporate these technologies in the next version of this deliverable.

2 Semantic and Diversified Document Ranking (State of the Art)

As with any kind of research, a sufficient analysis of the state of the art is inevitable. We took the effort and analyzed the state of the art in ranking structured semantic data (section 2.1) and in the field of search result diversification (section 2.2).

2.1 Ranking of Structured Semantic Data

2.1.1 Link-Based Analysis

Applying content analysis to data does not always yield accurate results due to the different ways in which information can be interpreted. A single word can have different meanings, depending on the context in which it is used. For example, the word “Java” can refer to multiple elements: a programming language, software platform, a brand of Russian cigarettes, a type of coffee or alcohol, a French band, a song, a fictional character (e.g. a comic book villain), a dance, a board game, and island or sea in Indonesia, a city in the USA (in Virginia, New York, Georgia, or South Dakota), or even to one of the oldest and rarest American breed of chicken[1] (and even many more not listed here). A safe assumption is usually that the number of meanings of any word is unbound. However, humans can distinguish among these meanings by taking into account the context (the relationships with the other elements in the text) while a machine that ignores such relationships cannot.

In order to overcome such ambiguity issues, content analysis techniques are combined with other approaches, such as link analysis, to extract the content around the resources and improve the accuracy of the results. Link analysis techniques focus on evaluating the relationships (connections) that define the structure of the information to be analyzed. These techniques provide additional information through the relationships between the different items contained by the information analyzed. Unlike the techniques presented in the previous section, link analysis relies on examining the graphs established among items, i.e. the nodes (items to rank) and edges (relationships connecting them). Therefore, using these techniques, implicit properties can be derived and included in the ranking process.

Two page ranking algorithms, are commonly used in web structure mining: (1) PageRank algorithm [31], and (2) HITS (Hyper-text Induced Topic Selection) [23]. Both algorithms consider all links equal in distributing the rank scores [47].

Based on a random walk algorithm, the PageRank system evaluated the probability of finding a random web surfer on any given page. A search engine may first retrieve a list of relevant pages to a given query based on keywords, and second applies the PageRank algorithm to adjust the results so that more “important”/ “relevant” pages are provided at the top of the page list. The PageRank algorithm states that if a page has important links to it, the links it provides to outside pages become also important; thus, the algorithm takes the backlinks into account and propagates the ranking through them: a page has a high rank if the sum of the ranks of its backlinks is high [31] [34].

On the other hand, HITS is a purely link-based algorithm and ranks webpages by analyzing their interlinks and outlinks: webpages pointed by many hyperlinks are called *authorities,* whereas webpages that point to many hyperlinks are called *hubs* [8], [23], [44]. Once the pages have been assembled, HITS ignores textual content and focuses on the structure of the Web only. Methods such as HITS [23] estimate the quality of Web pages and the topic relevance between the Web pages and the query.

Therefore, Link analysis implementations have been successfully applied for query independent ranking (also called static ranking). Three main extensions have been developed to increase the corpus of link analysis methodologies: (1) weighted link analysis, (2) hierarchical link analysis, and (3) semantic web link analysis.

Weighted link analysis refers to assigning more relevance to certain kind of links in accordance to their type. As most approaches were proposed for database research (and thus are not directly applicable on a

[1] http://en.wikipedia.org/wiki/Java_%28disambiguation%29, last checked 15.05.2012

web-scale), the main challenge of the present category of techniques consists in assigning weights to the links without affecting the overall performance.
Examples of implementations of this technique consist of WLRank, proposed by [3], and Weighted PageRank algorithm, proposed by [47]. Both approaches are offered as extensions to the PageRank algorithm, which consider different web page attributes to give more weight to some links, and, thus, improve the precisions of the answers. WLRank uses the relative position in the page tag where the link is contained and length of the anchor text. Similarly, Weighted PageRank assigns larger rank values to more important/popular pages instead of dividing the rank value of a page evenly among its outlink pages – each outlink page gets a value proportional to its popularity [47].

Intended for distributed environments, *hierarchical link analysis* performs a layered exploration of the underlying data. Xue et al. [48] argue that although link analysis algorithms have been used extensively in Web information retrieval, the algorithms generally work on a flat link graph and ignore the hierarchical structure of the Web graph. The authors propose a new link analysis algorithm, called Hierarchical Rank, which consists of two main components: (1) a new Web-link graph, which contains two layers, i.e. the upper-layer graph and the lower-layer graph, and (2) a random walk model which assumes that the user searches for information by starting from the upper-layer and either jumps to another upper-layer mode, or follows the hierarchical links down the lower-layer. The authors argue that the proposed algorithm can significantly improve the performance of the Web search, efficiently alleviate the rank problem and assign the reasonable rank to the newly-emerging Web Pages. Similarly, [21], [12], [45], [5] exploit the hierarchical structure of distributed environments and of the Web. However, the model presented has never been applied on semi-structured data sources with distinct semantics and none have taken into consideration weighted relations between supernodes [7].

Semantic Web link analysis methods aim to exploit the semantics of relationships during the ranking process. The techniques represent an evolution of the weighted link analysis applied to the Semantic Web context. In this respect, [2] propose SemRank, a method to rank semantic relationships using a blend of semantic and information-theoretic techniques with heuristics. The model supports the idea of modulative search, where users may vary their search modes to effect the changes in the ordering of results depending on their needs. On the other hand, the model proposed is solely focused on ranking and retrieval of relationships.
[33] propose a framework which uses related concepts inclusion and applies appropriate weighting functions. The ranking is done by scoring semantic document annotations based on document richness through their research prototype retrieval called PicoDoc.
Another example is the Swoogle search engine [10], [9] which proposes OntoRank, a variation of the PageRank algorithm for Semantic Web resources. OntoRank emulates user's navigation behavior at document level granularity using a rational surfer model. Ranking Semantic Web Documents (SWD) emulates a "rational" agent acquiring knowledge on the semantic web using the hyperlinks provided by [9]'s and [10]'s "semantic web navigation model" at document level. Intuitively, the algorithm estimates the probability that a rational surfer will visit a SWD, with the bias that ontologies are more preferred to instance data.

Additionally, link analysis methods can be classified in two main categories: untyped (any relationships between two items are equally considered regardless of the nature of semantic of such items) and typed (certain links are more relevant than others). Although link analysis techniques produce richer data models, they require a more complex processing.

2.1.2 Classification Based on the Information Used

According to the information used, ranking models can be classified in four categories: (1) Boolean, (2) statistical and probabilistic, (3) hyperlink based, and (4) conceptual models.

The *Boolean model* is considered the simplest form of retrieving documents according to how relevant they are to the user query. In this model, the query is formulated as a Boolean combination of terms using the classical operators AND, OR, and NOT. More complex queries can be built up out of these operators and are evaluated according to the traditional rules of Boolean algebra. Therefore, the query is a weightless phrase (or, it can also be seen as using only two weights – zero when the term is absent and one when it is present) and is either true of false [14]. The model indicates that the document either satisfies a query (is "relevant") or does not satisfy it (is "non-relevant") and, thus the document's tank will not be computed.

However, this type of ranking models has a large number of limitations. For instance, the results of such Boolean model algorithms either produce a large number of documents or none at all are retrieved. Moreover, classical Boolean models produce counter-intuitive results due to their all-or-nothing approach [14], such as the response to a multi-term OR, "a document containing all [or not many of] the query terms is not treated better than a document containing one term" [37]. In order to cope with these shortcomings, the Extended Boolean operators have been proposed. The Extended model assigns weights to both the document and the query's words and Extended Boolean operators are used to compute similarity measures.

One of the most important implementation of the Extended Boolean Model is $\rho - norm$ [25].

The *statistical model* is one of the oldest and most common models used for document ranking which utilizes a list of term for representing documents and queries. The methods from this category disregard any conceptual relations among terms (similar to the methods contained by the content-based analysis category). The focus of these techniques is on exploiting statistical information (e.g. term frequency, document length, etc.) for computing the similarity degree of the document and query.

An example of such a method is the Vector Space Model [24]. After removing stop words and stemming, the model computes the term's weights by the $tf \times idf$ formula [38]. The terms weighted build the document vector and the query vector, which will be normalized. The similarity degree of the document and the query is calculated using methods such as the calculation of the cosine of the angle between the two vectors, or distance functions. Methods contained by the vector space model techniques do not distinguish homonyms (similar words with different meaning) and synonyms (different words with similar meanings).

An alternative form of vector space model is LSI (Latent Semantic Index) [6] which uses statistical properties to extract term's conceptual relations to eliminate the drawbacks of vector space model.

Hyperlink-based models use, as the name suggests, hyperlink structured for ranking. This category includes algorithms such as PageRank, HITS (both discussed in the previous sections) or SALSA (an algorithm that uses the combination of ideas from both HITS and PageRank) [26].

In this category [50] introduce the concept of ranking the Semantic-linked Network, which contains different kinds of links between documents. The authors propose a personalized ranking algorithm where fuzzy-set rank is used to record the contribution of different semantic links to the rank of a semantic component. The rank of the document will be an average of the ranks of all links.

Conceptual Models try to extract the concepts of the documents and the query to compare them [38]. The ontology based model proposed by [42] is one of the conceptual models which map the phrases in a document into conceptual instances using annotations. The retrieval model is based on an adaptation of the classic vector-space model, including an annotation weighting algorithm and a ranking algorithm. Weights are assigned to the instances, indicating the relevancy degree of conceptual instances to document meaning. The user query is converted to an internal knowledge base which returns a list of relevant instances (e.g. conceptual tuples) that satisfy the query. Document ranking can be achieved through the vector space model.

TAP [18] presents a view of the Semantic Web where documents and concepts are nodes in a semantic network. This approach addressed two issues: (1) the development of a distributed query infrastructure for ontology data in the semantic Web, and (2) the presentation of query execution results (improving results by using the data from the surrounding models) [42].

Similarly, the model proposed by [35], based on Spread Activation (SA), represents documents and their concepts in a semantic network. Each link has a weight attached according to a SA based on certain properties of the ontology (e.g. similarity, specificity measure), measuring the strength of the relation. SA techniques are used to find related concepts in the ontology given an initial set of concepts and corresponding initial activation values. These initial values are obtained by applying classical search to the data. SA algorithm expands a set of initial components to their relevant components.
Other models for document ranking exist, such as language model [32], relaxation algorithm [46], which use natural language processing techniques, and take into consideration syntactic structure and morphological form of terms. In some cases, such as Content Based Model [36], information about the user is considered when calculating the rank of the documents.

2.2 Search Result Diversification

In the past, the problem of search result diversification has been tackled by a group of researchers. The first point that becomes obvious is that the term diversification can be interpreted in a variety of ways. [11] distinguishes between algorithms that focus on:

1. **Content diversity** focuses on items that are dissimilar in content (e.g. extracted keywords)
2. **Novelty diversity** focuses on items that contain additional information in comparison to already browsed ones
3. **Coverage diversity** focuses on items that cover as many topics or categories as possible.

However, as pointed out by [30], these perceptions of diversity are rather limited and do not cover sentiments or opinions. Unfortunately, the authors do not provide or point to valuable alternatives or solutions. As RENDER follows a holistic interpretation of diversity we focus on this novel track of perceiving diversity which is generally different from the above mentioned viewpoints. Nevertheless, the algorithms that were developed in the contexts of the above mentioned categories can serve as a baseline. In the following, we will analyse the past years' major contributions to search result diversification and we point out common weaknesses and strengths.

A rather early publication by Zhai et al. [49] describes so-called subtopic retrieval. The idea is to retrieve documents that include a lot of subtopics of the actual query topic. The authors also include an evaluation method that suits for this kind of information retrieval and states an alternative to the traditional relevance-based precision/recall measure. Moreover, a variety of methods addressing the problem of subtopic retrieval are introduced and evaluated with the introduced measures s-precision/s-recall. Of course, this paper can fits to the 3rd of the above mentioned categories. The introduced approaches for topic modelling are a novelty-based approach and a hybrid method that combines novelty and relevance.

Vee et al. perform diversity-aware ranking on structured data [43]. For this, the authors assume a table-based i.e. relational data model. Therefore, one of the key inputs for their algorithm is a ranking of the importance of the columns that has to be determined from domain to domain. Based on this, they employ a so-called Dewey Tree as a data model. Based on this, they employ a greedy algorithm replaces branches of the Dewey tree and maintains a “tentative result set”. Although the algorithm operates on structured data, we are not able to apply this algorithm for diversity ranking as – even though we have a domain model – we cannot provide a ranking between the properties as this varies between use cases.

Similarly, [19] also provide a ranking algorithm on structured data. The algorithm is based on the goal to find the so-called k-nearest diverse neighbour (KNDN). The author employs the greedy Motley algorithm that works with R-trees and minimum bounding rectangles for database navigation. In addition to that, the author introduces measures that help to estimate the degree of diversity. Unfortunately, the algorithm is tailored to relational databases and their spatial index structures.

[13] discusses a mathematical approach to search result diversification. The goal is to provide a solution to the contradictory-sounding goal of providing diverse and at the same time relevant results. The idea is to maximize a utility function that is tailored to a set of axioms. However, while introducing these axioms the

authors also prove that not all of them can be fulfilled at the same time. Having defined the goal and the constraints, a pragmatic approach is to try to maximize the value of the utility function for given a variety of possible result sets. The authors prove that the problem of MaxSumDispersion – that has been proven to be NP-hard - can be reduced to the optimization of the utility function. Although a 2-approximation algorithm exists for the MaxSumDispersion problem, we are not confident that the solution scales on a huge amount of Tweets (that is included in the RENDER use case).

Similar to approach by [13], the approach presented in [1] describes the problem of diversification of a result set as a mathematical function. The main focus of this paper lies on the diversification of results for ambiguous queries as an input. They also conclude that the problem is NP-hard and therefore a solution is only provided if it is tried to approximate the optimal solution. For this, the authors employ a greedy algorithm that iterates over the set of documents and – at each step – it selects the document with highest marginal utility. The authors evaluate the results with a set of standard measures over various data sets and also compare to with Mechanical Turk judgments.

As we can see on the publications above, the problem of result set diversification has been tackled from a variety of angles but one of the most obvious has not been treated yet, i.e. clustering. Although presenting a rather obvious solution to a complex problem, clustering is usually not considered for two main issues:

- Usually, it is computationally too intensive to be performed online.
- It is performed offline (pre-computed clusters) and therefore restricts the ability to adapt to the query.

However, van Leuken et al. present an efficient light-weight clustering-based approach for the presentation of diverse results for image search [27]. The authors consider two different scenarios: having a list of results that has already been ranked by relevance or starting with a set of equally important results. In this realm, the authors provide three algorithms of which one covers the first scenario and the two others relate to the second. The clustering finally enables to provide an interface with which the user does not get swamped by the flood of information. The clustering approach also proves to be flexible as the similarity measure can easily be adapted to new data models and information.

Above we have treated the most important publications in the field of search result diversification. Additionally, we also want to mention other publications that provide state of the art reviews about the topic of search result diversification. This gives us the opportunity to point the likeminded reader to [30] and [11].

Concluding the findings of above, we can separate the works into various categories: There is research which approaches the problem from a database point of view. In addition the problem of diversification has been formulated in mathematical terms and as such as a function for which estimating the optimum is in the category of the NP-hard problems. In our conclusion, we support the following points made by [30]:

- Most approaches employ greedy approximation algorithms.
- All algorithms are designed to work online.
- None of the state-of-the-art works try to perform offline pre-computation of (parts of) the results.
- Different notions of diversity such as combining opinion, sentiment and topics are not available.
- Clustering provides the option of flexible similarity measures.

In the following chapter we will consider these points while coming up with the description of the prototype of diversity-aware ranking.

3 Prototype: Diversity-aware Ranking of Diversity Information

The prototype of the diversity-aware ranking component focuses on data that has been pre-processed by an NLP engine which has topics and sentiments extracted. We assume that this information is represented in accordance to a diversity-enabling data model, i.e. the Knowledge Diversity Ontology (KDO, cf. D3.1.1 and D3.1.2).

This chapter is split into two main sections. In section 3.1 we will introduce the algorithm and methodology which is utilized by the diversity-aware ranking component. After that, in section 3.2, we describe the diversity-aware ranking interface.

3.1 Algorithm: Diversity-aware Ranking

The core of the algorithm for diversity-aware ranking is built on the findings of [27]. From this paper, we employ the “Fold” algorithm which takes a pre-ranked list of relevant – but not necessarily diverse – documents. In accordance to the prototypes of the opinion mining tool (D2.1.1) and the prototype of the fact mining tool (D2.2.1) the objective of the diversification focuses on covered topics in relation to the sentiment score (see the example in Figure 1).

Therefore, the approach is naturally also aligned to the “three steps to diversity” identified in [30], i.e. Relevance Measure, Diversity Measure, and Diversification Objective.

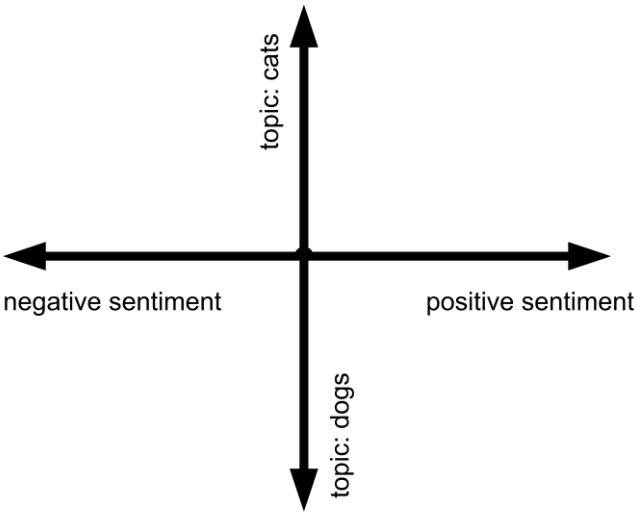


Figure 1: Visualization of the considered dimensions of the prototype.

3.1.1 Relevance Measure

The clustering algorithm takes as an input an initial ranking that is provided by a relevance measure. The state of the art in this field is extensively discussed in section 2.1. In our prototypical implementation Ontotext’s PageRank-related RDF rank[2] feature will be utilized for this task. However, as it is the task of research to enhance the state of the art, we discuss various case studies for this task in section 4. In fact, high quality input for the initial ranking is needed in order to produce valuable results [27].

[2] http://owlim.ontotext.com/display/OWLIMv50/OWLIM-SE+RDF+Rank, last checked on 15.05.2012

3.1.2 Diversity Measure

As mentioned in the beginning of this section, the prototype implements a clustering method called "Folding" and is introduced in [27].

At this point we will give a short summary of this approach:

Based on an initial ranking we select as a representative the highest ranked statement. Traversing the list of results, the remaining results are compared to the set of already selected representatives. New representatives are chosen if the statement is sufficiently different from all previous representatives. After this step, we have a set of representatives and the statements that were in the results set but not selected as representatives. For each of the remaining statements, we now choose as a cluster the representative that is closest to it. This leads to a non-fixed number of clusters.

The approach presented in [27] applies this clustering approach to images. In our case, of course, we cannot utilize the similarity function that has been introduced for the images. Therefore, we introduce a similarity measure that accounts for topic similarity as well as for sentiment similarity.

Topic Similarity:

Topics are attached to a statement via the property sioc:topic. Topics are represented as they are extracted by the fact mining toolkit (cf. D2.2.1).

Operating on structured data, we can easily introduce a very well known similarity measure that helps us to determine whether two statements cover the same topics.

We make use of the Jaccard[3] similarity; the function topics(x) retrieves the set of topics of the statement x:

$$Jacc(S_1, S_2) = \frac{|topics(S_1) \cap topics(S_2)|}{|topics(S_1) \cup topics(S_2)|}$$

Another important point is the similarity measure. Of course, the similarity measures for image similarity do not apply for our perception of diversity. As the current output of Enrycher only extracts topics (also in the sense of named entities) and sentiments, we start with a simple two fold approach.

Sentiment Similarity:

The sentiment score is extracted by the opinion mining toolkit (cf. D2.1.1). The representation of the sentiment score is via KDO by using the property kdo:hasScore. For determining sentiment similarity, we make use of a simple subtraction of the sentiment scores from each other. The function score(x) denotes the double sentiment score value of the sentiment of the statement x:

$$Sent(S_1, S_2) = 1 - |score(S_1) - \ score(S_2)|$$

Combining topic and sentiment similarity:

Both similarity scores, Jacc and Sent, range in the interval between 0 and 1 where 1 means full similarity and 0 means no similarity at all. Therefore, we can combine the two scores by simply averaging them:

$$SimAvg(S_1, S_2) = \frac{Jacc(S_1, S_2) + Sent(S_1, S_2)}{2}$$

More general, we can denote the similarity score by linearly combining them.

$$SimLin(S_1, S_2) = \ \gamma \ \times Jacc(S_1, S_2) + (1 - \ \gamma) \times Sent(S_1, S_2)$$

The restriction on this combination is: $0 \ \leq \gamma \ \leq 1$. Therefore, SimAvg can be represented by SimLin with $\gamma \ = 0.5$.

[3] http://en.wikipedia.org/wiki/Jaccard_index, last checked on 15.05.2012

3.1.3 Diversification objective

In the last step, we produce an interface where the user can put emphasis on one of the above mentioned diversity aspects.

In our prototype, we implement “emphasis” on one of the aspects with $\gamma\ = 0.75\ (\text{emphasis on topic})$ or $\gamma\ = 0.25(\text{emphasis on sentment})$. However, if this is an optimal estimate or if the users should be able to adjust the linear combination by themselves is due to evaluation.

3.2 RESTful Service for Diversity-Aware Ranking

The prototype of the ranking component is provided as a RESTful[4] service. The interface follows a SPARQL[5]-like query system. However, as we want to rank kdo:Statements, we provide only an interface for the WHERE clause. The restrictions (e.g. certain topics, authors, publication data ranges, etc.) have to be formulated on a fictive variable “**?s**”.

The RESTful service uses the HTTP GET method. The following parameters are mandatory:

- **endpoint** - defines the SPARQL endpoint which contains information according to the KDO ontology.
- **restrictions** - define restrictions (which are compliant to the syntax of the SPARQL "where" part) on the variable ?s which represents instances of kdo:Statement.

As mentioned above, we also want to provide the option to define a diversification objective. This is done by the following optional parameter:

- **rank** - defines an emphasis on a certain property (currently either kdo:hasSentiment or sioc:topic). The full URI for the property is needed e.g.

  "rank=http://kdo.render-project.eu/kdo#hasSentiment".

[4] http://en.wikipedia.org/wiki/RESTful, last checked on 15.05.2012

[5] http://www.w3.org/TR/rdf-sparql-query, last checked on 15.05.2012

4 Case studies: RDF ranking

The following subsections provide a detailed insight into case studies that were carried out in the context of exploring relevance and diversity ranking objectives. Parts of these case studies were also published.

4.1 Spreading Activation for RDF ranking

This section describes an approach for diversity-aware ranking based on spreading activation (SA) and clustering techniques. The developments presented here build on work done in LarKC EC project [16][39] and are aimed at exploring the potential of SA as a mechanism for diversity ranking, based on the idea that more typical items will get more activated than less typical ones and thus the level of activation can serve as a measure of typicality. The second idea, explored in this section, is to evaluate to what extent clustering information performed on RDF data, can give information about diversity, based on measuring similarity to a query on a topic and intra and inter cluster distances between the cluster members. A third approach combining SA and clustering, is to use the cluster information – similarity matrices, level of typicality of a cluster member, agglomerative hierarchical clustering information – as a basis for intra-cluster and inter-cluster SA.

4.1.1 DualRDF and PageRankRDF components in OWLIM

Two SA inspired mechanisms comes as standard OWLIM components – DualRDF and PageRankRDF [39]. The addition of such plug-ins in OWLIM was inspired by the goal to experiment with cognitively-inspired methods for selection and ranking based on popular connectionist approaches on top of RDF datasets. These components allow for evaluation of different activation implementations for diversity-aware ranking. The PageRank algorithm, implemented in OWLIM allows for ranking of nodes in large RDF datasets. It is an implementation of the original algorithm, intended to operate on web pages' graph (as nodes) and hyperlinks (as edges). The importance of a node is based on the importance of nodes which have links to it.

PageRankRDF is based on the counting of connections (predicates) between resources. A resource has a higher RDF rank if it is subject and object in many statements or/and if it has a high RDF ranked neighbour. OWLIM plug-in calculates RDF Rank values for the entire graph and the ranks values are available through a system predicate – http://www.ontotext.com/owlim/RDFRank#hasRDFRank.

For example, the computation of the RDFRanks for 400M LOD statements takes 310 seconds. RDF Rank values are not updated automatically and if a considerable change in the dataset has been done they have to be recomputed. The namespace for RDFRank can be found at http://www.ontotext.com/owlim/RDFRank#.

DualRDF implements the full functionality of standard SA with a full set of parameters which can be set by the user like activation thresholds and activation functions.

DualRDF (standard SA) and PageRankRDF are implemented as extensions the core of OWLIM – the TRREE engine. The configuration parameters required for the SA and PageRank features usage can be initiated and managed through SPARQL ASK queries. SA is performed from URIs referred to in a SPARQL query and returning as a result a tripleset, which contains the selected part of the graph.

DualRDF and PageRankRDF are powerful mechanisms which can rank RDF URIs on the bases of their connectivity with other URIs. If this level of connectivity (or available information) about a URI can characterize the diversity of the data, these two mechanisms can be used for diversity ranking.

4.1.2 Fast Approximate SA Approaches

Although the DualRDF component implements full standard SA with all its virtues it cannot be used for very large dataset due to speed limitations [39]. In order to remedy to this the so-called Node Selection based SA (NSbSA) [16][39] was developed. It takes advantage of the sparsity of the nodes' connectivity (node-predicate-node) matrix described above and the existing formats for compact representation of sparse matrices. NSbSA is based on non-zero elements finding and implements path finding in a graph defined by the nodes' connectivity matrix (similar to a breadth-first-search algorithm in graphs). This procedure results in a list of 'activated' nodes and the number of time they have been encountered. Thus, NSbSA selects the nodes which would have been activated at some point in a standard SA process. The number of times a node is reached via different connections is an estimate of its level of activity. The method can also use a different connection matrix, for instance derived from a similarity matrix or from a clustering of the dataset. Details and some computational explorations of the method can be found in [16] and [39]. The method has been successfully tested using FactForge with hundreds of millions of statements (http://factforge.net) with parallel computations on NVIDIA CUDA card.

The efficiency of NSbSA is based on a sparse numerical representation of the original nodes' connectivity matrix, which contains only 1's as weights. This matrix can be extracted off-line using the statements from the dataset. The SPARQL query plays the role of a source of activation and is represented by a vector with ones at the places of the URIs from the query.

The NSbSA method can be summarized as follows:

- Instead of matrix-vector multiplication used in SA approaches implements SA by searching for non-zero elements in a sparse vector;
- This is a process of following the connections of the nodes from the query (the seeds), finding the nodes they are connected to, then repeat this procedure with the newly found nodes;
- This will lead to newly selected nodes and the process is repeated iteratively;
- The efficiency is related to the use of each node which is a source of activation *only once*.

In the RENDER project, the previously existing experimental prototype for NSbSA has been completely reimplemented using the new GPU BFS library [29]. Moreover, the reimplementation includes fan-out and decay effects important for the use of SA for selection and similarity as well as some additional mechanisms shown to be essential in [41][4]. The computation speed achieved is of the order of a hundreds of millions of statements per second.

As discussed above, SA can be useful in diversity-aware ranking only if the weights or the connection matrix of the dataset needed for SA can reflect this aspect of the data. When discussing sets of RDF triples the number and type of connections a resource is involved in can reflect the specificity of the data and thus its diversity. However, SA is based on the number of connections (if there is no way to weigh them) and URIs with similar level of connectivity will tend to have the same activation if there is a path relating them to the query and thus they cannot be distinguished based on activation. Activation would work in cases when some type of content is described with a relatively large number of triples as opposed to other content described with a relatively small number of triples.

One way to find a connectivity matrix which reflects the diversity of the data is to apply clustering techniques which group the data based on some appropriate similarity measure. This approach would lead to clusters whose relative distance and the distances among their members can be used as weights [15][39]. How clustering can be used for diversity ranking is discussed in the next section.

4.2 Clustering for RDF ranking

Clustering is based on the assumption that meaningful clusters of nodes can be found in a dataset. Such clusters can be based on the dataset RDF triples, based on the various existing techniques for evaluating the similarity between the nodes e.g. based on their degree of connectivity, their type and role in the statement, taxonomical organization, structure of the matrix, predicates, etc. On the other hand it can fetch semantic or associative information not contained in the dataset alone, e.g. by using Wordnet (http://wordnet.princeton.edu), text mining techniques like LSA, or some prebuilt classification structure as in DBpedia ontology (http://dbpedia.org) .

When various clustering schemes for a single dataset exist, depending on the nature of the clusters, each node or predicate can be related to the cluster it belongs to and furthermore the distance to the cluster center can be a measure of how typical it is. Using these measures for the dataset, or parts of the dataset (e.g. tweets, e-mails, etc.) one can attribute a level of typicality which in some cases can be inversely proportional to its diversity.

On the other hand such levels of typicality (or centrality) can be used in SA as weights. This is the approach of the so-called Cluster based SA (CbSA) [15][39]. The basic idea of this approach is to substitute the very high dimensional connectivity matrix based on predicates with a lower dimensional and more relevant weight matrix (Figure 2). As the number of clusters is expected to be much less than the number of nodes, clustering will strongly reduce the size of the connectivity matrix and increase the computational efficiency of the respective SA implementation.

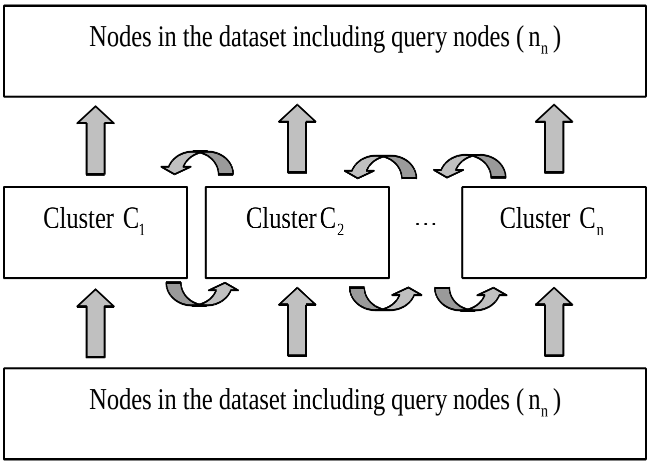

Nodes in the dataset including query nodes (n_n)
Cluster C_1
Cluster C_2
…
Cluster C_n
Nodes in the dataset including query nodes (n_n)

Figure 2: Cluster based SA (CbSA).

Types of clustering that can be interesting for diversity-aware ranking are the following:

- Clusters of nodes based on connectivity;
- Clusters based on sharead predicates;
- Clusters in the dataset based on dense interconnection;
- Semantic associations;
- Wordnet based similarity;
- Taxonomic and relational similarity.

4.3 Examples of RDF SA and clustering: Preliminary results

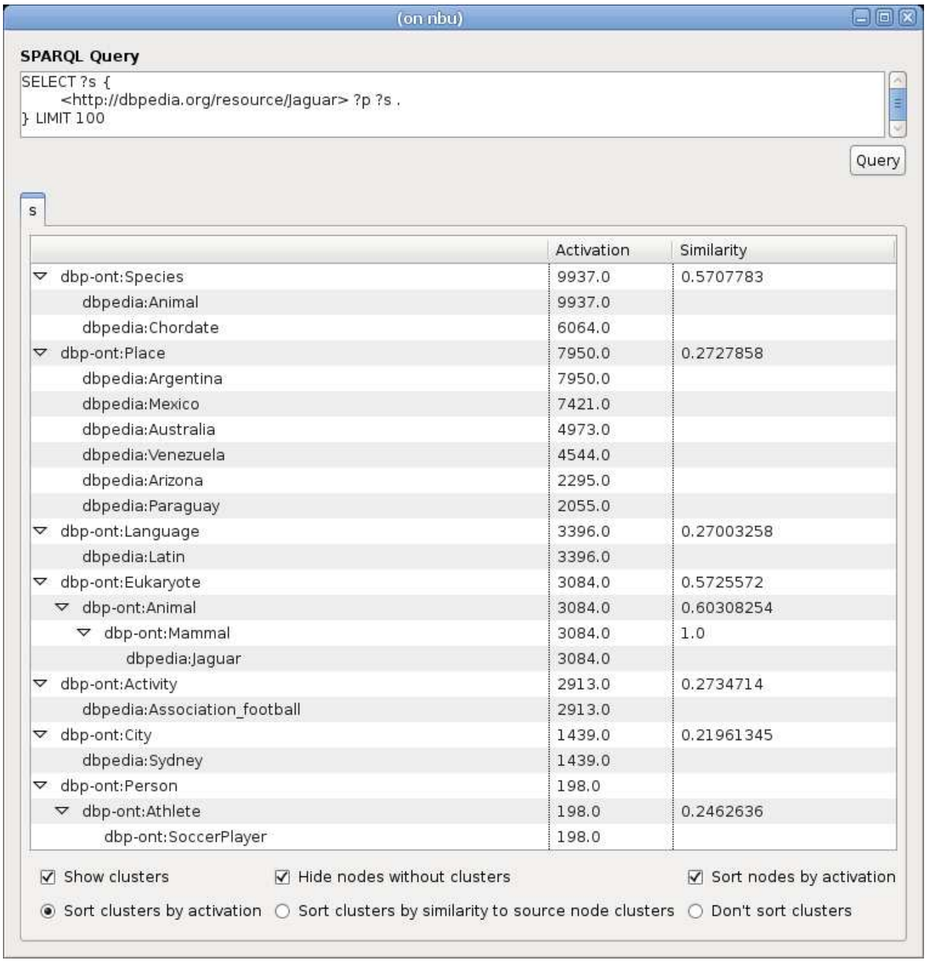
In this subsection some preliminary exploration of the approaches to diversity ranking introduced in this section will be presented. So far, the datasets from RENDER use cases are not available in RDF form and cannot be used. So, we have used other datasets like the DBpedia ontology and a large recipe dataset. DBpedia ontology is appropriate for testing cluster based approaches as it has a clear class structure which ensures good clustering. The recipe dataset represents a set of recipes (~280, 000) with the respective foods for each of them (~8 on average per recipe and ~270 in total).

4.3.1 DBpedia example

The queries in this example contain a single URI – http://dbpedia.org/resource/jaguar and http://dbpedia.org/resource/Friedrich_Nietzsche. The following quantities were calculated: activation using the RDF triples connectivity matrix (the predicates are edges in this representation), the clustering of DBpedia ontology which is identical to DBpedia classes and the similarity of the query URI to the URIs retrieved. The similarities are calculated using Wordnet synset similarities (only URIs with existing mapping to Wordnet have been used).

The prototype testing tool implements a Java based interface with SPARQLE end point and DBpedia as a single dataset. The results corresponding to a query are structured based on existing clusters they belong to and ranked with respect to activation or similarity to the query.

Two examples of such queries are given in Figure 3 and Figure 4. It is seen that the more typical association like species, habitat, etc. have a higher activation than the less typical related to language or person (seeFigure 3). ‘Mammal’ has similarity 1.0 whereas ‘Place’ and ‘City’ have much lower similarity (0.27 and 0.22, respectively) indicating a more distant context for ‘jaguar.’



SPARQL Query

```
SELECT ?s {
      <http://dbpedia.org/resource/Jaguar> ?p ?s .
} LIMIT 100
```

	Activation	Similarity
▽ dbp-ont:Species	9937.0	0.5707783
dbpedia:Animal	9937.0	
dbpedia:Chordate	6064.0	
▽ dbp-ont:Place	7950.0	0.2727858
dbpedia:Argentina	7950.0	
dbpedia:Mexico	7421.0	
dbpedia:Australia	4973.0	
dbpedia:Venezuela	4544.0	
dbpedia:Arizona	2295.0	
dbpedia:Paraguay	2055.0	
▽ dbp-ont:Language	3396.0	0.27003258
dbpedia:Latin	3396.0	
▽ dbp-ont:Eukaryote	3084.0	0.5725572
▽ dbp-ont:Animal	3084.0	0.60308254
▽ dbp-ont:Mammal	3084.0	1.0
dbpedia:Jaguar	3084.0	
▽ dbp-ont:Activity	2913.0	0.2734714
dbpedia:Association_football	2913.0	
▽ dbp-ont:City	1439.0	0.21961345
dbpedia:Sydney	1439.0	
▽ dbp-ont:Person	198.0	
▽ dbp-ont:Athlete	198.0	0.2462636
dbp-ont:SoccerPlayer	198.0	

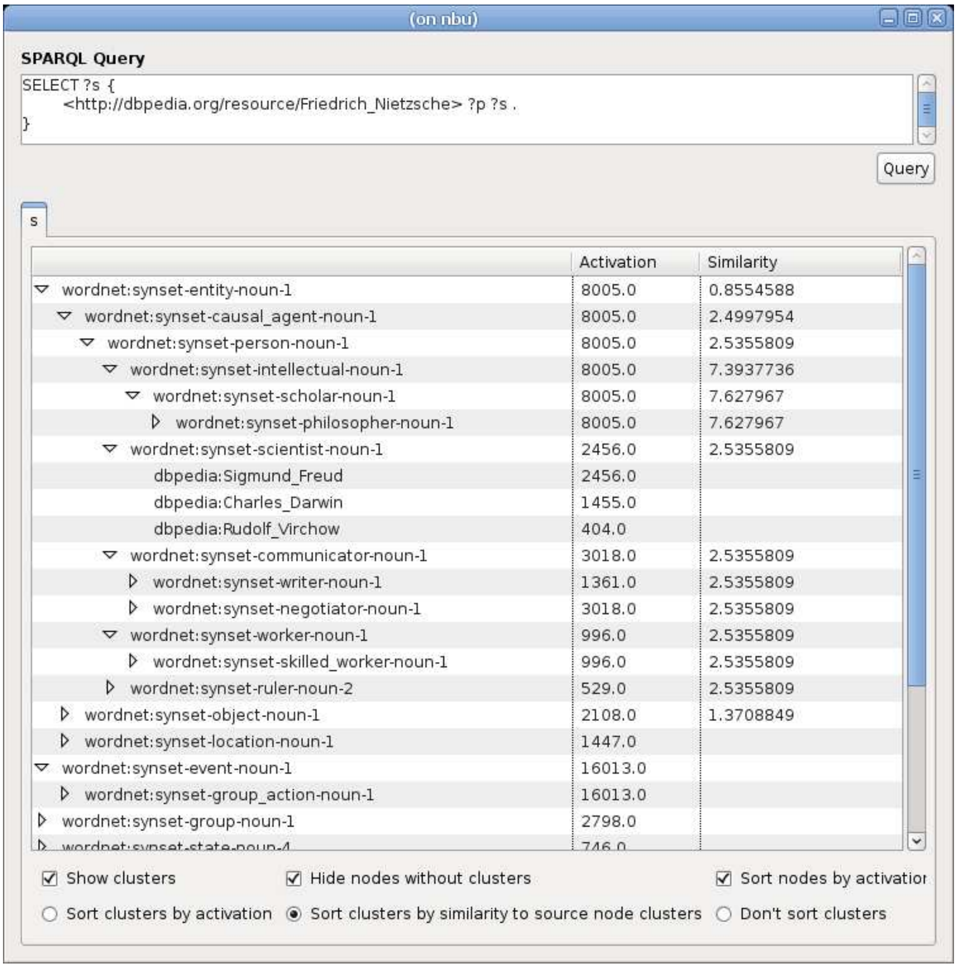
☑ Show clusters ☑ Hide nodes without clusters ☑ Sort nodes by activation

● Sort clusters by activation ○ Sort clusters by similarity to source node clusters ○ Don't sort clusters

Figure 3: Results from http://dbpedia.org/resource/jaguar URI with clusters sorted by decreasing mean activation. Nodes withing the clusters are also sorted by decreasing activation.

Similarly, for http://dbpedia.org/resource/Friedrich_Nietzsche (see Figure 4), the largest similarity is obtained for the Wordnet synset containing ‘philosopher’ (7.6). Less similar with the query are the synsets of ‘scientist’ and ‘communicator’ (2.5).

Although, more tests are needed to assess the full potential of this approach it seems quite promising in the combination of SA mechanism based on the RDF dataset and external associative strength based on Wordnet.



SPARQL Query

```
SELECT ?s {
      <http://dbpedia.org/resource/Friedrich_Nietzsche> ?p ?s .
}
```

	Activation	Similarity
▽ wordnet:synset-entity-noun-1	8005.0	0.8554588
▽ wordnet:synset-causal_agent-noun-1	8005.0	2.4997954
▽ wordnet:synset-person-noun-1	8005.0	2.5355809
▽ wordnet:synset-intellectual-noun-1	8005.0	7.3937736
▽ wordnet:synset-scholar-noun-1	8005.0	7.627967
▷ wordnet:synset-philosopher-noun-1	8005.0	7.627967
▽ wordnet:synset-scientist-noun-1	2456.0	2.5355809
dbpedia:Sigmund_Freud	2456.0	
dbpedia:Charles_Darwin	1455.0	
dbpedia:Rudolf_Virchow	404.0	
▽ wordnet:synset-communicator-noun-1	3018.0	2.5355809
▷ wordnet:synset-writer-noun-1	1361.0	2.5355809
▷ wordnet:synset-negotiator-noun-1	3018.0	2.5355809
▽ wordnet:synset-worker-noun-1	996.0	2.5355809
▷ wordnet:synset-skilled_worker-noun-1	996.0	2.5355809
▷ wordnet:synset-ruler-noun-2	529.0	2.5355809
▷ wordnet:synset-object-noun-1	2108.0	1.3708849
▷ wordnet:synset-location-noun-1	1447.0	
▽ wordnet:synset-event-noun-1	16013.0	
▷ wordnet:synset-group_action-noun-1	16013.0	
▷ wordnet:synset-group-noun-1	2798.0	
▷ wordnet:synset-state-noun-4	746.0	

☑ Show clusters ☑ Hide nodes without clusters ☑ Sort nodes by activatior

○ Sort clusters by activation ● Sort clusters by similarity to source node clusters ○ Don't sort clusters

Figure 4: Results from http://dbpedia.org/resource/Friedrich_Nietzsche URI with clusters sorted by decreasing mean similarirty. Nodes within the clusters are also sorted by decreasing activation.

4.3.2 Recipe dataset example

The recipe dataset consists in a collection of recipes, collected on the Web for which the foods, the treatments and some additional data are known. For the purposes of the example only a set of 287 foods like fish, beef, etc. has been selected and 282, 332 recipes.

The recipes are characterized by the foods used in their preparation. The Cluto library [22] was used to cluster the dataset. Experimentation with the number of clusters and the value of the cluster quality criterion led to a clustering with 300 clusters. The Cluto library provides an estimation of the level of belonging of an object to the cluster it belongs to [22]. This quantity gives (obtained when the '-zscores' parameter is set) the relative mean similarity of a cluster element to all other element in the cluster with respect to the mean value of this quantity for the cluster. Elements for which this value is larger are closer to the center of the cluster. Table 1 gives information about Cluster 109 of fish recipes with parsley as a typical example from the recipe dataset.

Table 1: Characteristics of Cluster 109 from the recipe dataset obtained with Cluto.

<table>
<tr><th colspan="2">Cluster 109, Size: 664, ISim: 0.562, ESim: 0.089</th></tr>
<tr><th>Percentage of recipes sharing a set of foods</th><th>Descriptive and Discriminating foods in the cluster</th></tr>
<tr><td>30.57% Fish Parsley Lemons Oil Salt Spices
15.51% Fish Parsley Lemons Oil Garlic Salt Spices
16.57% Fish Parsley Salt Butter Spices</td><td><b>Descriptive</b>: Fish 52.0%, Parsley 28.5%, Lemons 6.8%, Oil 2.1%, Garlic 1.9%, Bread 1.5%, Salt 0.9%, Butter 0.8%, Spices 0.8%, Onion 0.6%
<b>Discriminating</b>: Fish 37.3%, Parsley 14.9%, Sugars</td></tr>
</table>

16.27% Fish Parsley Bread Salt Spices	4.1%, Egg 2.4%, Cheese 2.4%, Water 1.9%, Vanilla 1.9%, Macaroni 1.9%, Milk 1.8%, Leavening 1.7%

Similarity to Center	**Cluster 109, Size: 664, ISim: 0.562, ESim: 0.089**
1.7	Oven baked salmon
1.7	Roasted rainbow trout with lemon and thyme
1.7	Lemon cod
…	…
-2.7	Stuffed sardines in grape leaves
-3.2	Brandade de mourue
-3.4	Flounder rolls

As a second step, we used the clustering information for SA. This was done as follows. Cluto can build an agglomerative hierarchical tree with leaves the clusters found. It also calculates the similarity between adjacent clusters.

Starting from a recipe, its cluster center is activated using as weight the similarity of the recipe to the cluster center. Then, each cluster member is activated using the cluster center activation and their similarities to the center as weights. The average activation of the cluster members is considered to be the activation of the cluster. Then, using the similarities between adjacent clusters provided by Cluto, the adjacent cluster centers are activated. Then their members are activated using their similarities to the cluster center as weights. This process is repeated for a number of iterations with a decay which is a parameter. The result of this procedure will be a number of clusters which are activated depending on their similarity to the initial recipe. The size, density, and distribution of elements of the clusters will be a measure of the diversity of the content corresponding to this cluster. The diversity defined in this way will be dependent on the features used for the clustering.

For each cluster, Cluto gives the foods which are shared by the majority of the cluster members (Table 1). Those members which share less of these features and more from the remaining features well differ more from the core members of the cluster. Starting from a recipe or a set of foods and using them as a source of activation, we can activate all the clusters and their members having something in common with them (foods or similar recipes) and the activation will be the inverse of a sort of diversity rank with respect to the query and existing clustering (set of features). Part of the result of such a CbSA starting with a member recipe from cluster 109 is shown in Table 2.

Table 2: Clusters activated by activating the recipe closest to the center of cluster 109. (Each recipe is described by the set of foods which are shared by most recipes in the cluster.)

Activation	Clusters
0.42	**30.57% Fish Parsley Lemons Oil Salt Spices (cl. 109)**
0.41	34.98% Fish Oil Spices Salt
0.26	86.66% Fish Wine Spices
0.20	40.37% Olives Cheese Spices
0.16	63.61% Capers Olives Garlic Oil
0.13	22.50% Capers Parsley Lemons Oil

0.11	44.05%	Capers Fish Lemons Oil
0.09	44.40%	Olives Fish Oil Garlic
0.06	77.08%	Olives Parsley Oil

If we assume that a similar procedure can be carried on with tweets for a cluster of tweets we can thus obtain clusters of similar tweets.

The results presented show that the SA and CbSA approaches have the potential to be applied for diversity aware ranking and should be explored further with datasets from Render use cases.

4.4 Leveraging Usage Data for the Ranking of Entity Features

So far we have considered different ways of ranking information that is represented in KDO. The contribution of this section summarizes the paper "Leveraging Usage Data for Linked Data Movie Entity Summarization"[6] and explains potential uses for diversity-enabled semantic document ranking. The paper can be found Annex A.

In the aforementioned publication we focused on establishing similarity between Linked Data entities by analysing usage behaviour. The idea is to exploit the "wisdom of the crowd" in order to gain insights on which features of entities are of particular importance for individuals.

This technique can easily be exploited in RENDER: Tweets and Wikipedia articles are not only browsed but also republished (i.e. re-tweets in Twitter) or edited (i.e. article edits in Wikipedia). Therefore, we can argue that different tweets/articles that are republished/edited by a similar set of people might have a specific set of diversity features in common. In the following, we provide two examples where this type of feature ranking is applicable for the RENDER use cases:

- **Movistar**

  Through the re-tweet behaviour of users we can derive that some posts are more similar than others. Comparing the semantics of one of these posts with the semantics of its neighbours, we find out that for this post the feature sentiment ("positive") and topic ("Movistar")[7] are of particular importance. By summing up feature importance we can get a entity ranking.

- **Wikipedia administrative departments**

  Some articles are edited by a particular set of people e.g. people living in the same department have very good knowledge about the towns, municipalities and villages in it. Comparing articles that include the implicit similarity through editing behaviour, we discover that many of them share

[6] Published in the proceedings of the USEWOD Workshop held at the 21st International Conference on World Wide Web (WWW), 2012 (cf. [1]).

[7] http://topics.render-project.eu/telefonica#Movistar

References

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, Samuel Ieong: Diversifying search results. Proceedings of the Second ACM International Conference on Web Search and Data Mining, ACM, NY USA. 2009

[2] Anyanwu, K., Maduko, A., & Sheth, A. (2005). Semrank: ranking complex relationship search results on the semantic web. *Proceedings of the 14th international conference on World Wide Web* (pp. 117-127). ACM.

[3] Baeza-Yates, R., & Davis, E. (2004). Web page ranking using link attributes. Proceedings of the 13th international World Wide Web conference on Alternate track papers. ACM.

[4] M. R. Berthold, U. Brandes, T. Kötter, M. Mader, U. Nagel, and K. Thiel, “Pure spreading activation is pointless,” in Proceedings of the CIKM the 18th Conference on Information and Knowledge Management, 2009, pp. 1915–1919.

[5] Broder, A., Lempel, R., Maghoul, F., & Pedersen, J. (2006). Effcient pagerank approximation via graph aggregation. *Information Retrieval 9*, 123-138.

[6] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, Volume 41, Issue 6*, 391-407.

[7] Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., & Decker, S. (2010). Hierarchical Link Analysis for Ranking Web Data. *In Proceedings of the 7th Extended Semantic Web Conference.* ESWC.

[8] Ding, C., He, X., Husbands, P., Zha, H., & Simon, H. (2001). *Link analysis: Hubs and authorities on the world.* Technical report: 47847.

[9] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, S., Peng, Y., et al. (2004). Swoogle: a search and metadata engine for the semantic web. *CIKM*.

[10] Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., & Kolari, P. (2005). Finding and ranking knowledge on the semantic web. *Proceedings of the International SemanticWeb Conference*, (pp. 156-170).

[11] Marina Drosou, Evaggelia Pitoura: Search result diversification. ACM SIGMOD Record archive Volume 39 Issue 1, March 2010. ACM New York, USA. 2010

[12] Eiron, N., McCurley, K., & Tomlin, J. (2004). Ranking the Web Frontier. *Proceedings of the 13th conference on World Wide Web Number 2* (pp. 309-318). New York, NY, USA: ACM Press.

[13] Sreenivas Gollapudi, Aneesh Sharma: An Axiomatic Approach for Result Diversification. Proceedings of the 18th international conference on World Wide Web (WWW ’09), ACM, NY USA. 2009

[14] Greengrass, E. (2000). *Information Retrieval: A survey.* DOD Technical Report TR-R52-008-001.

[15] Maurice Grinberg and Hristo Stefanov (2012). Spreading activation and clustering techniques for RDF ranking. To be submitted to ICSC 2012.

[16] Grinberg, M., Haltakov, V., Stefanov, H. (2010) Spreading Activation Mechanisms for Efficient Knowledge Retrieval form Large Datasets. In: Proceedings of WIRN 2010, COST 2102 Special session. IOS press.

[17] Gunnar Astrand Grimnes, Peter Edwards, and Alun Preece (2008). Instance Based Clustering of Semantic Web Resources. In: S. Bechhofer et al. (Eds.): ESWC 2008, LNCS 5021, pp. 303–317, Springer-Verlag, Berlin Heidelberg.

[18] Guha, R. V., McCool, R., & Miller, E. (2003). Semantic search. *In Proc. of the 12th Intl. World Wide Web Conference (WWW 2003)* (pp. 700-709). Budapest, Hungary: WWW 2003.

[19] Jayant R. Haritsa: The KNDN Problem: A Quest for Unity in Diversity. IEEE Data Eng. Bull. 32(4): 15-22 (2009). 2008.

[20] Andreas Hotho, Alexander Maedche, and Steffen Staab (2001). Ontology-based Text Clustering. Workshop "Text Learning: Beyond Supervision", IJCAI 2001.

[21] Kamvar, S., Haveliwala, T., Manning, C., & Golub, G. (2003). *Exploiting the block structure of the web for computing pagerank.* Stanford InfoLab.

[22] Karypis, G. (2003). CLUTO: A Clustering Toolkit, http://www.cs.umn.edu/~karypis/cluto/.

[23] Kleinberg , J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM 46(5)*, 604–632.

[24] Lee, D. L., Chuang, H., & Seamons, K. (1997). Document ranking and the Vector-Space model. *IEEE Software, Volume 14, Issue 2*, 67 - 75.

[25] Lee, J. (1994). Properties of extended boolean models in information retrieval. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 182 - 190). ACM.

[26] Lempel, R., & Moran, S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *In The Ninth International WWW Conference.*

[27] Reinier H. van Leuken, Lluis Garcia, Ximena Olivares, Roelof van Zwol: Visual Diversification of Image Search Results. Proceedings of the 18th international conference on World Wide Web (WWW '09), ACM, NY USA. 2009

[28] Chuan Lin, Young-rae Cho, Woo-chang Hwang, Pengjun Pei, and Aidong Zhangin (2007). Clustering methods in protein-protein interaction network. In: Xiaohua Hu and Yi Pan (Eds.), Knowledge Discovery in Bioinformatics: Techniques, Methods and Applications, ISBN: 047177796X, John Wiley & Sons Inc.

[29] Merrill, D., Garland, M., and Grimshaw, A. (2011) High Performance and Scalable GPU Graph Traversal. Technical Report CS2011-05, Department of Computer Science, University of Virginia. Aug. 2011.

[30] Enrico Minack, Gianluca Demartini , Wolfgang Nejdl: Current Approaches to Search Result Diversification. Proc. of 1st Intl. Workshop on Living Web, 2009

[31] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web.* Stanford Digital Libraries SIDL-WP-1999-0120.

[32] Ponte, J., & Croft, W. (n.d.). A language modeling approach to information retrieval. *In Proceedings of the 21st ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 275 - 281). ACM.

[33] Rahayu, S., & Noah, S. (2011). Annotated document: Scoring and ranking method. *Semantic Technology and Information Retrieval (STAIR), 2011 International Conference on* (pp. 167 - 170). Conference Publications.

[34] Ridings , C., & Shishigin, M. (2002). *Pagerank uncovered.*

[35] Rocha, C., Schwabe, D., & Poggi de Aragão, M. (2004). A hybrid approach for searching in the semantic web. *International World Wide Web Conference, Proceedings of the 13th international conference on World Wide Web*, (pp. 374 - 383).

[36] Rode, H., & Hiemstra, D. (2005). Conceptual language models for Context-Aware text retrieval. *Proceedings of the 13th Text Retrieval Conference (TREC).* NIST Special Publications.

[37] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24(5)*, pp. 513-523.

[38] Shamsfard, M., Nematzadeh, A., & Motiee, S. (2006). ORank: An Ontology Based System for Ranking Documents. *International Journal of Computer Science, vol .1*, pp.225- 231.

[39] Spreading activation components (v. 1-3) LacKC EC project deliverables D2.4.1, D2.4.2, and D2.4.3 http://www.larkc.eu/resources/deliverables/

[40] Andreas Thalhammer, Ioan Toma, Antonio J. Roa-Valverde, Dieter Fensel: Leveraging Usage Data for Linked Data Movie Entity Summarization. Proceedings of the 2nd International Workshop on Usage Analysis and the Web of Data (USEWOD2012) in the 21st International World Wide Web Conference (WWW2012), arXiv:1204.2718v1, Lyon, France, April 17th, 2012

[41] Kilian Thiel and Michael R. Berthold (2010). Node Similarities from Spreading Activation. In: G. I. Webb (Ed.), Data mining ICDM 2010, pp. 1080-1090.

[42] Vallet, D., Fernández, M., & Castells, P. (2005). An Ontology-Based information retrieval model. *2nd European Semantic Web Conference* (pp. 455-470). Heraklion, Greece: Springer Verlag Lecture Notes in Computer Science, Volume 3532.

[43] Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., Yahia, S.A: Efficient Computation of Diverse Query Results. Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on , vol., no., pp.228-236, 7-12 April 2008

[44] Wang, J., Chen, Z., Tao, L., Ma, W., & Liu, W. (2002). Ranking user's relevance to a topic through link analysis on web logs. *WIDM*, 49–54.

[45] Wang, Y., & DeWitt, D. (2004). Computing pagerank in a distributed internet search system. *Proceedings of the Thirtieth international conference on Very large data bases* (pp. 420-431). Toronto, Canada: VLDB Endowment.

[46] Woods, W. A., Bookman, L. A., Houston, A., Kuhns, R. J., Martin, P., & Green, S. (2000). Linguistic knowledge can improve information retrieval. *Applied Natural Language Conferences, Proceedings of the Sixth Conference on Applied Natural Language Processing*, (pp. 262-267).

[47] Xing, W., & Ghorbani, A. (2004). Weighted pagerank algorithm. Proceedings of the Second Annual Conference on Communication Networks and Services Research.

[48] Xue, G., Yang, Q., Zeng, H., Yu, Y., & Chen, Z. (2005). Exploiting the hierarchical structure for link analysis. *Proceedings of the 28th annual international ACM SIGIR conference* (pp. 186-193). New York, NY, USA: ACM .

[49] Cheng Xiang Zhai, William W. Cohen, John Lafferty: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM New York, NY, USA. 2003

[50] Zhuge, H., & Zheng, L. (2003). Ranking Semantic-Linked network. *WWW (Posters)*.

Annex A Leveraging Usage Data for Linked Data Movie Entity Summarization

Leveraging Usage Data for Linked Data Movie Entity Summarization

Andreas Thalhammer, Ioan Toma, Antonio J. Roa-Valverde, Dieter Fensel
Semantic Technology Institute
University of Innsbruck
Technikerstraße 21a
6020 Innsbruck, Austria
{firstname.lastname}@sti2.at

ABSTRACT

Novel research in the field of Linked Data focuses on the problem of entity summarization. This field addresses the problem of ranking features according to their importance for the task of identifying a particular entity. Next to a more human friendly presentation, these summarizations can play a central role for semantic search engines and semantic recommender systems. In current approaches, it has been tried to apply entity summarization based on patterns that are inherent to the regarded data.

The proposed approach of this paper focuses on the movie domain. It utilizes usage data in order to support measuring the similarity between movie entities. Using this similarity it is possible to determine the k-nearest neighbors of an entity. This leads to the idea that features that entities share with their nearest neighbors can be considered as significant or important for these entities. Additionally, we introduce a downgrading factor (similar to TF-IDF) in order to overcome the high number of commonly occurring features. We exemplify the approach based on a movie-ratings dataset that has been linked to Freebase entities.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human factors—*Usage Data Mining*; H.3.5 [**On-line Information Services**]: Data sharing—*Linked Open Data*

General Terms

Human Factors, Experimentation

Keywords

linked data, entity summarization, ranking, item similarity

1. INTRODUCTION

Linked Data, which connects different pieces of machine-readable information (resources) via machine-readable relationships (properties) has rapidly grown in the past years, changing the way data is published and consumed on the Web. Data referring to real-world entities is being linked resulting into vast network of structured, interlinked descriptions that can be used to infer new knowledge. The rapid growth of Linked Data (LD) introduces however a set of new challenges. One in particular becomes very important when it comes to characterizing real world entities: their LD descriptions need to be processed and understood quickly and effectively. The problem known as entity summarization [5] is concerned with identifying the most important features of lengthy LD or Linked Open Data (LOD)[1] descriptions. Solutions to this problem help applications and users of LD to quickly and effectively understand and work with the vast amount of data from LOD cloud.

In this paper we propose a novel approach that leverages usage data in order to summarize entities in the LOD space. More precisely, we perform data analysis on LD in order to identify features of entities that best characterize them. Our approach is simple and effective. We first measure similarities between entities and identify a set of nearest neighbors for each entity. For each feature of the entity we then count the number of entities having the same feature in the nearest neighbors group as well as in the set of all entities. Based on this we compute a weight for each entity, order the entities descending and select the top-n features as the summarization for each entity. To validate our approach we run a set of experiments using two datasets, namely the HetRec2011 MovieLens2k dataset [4] and data crawled from Freebase.[2] Results obtained from these datasets show that our approach is capable to identify relevant features that are shared with similar entities and thus provide meaningful summarizations.

The remainder of this paper is organized as follows. Section 2 details our approach on leveraging usage data for linked data movie entity summarization. Section 3 presents the related work in the areas of entity summarization, usage mining and semantic representation of user profiles. Section 4 introduces the datasets used in our experiments while Section 5 discusses the preliminary results obtained, focusing more on the neighborhood formation and neighborhood-based entity summarization results. Finally, Section 6 concludes the paper and Section 7 outlines future work that we plan based on the approach presented in this paper.

Please note, we use the terms item and entity interchangeable in this paper.

2. PROPOSED APPROACH

The main idea introduced in this work is that property-value pairs - consecutively also called features - that an entity shares with its k-nearest neighbors are more relevant than features that are shared with entities that are not in the k-nearest neighbors range. Figure 1 visualizes this situ-

[1] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[2] http://freebase.com/

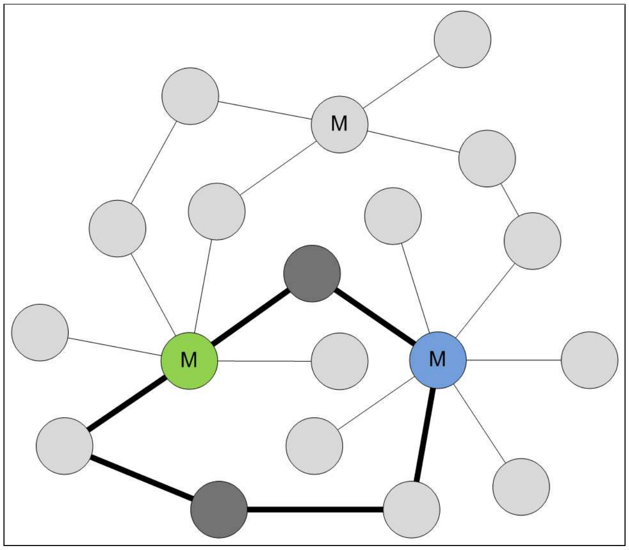


Figure 1: Visualization of shared features (strong lines and dark gray nodes) between k-nearest neighbors (green and blue nodes).

ation. Two nodes (green and blue) of the same type (M) are in each other's neighborhood. The features shared with each other (strong lines and dark gray nodes) are considered to be more important for their idendity than features they share with a node (light gray M) that is not in their respective neighborhood. The neighborhood formation of each node is based on usage data.

A detailed problem statement of entity summarization is given in [5]. The authors of this paper define the summarization of an entity e as follows:

> "Given $FS(e)$ and a positive integer $k < |FS(e)|$, the problem of entity summarization is to select $Summ(e) \subset FS(e)$ such that $|Summ(e)| = k$. $Summ(e)$ is called a summary of e."[3]

$FS(e)$ denotes the feature set of a given entity e. More informally, the feature set of an entity e is defined as the property-value pair set of e. An example for such a property-value pair for the entity fb : *en.toy_story*[4] is:

```
(fb:film.film.production_companies, fb:en.pixar)
```

In the following, E denotes the set of all entities. Our approach to provide a summarization of a given entity $e \in E$ is based on usage data and includes six steps:

1. Generate the user-item matrix.

2. Measure the similarity between e and other items and identify a set $N_{k,e} \subseteq E$ of k-nearest neighbors of e.

3. For each feature $f \in FS(e)$ collect the items $A_{e,f} \subseteq N_{k,e}$ that share the same feature.

4. For each feature $f \in FS(e)$ collect the items $B_{e,f} \subseteq E$ that share the same feature.

5. The weight w of f is the following ratio:

$$w_e(f) = |A_{e,f}| \times \log \frac{|E|}{|B_{e,f}|}$$

6. Order the features $f \in FS(e)$ descending according to their given weight $w_e(f)$. Select the n most relevant features as a summarization of e.

The concept of a user-item matrix (step 1) is a well-known principle in the field of recommender systems. Each column of the matrix represents a single item and each row represents a single user. The entries of the matrix are either the ratings (a numerical score) or empty if a user has not rated a particular item (which is the standard case). The column or row vectors can be used to compare items or users amongst each other respectively. For this, several similarity measures have been introduced of which cosine similarity and Pearson correlation (comparing the vectors with regard to their angular distance) are the most common techniques [1].

In our current implementation, we apply the *log-likelihood ratio score* [8] for computing item similarity (step 2). In the context of item similarity, the ratio takes into account four parameters: the number of users who rated both items, the number of users who rated the first but not the second item and vice versa, and the number of users who rated none of the two items. Note that this similarity measure does not consider the numerical values of the ratings and therefore also works with binary data like web site visits.[5] Finally, with the similarity scores it is easy to identify a set of k-nearest neighbors (kNN) for a given item.

Listing 1 states a SPARQL[6] query that is used for the retrieval of common features (property-value pairs) between the item (`fb:movie.uri`) and its 20 nearest neighbors (step 3). For measuring the similarity to all items in the dataset (step 4), the same query can be executed but without line 3. For each of the two result sets, the property-value pairs can be counted by occurrence. The filter rule (line 7) filters out property-value pairs that stem from the given entity (`fb:movie.uri`). Additionally, we also filter out the commonality of similar nearest neighbors because those features were added in the course of applying the approach and do not contribute to the summarization of the given entity.

In the result set of the nearest neighbors, a lot of features are frequently occurring; such as the following property-value pair:

```
(fb:film.film.country, fb:en.united_states)
```

If the weighting involved only counting, features like the above would be considered as highly relevant for many movies. However, as these features do not only occur often in the neighbors set but also in the overall set, they can be downgraded (step 5). As for the downgrading technique, we use the idea of the classic information retrieval method *term frequency - inverse document frequency* (TF-IDF). In our case

[3] In our approach, k is already used for the k-nearest neighbors method. Therefore, we refer to the cardinality of the summarization as n.

[4] fb denotes the Freebase namespace: `http://rdf.freebase.com/ns/`

[5] This is the reason why we refer to the term "usage data" rather than "rating data": we conclude usage from the process of giving a rating. We do not consider the numerical values of the ratings.

[6] SPARQL W3C Recommendation - `http://www.w3.org/TR/rdf-sparql-query/`

Listing 1: SPARQL query: retrieving property-value pairs shared with at least one of the 20-nearest neighbors.

```
select ?p ?o where {
fb:movie.uri ?p ?o.
fb:movie.uri knn:20 ?s.
?s ?p ?o.
?s rdf:type fb:film.film.
FILTER((?s != fb:movie.uri) && (?p != knn:20))
}
```

a “term” is stated by a single feature and the term frequency is the frequency of the feature in the nearest neighbors set. After this step, every feature that is shared with at least one of the k-nearest neighbors has an assigned weight.

Finally, in step 6, we select the n most relevant property-value pairs in accordance to their weight.

3. RELATED WORK

In the field of entity summarization, initial work has been presented in [5], where an approach called RELIN is introduced. The authors apply an adapted version of the random surfer model[7] - called goal directed surfer - in order to combine informativeness and relatedness for the ranking of features. In the conclusion, it is stated that a “user-specific notion of informativeness (...) could be implemented by leveraging user profiles or feedback” in order to mitigate the issue of presenting summarizations that help domain experts but not average users. Our approach can be considered as a first step into this direction as it focuses on leveraging usage data for providing summarizations. Our summarizations are not adapted to each user individually but present a consensus that has been reached by similar behavior in the past.

[7] uses combines hierarchical link analysis with weighted link analysis. For the latter, the authors suggest to combine PageRank with a TF-IDF-related weighting scheme. In this work, usage or feedback data is not considered as an additional source of information.

In the field of recommender systems, [9] propose an approach based on Latent Dirichlet Allocation (LDA) [2] for discovering hidden semantic relationships between items. This includes the extraction of what is considered to be the most important feature of an item (e.g. genre: adventure). The approach is exemplified on a movie and a real estate dataset.

In the field of user modeling, there exist several approaches for leveraging (weighted) semantic knowledge about items [6, 11, 10]. The approach presented in [6] proposes an aggregated presentation of user profiles by extracting and combining the domain knowledge of different items. [11] models users and items each as a feature matrix. For feature weighting in the user profile, an adapted version of TF-IDF is introduced. In the recommendation approach, the authors form neighborhoods of users based on the user-feature matrix. [10] introduces an impact measure that indicates the influences on user behavior by item features modeled as a domain ontology. The approach is presented with examples from the movie domain.

Table 1: 20-nearest neighbors: Beauty and the Beast

Score	Neighbor
0.999	fb:en.aladdin_1992
0.999	fb:en.the_lion_king
0.998	fb:en.the_little_mermaid_1989
0.998	fb:en.home_alone
0.998	fb:en.snow_white_and_the_seven_dwarfs
0.998	fb:en.toy_story
0.998	fb:en.mrs_doubtfire
0.998	fb:en.the_mask_1994
0.998	fb:en.e_t_the_extra_terrestrial
0.998	fb:en.a_bugs_life_1998
0.998	fb:en.babe
0.997	fb:en.willy_wonka_the_chocolate_factory
0.997	fb:en.honey_i_shrunk_the_kids
0.997	fb:en.men_in_black_1997
0.997	fb:en.jumanji_1995
0.997	fb:en.batman_forever
0.997	fb:en.toy_story
0.997	fb:en.the_wizard_of_oz
0.997	fb:en.santa_claus_the_movie
0.997	fb:en.who_framed_roger_rabbit

4. DATASET

For the preparation of first tests, we combined the usage data of the HetRec2011 MovieLens2k dataset [4] with Freebase.[8] The usage dataset extends the original MovieLens10M dataset[9] by additional metadata: directors, actors, countries, and locations have been added to the original dataset. Although this dataset already contains valuable material to perform our tests without making use of LOD (i.e. Freebase), the search space for properties and objects is very restricted. In particular, 26 properties (the four mentioned above plus 22 other properties such as the genre, year, Spanish title, rotten tomatoes[10] rating etc.) are opposed to more than 240 Freebase properties. Also, the range in Freebase is much broader as - for example - more than 380 different genres (`fb:film.film.genre`) are covered in contrast to 20 fixed genres contained in the HetRec2011 MovieLens2k dataset.

The HetRec2011 MovieLens2k dataset includes IMDb[11] identifiers for each movie. This makes the linking to Freebase easy as querying[12] for the IMDb identifier is simple (see listing 2). Given only this query, we were able to match more than 10000 out of 10197 movies.[13]

For performance reasons, we crawled the RDF-XML[14] representation from Freebase[15] and stored it to a local triple store. Using the usage data, we computed the 20-nearest neighbors for each movie and stored the results also in the

[7] See also PageRank [3].

[8] `http://freebase.com`

[9] `http://www.grouplens.org`

[10] `http://www.rottentomatoes.com/`

[11] `http://www.imdb.com/`

[12] Freebase uses a special query language called Metaweb Query Language (MQL).

[13] Unmatched items are mostly TV series that do not match the pattern `"type"="film/film/"`.

[14] `http://www.w3.org/TR/REC-rdf-syntax/`

[15] `http://rdf.freebase.com/`

Listing 2: MQL query: retrieving the Freebase identifiers given an IMDb identifier.

```
{
  "id"= null,
  "imdb_id"="ttIMDb_ID",
  "type"= "/film/film"
}
```

triple store; like in the following example:

```
(fb:en.pulp_fiction, knn:20, fb:en.reservoir_dogs)
```

Using SPARQL queries (like in listing 1) we are able to retrieve common properties between single movies and their neighbors. The results of first tests with this setup are discussed in the following section.

5. PRELIMINARY RESULTS

With the created dataset, we were able to identify and rank features that connect an entity to one of their nearest neighbors. We do not plan to conduct a separate evaluation at the level of neighborhood quality but we are currently in the process of performing comparisons on the level of quality of summarizations. In this analysis, we are also conducting different similarity measures as well as estimating the optimal size of the neighborhood. At the current stage of our work, statistics for the presentation of these results have not been produced.

We will discuss our findings regarding the neighborhood formation in section 5.1. Moreover, preliminary results of the entity summarization approach are presented in section 5.2.

5.1 Neighborhood formation

One of the most important steps is the neighborhood formation dependent solely on usage data. An example for such a neighborhood is presented in table 1. In general the presented neighborhood of the movie "Beauty and the Beast" fits the perception of most observers and also overlaps with related movies presented in IMDb.[16] The scores presented in table 1 are all very close to each other and every score is also close to a perfect match (1.0). In this respect, the question arises whether the k-nearest neighbor approach makes sense with such dense scores. An alternative could be to introduce a threshold rather than just selecting a fixed amount of neighbors (e.g. all movies that have a similarity higher than 0.95). As a matter of fact, the runtime of the SPARQL queries would turn into a gambling game as it can not be decided in advance whether there are 10 or 500 neighbors that cross the threshold. Another approach to address this question would be to introduce different or additional similarity measures that improve the result set while - at the same time - widens the range of the scores. Finally, the optimal neighborhood size is still due for evaluation. As such, the current size of 20 was selected to serve for the creation of first results.

A particularity about the neighborhood is that one movie (`fb:en.toy_story`) occurs twice in the list. This is due to the HetRec2011 MovieLens2k dataset that contains several duplicates with different identifiers. We suppose that these duplicates occur due to an automatic processing that has been conducted in the course of enriching the original MovieLens10M dataset with additional data.

5.2 Neighborhood-based entity summarization

After the neighborhood formation step we are able to extract the 10 most important features for each entity. Tables 2 to 5 each provide an example for a movie entity summarization.

In general, most of the presented examples have genre as one of the strongest components. In this realm, one of the real advantages of LOD can be exemplified, i.e. data richness: genres such as "costume drama", "crime fiction" or "parody" are missing in the HetRec2011 MovieLens2k dataset and can not be circumscribed. It is interesting to see that the property `fb:film.film.written_by` affects all of the presented movies. In the results, the movie "Bridget Jones's Diary" shares with its neighbors that the scene plays in the United Kingdom while Walt Disney as the production company is surely important for the movie "Beauty and the Beast". It is also worth to mention that, according to our results, "Pulp Fiction" is under heavy influence by its director Quentin Tarantino.

The mindful reader will surely notice that not a single actor influences the presented movies. At least "The Naked Gun - From the Files of Police Squad" should have as an important feature the main actor Leslie Nielsen. This is due to the fact that - in Freebase - the actors are hidden behind another node that connects movies, actors, and characters. Queries that deal with such "two-hops-each" relationships (see listing 3) are hard to resolve for triple stores and yet, we were not able to produce a result set from the triple store.[17] However, for the near future we consider ways to circumvent this issue that does not only affect the actor feature and also help to improve the "hop-radius" of such queries.

Another issue that is visible in the results is the problem of data quality and the constant evolution of the data. Newly added property-value pairs like

```
(fb:user.robert.(...).ew_rating, 92)
```

are shared with one or two neighbors but - at this stage - have not been assigned to a sufficient amount of entities to be downgraded with the weighting method introduced in section 2.

6. CONCLUSION

In the following we will summarize the key findings of this early stage of research.

We have presented an approach that tries to leverage usage data in order to summarize movie entities in the LOD space. This part of Semantic Web research is connected to a variety of fields, including semantic user modeling, user interfaces, and information ranking.

The goal of our research is to provide meaningful summarizations of entities. This is the task of identifying features that "not just represent the main themes of the original data, but rather, can best identify the underlying entity" [5]. Our

[16] `http://www.imdb.com/title/tt0101414/`, as of February 2012

[17] We currently employ Sesame with the Native Java Store (see also `http://www.openrdf.org/`)

Table 2: Top-10 features: Beauty and the Beast

Score	Property	Value
39.56	fb:film.film.genre	fb:en.fantasy
29.40	fb:film.film.rating	fb:en.g_usa
19.23	fb:film.film.production_companies	fb:en.the_walt_disney_company
16.89	fb:film.film.music	fb:en.howard_ashman
13.31	fb:film.film.music	fb:en.alan_menken
12.86	fb:film.film.subjects	fb:en.fairy_tale
9.14	fb:film.film.film_casting_director	fb:en.albert_tavares
8.04	fb:film.film.written_by	fb:en.linda_woolverton
7.75	fb:film.film.produced_by	fb:en.don_hahn
7.30	fb:film.film.genre	fb:en.costume_drama

Table 3: Top-10 features: The Naked Gun - From the Files of Police Squad!

Score	Property	Value
27.77	fb:film.film.written_by	fb:en.jim_abrahams
26.00	fb:film.film.written_by	fb:en.pat_proft
22.59	fb:film.film.written_by	fb:en.jerry_zucker
22.04	fb:film.film.written_by	fb:en.david_zucker
18.92	fb:film.film.music	fb:en.ira_newborn
18.44	fb:media_common.netflix_title.netflix_genres	fb:en.comedy
16.89	fb:film.film.film_series	fb:m.0dl08h
16.38	fb:film.film.featured_film_locations	fb:en.los_angeles
16.12	fb:film.film.genre	fb:m.02kdv5l
15.97	fb:film.film.genre	fb:en.parody

Listing 3: SPARQL query: retrieving property-value pairs shared with at least one of the 20-nearest neighbors.

```
select ?p ?q ?t where {
fb:movie.uri ?p ?o.
fb:movie.uri knn:20 ?s.
?o ?q ?t.
?s ?p ?r.
?r ?q ?t.
?s rdf:type fb:film.film.
FILTER((?s != fb:movie.uri) && (?p != knn:20))
}
```

approach can be considered as a further step to this direction. Properties such as rdf:label or fb:type.object.name are currently missing as they are usually not shared with any other entity. With regard to this issue, the approach can easily be combined with another feature ranking strategy. The question whether strong weights for features that are shared with a usage-data-based neighborhood enhance the state of the art is subject to an extensive evaluation that is currently in progress of being conducted.

Additionally, we want to discuss the fact that the presented approach is restricted to a single domain and whether it can work for multiple domains or even cross-domain. Consider a electronics web shop that includes semantic meta-information about the items to be sold. Users that search a for product that fulfills their requirements (whatever those are) provide usage data that can be used to compare two products on the basis of whether they have been browsed by a same set of users (each user has watched a set of items within a given time-frame). Utilizing this information with the proposed approach can lead to a ranked list of features that a product has (e.g. 12 mega pixels in the case of digital cameras). This may help to provide meaningful product summarizations rather than listing all features that it has. However, for data hubs like DBpedia and Freebase, filtering mechanisms (like restricting to rdf:type film) have to be applied for not to compare apples with pears.

7. FUTURE WORK

Considering the simplicity of our current approach and the subjective quality that has already been reached, we plan to follow this track of research. In our next contributions we plan the following enhancements:

- An extensive evaluation of the approach will be conducted: the analysis is will include an intrinsic as well as an extrinsic evaluation with user surveys.

- Features that are specific to an entity (and not shared with others) will be considered in future versions of this approach. It has to be evaluated whether usage data can help with this task.

- The problem of intermediate nodes needs to be addressed in order to provide a scalable solution. This could be done with a fixed set of important property-value pairs (like actors and characters). Another solution would be to set up triple store indexes.

- The ideas of diversifying the results as well as a possible adaption to user profiles and context state interesting challenges.

Table 4: Top-10 features: Bridget Jones's Diary

Score	Property	Value
29.67	`fb:film.film.genre`	`fb:en.romantic_comedy`
29.39	`fb:film.film.written_by`	`fb:en.richard_curtis`
19.40	`fb:film.film.country`	`fb:en.united_kingdom`
18.43	`fb:film.film.film_casting_director`	`fb:en.michelle_guish`
16.75	`fb:film.film.produced_by`	`fb:en.eric_fellner`
16.50	`fb:film.film.produced_by`	`fb:en.tim_bevan`
13.05	`fb:user.robert.default_domain.rated_film.ew_rating`	69
12.79	`fb:film.film.film_format`	`fb:en.super_35_mm_film`
12.51	`fb:film.film.production_companies`	`fb:en.universal_studios`
9.140	`fb:film.film.story_by`	`fb:en.helen_fielding`

Table 5: Top-10 features: Pulp Fiction

Score	Property	Value
21.58	`fb:film.film.directed_by`	`fb:en.quentin_tarantino`
19.75	`fb:film.film.genre`	`fb:en.crime_fiction`
19.10	`fb:user.robert.default_domain.rated_film.ew_rating`	92
16.94	`fb:film.film.rating`	`fb:en.r_usa`
16.38	`fb:film.film.featured_film_locations`	`fb:en.los_angeles`
14.12	`fb:film.film.written_by`	`fb:en.quentin_tarantino`
13.72	`fb:film.film.film_collections`	`fb:en.afis_100_years_100_movies`
13.48	`fb:film.film.edited_by`	`fb:en.sally_menke`
13.31	`fb:film.film.film_production_design_by`	`fb:en.david_wasco`
12.39	`fb:film.film.produced_by`	`fb:en.lawrence_bender`

- With enhanced versions of the presented approach we want to move forward to the direction of user interfaces and user interaction in the context of Linked Data; also in combination with Social Media such as Twitter and Blogs.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257790.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". In: *IEEE Trans. on Knowl. and Data Eng.* 17 (6 2005), pp. 734–749. ISSN: 1041-4347.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation". In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 993–1022.

[3] Sergey Brin and Lawrence Page. "The anatomy of a large-scale hypertextual Web search engine". In: *Proc. of the 7th intl. conf. on World Wide Web 7.* WWW7. Brisbane, Australia: Elsevier Science Publishers B. V., 1998, pp. 107–117.

[4] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. "2nd Ws. on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)". In: *Proc. of the 5th ACM conf. on Recommender systems.* RecSys 2011. Chicago, IL, USA: ACM, 2011.

[5] Gong Cheng, Thanh Tran, and Yuzhong Qu. "RELIN: relatedness and informativeness-based centrality for entity summarization". In: *Proc. of the 10th intl. conf. on The semantic web - Volume Part I.* ISWC'11. Bonn, Germany: Springer-Verlag, 2011, pp. 114–129.

[6] Honghua Dai and Bamshad Mobasher. "Using ontologies to discover domain-level web usage profiles". In: *2nd Semantic Web Mining Ws. at ECML/PKDD-2002.* 2002.

[7] Renaud Delbru et al. "Hierarchical Link Analysis for Ranking Web Data". In: *The Semantic Web: Research and Applications.* Vol. 6089. Lecture Notes in Computer Science. Springer-Verlag, 2010, pp. 225–239.

[8] Ted Dunning. "Accurate Methods for the Statistics of Surprise and Coincidence". In: *COMPUTATIONAL LINGUISTICS* 19.1 (1993), pp. 61–74.

[9] Xin Jin, Yanzan Zhou, and Bamshad Mobasher. "A maximum entropy web recommendation system: combining collaborative and content features". In: *KDD '05: Proc. of the 11th ACM SIGKDD intl. conf. on KD in data mining.* New York, NY, USA: ACM Press, 2005, pp. 612–617.

[10] Patricia Kearney, Sarabjot Singh An, and Mary Shapcott. "Employing a domain ontology to gain insights into user behaviour". In: *In: Proc. of the 3rd Ws. on Intelligent Techniques for Web Personalization, at IJCAI 2005.* 2005.

[11] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. "Feature-Weighted User Model for Recommender Systems". In: *Proc. of the 11th intl. conf. on User Modeling.* UM '07. Corfu, Greece: Springer-Verlag, 2007, pp. 97–106.