Mastering Data-Intensive Collaboration and Decision Making

FP7 - Information and Communication Technologies

Grant Agreement no: 257184
Collaborative Project
Project start: 1 September 2010, Duration: 36 months

D3.3 – Data Mining in Data-Intensive and Cognitively-Complex Settings: Lessons Learnt from the Dicode Project

Due date of deliverable: 31 August 2013
Actual submission date: 30 August 2013
Responsible Partner: FHG, NEO
Contributing Partners: FHG, NEO

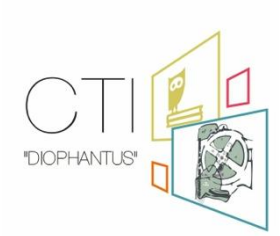
Nature: ☒ Report ☐ Prototype ☐ Demonstrator ☐ Other

Dissemination Level:
☒ PU : Public
☐ PP : Restricted to other programme participants (including the Commission Services)
☐ RE : Restricted to a group specified by the consortium (including the Commission Services)
☐ CO : Confidential, only for members of the consortium (including the Commission Services)

Keyword List: Data mining framework, data mining services, text mining services, visualization, usability, software development, software integration, lessons learned.

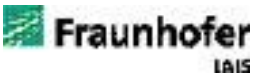
The Dicode project (dicode-project.eu) is funded by the European Commission, Information Society and Media Directorate General, under the FP7 Cooperation programme (ICT/SO 4.3: Intelligent Information Management).

The Dicode Consortium



Computer Technology Institute & Press "Diophantus" (CTI) (coordinator), Greece

University of Leeds (UOL), UK



Fraunhofer-Gesellschaft zur Foerderung der angewandten Forschung e.V. (FHG), Germany

Universidad Politécnica de Madrid (UPM), Spain

Neofonie GmbH (NEO), Germany

Image Analysis Limited (IMA), UK

Biomedical Research Foundation, Academy of Athens (BRF), Greece

Publicis Frankfurt Zweigniederlassung der PWW GmbH (PUB), Germany

Document history			
Version	Date	Status	Modifications made by
1	13-08-2013	First draft	Natalja Friesen, FHG
2	20-08-2013	Second draft	Natalja Friesen, FHG Jörg Kindermann, FHG Doris Maassen, NEO Dhavalkumar Thakker, UOL
3	21-08-2013	Sent to internal reviewers	Manolis Tzagarakis, CTI Fan Yang-Turner, UOL
4	23-08-2013	Reviewers' comments incorporated (version sent to the Dicode SC)	Natalja Friesen, FHG Jörg Kindermann, FHG Doris Maassen, NEO Dhavalkumar Thakker, UOL
5	30-08-2013	Final version (approved by SC, sent to the Project Officer)	Natalja Friesen, FHG

Deliverable manager

- Natalja Friesen, FHG

List of Contributors

- Jörg Kindermann, FHG
- Stefan Rüping, FHG
- Doris Maassen, NEO
- Dhavalkumar Thakker, UOL

List of Evaluators

- Manolis Tzagarakis, CTI
- Fan Yang-Turner, UOL

Summary

This deliverable reports on practical lessons learned while developing the Dicode's data mining services and using them in data-intensive and cognitively-complex settings. The lessons fall into the categories of experiences, concrete recommendation and best practices related to the process of developing the infrastructures for mining data. Various sources were taken into consideration to establish these lessons, including user feedback obtained from evaluation studies, observation of usage and discussion in teams. The lessons are presented in a way that could aid people who engage in various phases of developing similar kind of systems.

Table of Contents

1 Introduction

This deliverable reports on practical lessons learned while developing the Dicode's data mining services and testing them in data-intensive and cognitively-complex settings. The lessons fall into the categories of experiences, concrete recommendation and best practices related to the process of developing the infrastructures and services for mining data. The infrastructure and services have been designed and implemented in the context of WP3, WP4 and WP5 of the Dicode project (development of data mining services in WP4 concerns the recommendation mechanism of Task 4.2; in WP5, it concerns the development of a multi-perspective ontology for collaboration and decision support of Task 5.2). Various sources were taken into consideration to establish these lessons, including user feedback obtained from evaluation studies, discussion in teams, as well as observation of services' usage. The lessons are presented in a way that could aid people who engage in various phases of developing similar kind of systems.

The report consists of two major sections that reflect the different natures of raw data that have been investigated within the project. On the one hand, data are more or less unstructured mostly consisting of text documents; on the other hand, data are semi-structured as provided by Linked Data. For the former, scalability and latency are in the focus under the title of "Data Analytics and Big Data" (Section 2), while the latter come under the title of "Exploratory Search over Linked Data" (Section 3). The separation is justified due to the different nature of the data as well as of the different methods of exploring these data.

One might note that deliverable D3.1.2 ("The Dicode Data Mining Framework (enhanced version)") already includes a short description about the lessons learned concerning Hadoop and other Big Data technologies (in Section 3: "Lessons Learned"). It describes several major trends that have been observed during the run of the project. The most important trend described is a move from pure MapReduce to higher-level frameworks like Apache Pig, Apache Hive[1], Cascading[2] and, recently, Cascalog[3]. This trend is relatively old and was already stated at the beginning of Dicode. What we did not foresee at the beginning was the relatively slow pace of the development of comprehensive libraries for Big Data like, for example, Apache Mahout. Concerning the availability of open source utilities, the situation is the opposite: there is abundance of especially Pig Latin scripts for lots of typical tasks for text extraction and mining.[4]

2 Data Analytics and Big Data

A major concern of dealing with Big Data is the dichotomy between latency and scalability. Analysis of huge amounts of data takes time, which implies high latency, while obtaining analyses in short time is typically feasible only for small data with bad scalability. The

[1] http://hive.apache.org/
[2] http://www.cascading.org
[3] https://github.com/nathanmarz/cascalog
[4] For an example of a quickly developing project, see https://github.com/linkedin/datafu/pulse/monthly

Lambda Architecture[5] has been proposed as a compromise combining scalable and robust analysis of Big Data using Hadoop with real-time analysis, e.g. Storm. These developments have been so recent that they were beyond the scope of Dicode, but have and will have some impact on exploitations of the Dicode results. Dicode has focussed on improving scalability – and to some extend latency – using the Hadoop-based approaches to data-centred parallelisation. Lessons learned with regard to this technology are listed in the respective subsection.

Scalability and latency not only depend on the technology used but also on the services offered. Lessons learned with regard to services are subsumed in the respective subsection.

Finally, the effectiveness of analyses not only depends on technology and services, but also on appropriate and efficient usage of these. A subsection on user involvement provides lessons learned from this perspective.

2.1 Technology

Lesson 1: Innovative Big Data solutions can greatly simplify data processing tasks

In Dicode, we used Wikipedia as a central resource for Named Entity Disambiguation. From Wikipedia's link structure, we derived the probabilities of signifiers for certain entities[6]. Amongst other statistics, we count links from a certain surface form to a Wikipedia article (e.g. how often "George Bush" either links to the father or to the son).

With 4.2 Million articles, the German Wikipedia seems quite small from a Big Data perspective. For our analysis, each article is analysed several times, because global information like N-gram statistics is needed and recursive link processing is required. Before Hadoop was introduced, the Wikipedia analysis component was implemented as a sequential UIMA pipeline[7]. In total, the analysis took several days on a standard machine. In Dicode, the processing time was reduced to several hours. Now a cronjob[8] regularly checks if a new Wikipedia dump has been published and generates the statistics automatically. This lesson shows us that our idea of "size" changed tremendously during the last three years. Before Dicode, processing a collection of several million documents took a couple of days. Today, this is reduced to a couple of hours. In addition, configuration-intensive frameworks like UIMA today are often replaced by more light-weight approaches. Developing Big Data solutions in most cases does not require writing complex MapReduce jobs. With Apache Pig, an easy to use scripting language is available, which allows for rapid development. Dicode's new Wikipedia statistics component is pretty concise: it contains about 100 lines of code.[9]

---

[5] Marz, N. & Warren, J. (2012). Big Data - Principles and best practices of scalable real-time data systems. Manning Publications

[6] See D3.2.2 (Section 3.1.4: "Named entity service") for details about the Wikipedia statistics.

[7] http://uima.apache.org/

[8] http://en.wikipedia.org/wiki/Cron

[9] Wikipedia statistics is computed by the following pig script: https://github.com/dicode-project/pignlproc/blob/master/examples/nerd-stats/nerd-stats.pig

Lesson 2: Conducting a Big Data project without a shared cluster infrastructure slows down development

From the beginning of Dicode, it was clear that it was not feasible to rely on a shared hardware infrastructure designed for Big Data: none of the partners already had an appropriate cluster infrastructure at hand and there was no explicit hardware budget included in the project calculation. Each of the partners therefore set up a separate development infrastructure.

An important question was how to store and access commonly used data sets. Batch processing systems like Hadoop achieve high scalability and throughput by moving the computation to the data: algorithms are executed on a subset of data that is stored locally at the individual nodes and the results of those analyses are later collected and assembled. We first experimented with a combined approach: for development purposes, a small subset of documents was downloaded by the respective partner and subsequently used on their local infrastructure, e.g. for training of machine learning models. Later, the developed component was wrapped into a user defined function (UDF) and executed on the cluster. We used this rather cumbersome approach in the development of an early version of FHG's Named Entity Service, which was based on Conditional Random Fields[10]. FHG had no direct access to NEO's cluster, because a shell access to the development environment does not comply with the company's security standards. Additionally, FHG uses a different technology stack on their Hadoop infrastructure. The integration of the algorithms into a UDF was therefore performed by NEO which lead to longer development cycles.

To avoid this in the future, we would opt for a shared hardware infrastructure, for which an extra budget and investment into a secure set-up is required.

Lesson 3: MapReduce is not always the best choice for Big Data

At the beginning of Dicode, we focused on batch processing. In the final months of the projects, we studied how we can keep the advantages of batch-style distributed systems like Apache Hadoop when dealing with near real-time requirements. Our first reflections on this issue were presented in deliverable D3.1.2 ("The Dicode Data Mining Framework (enhanced version)") in Section 3 ("Lessons Learned"). In this document, we want to recapitulate the discussion and present our findings from the last months of the project.

Besides being an obstacle for efficient software development and debugging, as described in D3.1.2, the latency of batch processing also interferes with the requirement for "freshness", which is at the core of many information processing applications: in social media monitoring, the user demands an instant alert if there is a new report about the brand or event in question; in a news search, each article has to be analysed and annotated with automatically generated meta-data before being indexed for search. In both cases, a delay of several minutes is not acceptable.

[10] As described in deliverable D3.2.2, FHG's Emotion Detection Service uses similar technologies.

In the literature, a combination of batch and stream processing frameworks is suggested, which combines instant stream processing of incoming data and in-depth processing of batches of data for final results[11]. In Dicode, we evaluated techniques for low latency document analysis relatively late in the project. The envisioned architecture combines batch and stream processing. Large numbers of documents are processed in batch style on Apache Hadoop. Additionally, a fast lane processes high priority documents immediately. This architecture adds additional complexity to the already challenging Hadoop-only solution. Recently, we have seen former users of Hadoop switching completely to stream-processing frameworks. Lightweight stream-processing frameworks like Storm[12] now seem to fill a gap in Big Data scenarios and serve as an easier solution for large-scale text mining. In text mining, batch-processing remains important for the re-calculation of statistics, which requires access to the complete data set.

Lesson 4: Running a cluster consumes significant developer resources

Setup and operations of distributed systems like Hadoop requires profound server and network administration skills, which most software developers in the "Java world" do not have. At the beginning of the project, we underestimated the costs for the administration of the development infrastructure and especially for the skills acquisition in this field. At NEO, we started out with a small cluster of three nodes, because the project did not have any budget for the acquirement of a larger cluster. Later on, we integrated the three machines bought for the Dicode project into a larger cluster shared with another project. By doing so, we could test the technologies developed in Dicode with a more realistic setup. As more people were dependent on the cluster, reliable operations became an issue. Concerning software versions and monitoring tools, both FHG and NEO teams had to develop best practices. Scheduling of jobs and access rights had to be implemented according to the shares of the budget of both projects. Dicode contributed the Log Aggregation service (see D3.2.3) for the improvement of operations and debugging. In total, we spent much more developer resources than estimated on the operation of the cluster. As a result, the team members gained valuable knowledge, which improves NEO's competitive position (see deliverable D7.2.2, Section 3.5.3: "New skill profiles").

Lesson 5: Don't underestimate the importance of meaningful data visualization

Text mining is mostly about automatically generating meta-data about documents or document collections. Typically, the resulting annotations are persisted in a data store. In many use cases, text-mining results will not be presented to the end user directly. Typically the generated meta-data will be used as input for other processing steps. Named Entities, for example, might be indexed by a search engine. The search engine user can then query for Named Entities in a field search.

In other use cases, we want to present the results of our analysis to the end user directly. Most people agree that a good visualisation tells much more than a spread sheet. But we

[11] Marz, N. & Warren, J. (2012). Big Data - Principles and best practices of scalable realtime data systems. Manning Publications.
[12] http://storm-project.net/

have to find a good way of presenting the data. Visualization of Big Data can be challenging. Even if the amount of data to be visualized is big, the user should be able to get a general idea about the data on first sight. It should also be easy to switch to a more detailed view, e.g. by zooming into the graph.

Data visualization was not our main focus in Dicode. For us, visualization was mainly a way to present the project results to the partners and to customers of NEO. Our text mining technologies were developed mainly for the integration into back-end components. As all partners agreed on developing the Dicode Workbench as a collaboration support platform, we integrated our widgets into the Workbench. The integration was easy due to the lightweight approach that allowed for the integration of iFrames. In the first year of the project, NEO produced a couple of widgets for the visualization of the Twitter analysis. We used three different chart types: a pie diagram, a zoomable map and a tag cloud. A pie diagram is a typical diagram of the "small data" age. Maps are becoming more and more popular due to the increasing availability of geodata. Tag clouds can be seen as a typical Web 2.0 visualisation – initially they were used to display the ratios of tags assigned by users.

During the project we realized that Big Data visualization had become a very interesting field. Big Data seems to require new types of visualization. People in different disciplines have been experimenting with various new types of diagrams. Today, a tag cloud seems pretty old-fashioned. New types of diagrams emerge: chord diagrams that originally were used in bioinformatics are now used to visualize text mining results[13].

Lesson 6: Twitter's research stream is not suitable for Social Media Monitoring

Dicode chose Twitter as a major data provider for Use Case 3. Many research projects started out analyzing Tweets a couple of years ago. Tweets seemed to be the ideal candidates for Social Media Monitoring. The large amount of accessible Tweets was tempting. In addition, it was obvious that a marketing analyst would have to monitor Tweets because of the near real-time spreading of information, which seemed to be a prototypic case of viral marketing. Twitter also seemed like a good candidate for social network analysis. Twitter's openness towards the developer communities also made the use of Tweets quite attractive.

Today, the situation has changed. Twitter has restricted the API access severely in several ways[14]. Twitter's business model today is mostly based on paid advertisement (so-called "promoted tweets, accounts and trends"[15]). In addition, Twitter monetizes the data itself – either directly or via companies like Gnip and Datasift.

Our experience in Dicode was that the quality of Twitter's research stream – supposedly 1% of all global Tweets – is rather low. The daily tag cloud extracted from all Tweets in the research stream mainly shows star signs, computer games or teenager related topics. Pavlo

[13] See for example https://github.com/norvigaward/naward25/wiki/Babel-2012---Web-Language-Connections

[14] http://mashable.com/2012/08/16/twitter-api-big-changes/

[15] https://business.twitter.com/marketing-twitter
http://advertising.twitter.com/
http://techcrunch.com/2012/01/22/dld-2012-jack-dorsey-twitter-has-a-business-model-that-works/

Baron even suspects Twitter of deliberately adding "garbage" to the 1% research feed[16]. Due to the small sample, applying social network analysis algorithms to the 1% feed does not make too much sense - for such an analysis, large interconnected amounts of Tweets of a social network of each user would have to be available. A focus on Tweets in German leads to an even sparser dataset.

Various publications deal with the difficulties in mining Tweets. Tweets are short and tend to be idiomatic. This turned out to be a problem for the text mining algorithms implemented in Dicode.

Named Entity Recognition and Disambiguation: Our NERDist algorithm disambiguates the spotted entity candidates based on the context in which the candidate occurs. In Tweets, there is simply not enough context; our experiments showed that only a small share of Tweets contained more than one entity candidate. In addition, the text in Tweets is not well-formed regarding standard language, which makes it hard to re-use models trained on other text data.

Emotion Detection: The applied Conditional Random Field algorithm extracts positive or negative phrases from documents. Typically, those phrases are as long as a Tweet.

In both cases above, we decided not to adopt the algorithm and/or training to short idiomatic texts. Our reasons are the following: most text mining use cases of NEO and FHG deal with medium sized documents like news. Since both partners want to be able to use the algorithms developed in commercial projects, we decided in replacing Twitter by other text corpora like news and blogs – at least for higher level analyses like NERDist and Emotion Detection. Even if Twitter analysis will be required in a project, we could rely on third-party offers: Twitter data resellers like DataSift already provide sentiment analysis and named entity recognition[17].

2.2 Services

Lesson 7: Development should be based on use cases

Since the beginning of the project, we worked closely with the use case partners in order to identify their needs and then convert them into design specifications. Three use cases were defined in the context of the Dicode project, namely:

- **Use Case 1:** Clinico-Genomics Research Assimilator;
- **Use Case 2:** Trial of Clinical Treatment Effects;
- **Use Case 3:** Opinion Mining from Unstructured Web 2.0 Data.

This selection of use cases was intended to cover the full range of the features and functionalities of the project, while representing specific domain problems and dealing with various types of large scale and real-time data from heterogeneous sources.

[16] Baron, Pavlo: Big Data für IT-Entscheider. Riesige Datenmengen und moderne Technologien gewinnbringend nutzen. München 2013

[17] http://dev.datasift.com/blog/salience5

The use case based development of the Dicode services occurred in several steps, each of them was performed in collaboration and interaction with representatives of the use cases. First, the current work practices were reviewed and discussed. The technical partners elaborated the service ideas addressing the problems identified by users. This step enabled us to fully understand the user requirements and discover common characteristics of the use cases regarding the users, the activities and the data. In the second step, prototypes of the single services were created and showed to the use case partners in order to obtain their feedback and to improve the services. The early user feedback helped developers to manage the consequences of design change. In the third step of the service development, several actions were performed concerning the improvement of the services' usability, acceptability and overall quality.

Lesson 8: Efforts put into data conversion tasks should not be underestimated

Typical data mining software requires an availability of structured data that provide information in a meaningful and descriptive way. The simplest example of the structured data is a table, where the data is stored in columns; one column for each specific attribute and the data is also stored in the row. However, some applications, particularly in specific fields such as Rheumatoid Arthritis Treatment (Dicode's Use Case 2) prescribe their own requirements to the data format, namely DICOM[18] files. DICOM (Digital Imaging and Communications in Medicine) is a standard for handling, storing, printing, and transmitting information in medical imaging. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format.

The original approach for supporting the decision making in the field of rheumatoid arthritis treatment was to apply data mining techniques to doctor's reports about treated patients. Such report is an outcome of the specific software broadly used for analysis of patient data in medical research. This software requires the DICOM files as input format. The analysis of patient data is a very complex process consisting of many steps. In each step, the user is required to interact with the system: to give his feedback concerning classification of patient numbers, to mark region of interest in the image and so on. This analysis process assumes domain knowledge and cannot be reproduced automatically. Therefore, producing a meaningful data set needed for data mining is associated with high manual processing costs. They are a bottleneck regarding a user interaction, which was underestimated at the beginning of the project.

Although a large pool of DICOM files was available, the conversion of these files into structured data was not possible because of lack of user sources. The costs for conversion of original data into appropriate formats required by standard tools should not be underestimated.

Lesson 9: Real-world application needs analysis of unstructured data

The typical Big Data tools assume the availability of structured data, while in many real word applications only unstructured data are available (e.g. texts). This has been the case

[18] http://en.wikipedia.org/wiki/DICOM

with the brand watch and marketing applications of Use Case 3. On the other hand, text-mining algorithms often need to be trained on a set of texts before the "model" can be applied to new texts. Training runs of the algorithms used in the project were notoriously slow with Hadoop when the project started (see deliverable D3.1.1). We therefore decided to integrate sequential training routines in the workbench that operate on text collections of moderate size (see Section 3 in deliverable D3.1.2).

However, application of trained models is fast, and it was included in the CeBit 2013 demonstrator that extended FHG's project technology (see deliverable D7.2.2).

Lesson 10: Knowledge extraction yields results which are often hard to interpret

The goal of the Dicode Subgroup Discovery service is to generate a human understandable representation of the most interesting dependencies in the data in order to support decision-making in the Dicode Workbench. Therefore, the understanding and interpretation of the results are very important issues for the usage of this service. While the user is typically interested in a small yet meaningful output, the outcome of the existing techniques is a huge number of redundant patterns. Having shown the results of the Subgroup Discovery service to researchers, we realised that instead of being supported in knowledge extraction, the user is overwhelmed by the amount of information. Pattern interpretation is a time consuming task, since human experts must manually review the patterns.

In order to reduce the number of raw patterns to a subset of manageable size, we investigated two approaches: one is based on using statistical characteristics; the second includes user feedback in knowledge discovery process. The first approach (described in Grosskreuz at al.[19]) uses the statistical quality of pattern to output the *k* top-quality patterns. This modification leads to a considerable reduction in the amount of returned patterns without losing statistical descriptiveness and, as a consequence, a better understanding of the discovered dependencies. The second approach enables a user to influence the output by including/excluding certain attributes from the search. After reviewing the results, the user can set up a new iteration of the service run by specifying undesired (e.g. biological or medical) attributes. The combination of both approaches enables us to retrieve patterns that have a statistically better quality and, at the same time, are more relevant regarding to the user preferences.

Lesson 11: Use open source software whenever feasible

Some of the Dicode services are based on existing open source software such as R[20] and RapidMiner[21]. For instance, the Subgroup Discovery service uses R to build a connection to Gene Ontology (GO) and to enrich a set of gene names by their functional interpretation, which is given by GO terms.

---

[19] Fast and memory-efficient discovery of the top-k relevant subgroups in a reduced candidate space, Grosskreutz, Henrik; Paurat Daniel. Machine learning and knowledge discovery in databases. European conference, ECML PKDD 2011. Pt.1 : Athens, Greece, September 5-9, 2011; proceedings

[20] http://cran.r-project.org/

[21] http://rapid-i.com/

At the beginning of the Dicode project, we analysed a variety of data mining tools in order to select an appropriate platform for the Dicode data mining services. The outcome of the tool comparison was that R and RapidMiner are freely available open source frameworks that deliver reliable solution for Dicode issues. RapidMiner is the most popular one (even more popular than any commercial product) according to a poll at KDNuggets.com[22] - a well known and broadly trusted website amongst data miners. R satisfies the most requirements prescribed by the field of biomedical research (Use Case 1). The experts analysing genomic data have built a wide range of custom libraries for R. Bioconductor[23] uses the R statistical programming language and is one of the most popular open source and open development software for the analysis and comprehension of high-throughput genomic data. No one of the existing commercial frameworks offers such flexibility, as it is available with R.

Lesson 12: Parallelisation of data mining algorithms may be difficult and in many situations even unfeasible

Some data mining algorithms, e.g. the Subgroup Discovery algorithm, which Dicode Subgroup Discovery service is based on, cannot be efficiently parallelized using standard techniques and algorithms. The main challenge associated with parallelization is to break a data mining problem into independent trivial tasks. This requirement cannot be satisfied for the subgroup discovery algorithm. One reliable solution for such kind of problems is in-memory processing. This approach enables an efficient parallelization on the thread level. For better performance, the implemented Subgroup Discovery algorithm exploits the complex in-memory database based on the special data structure called FP-Tree[24]. An FP-tree is a compact data structure that represents the data set in tree form. Such data representation enables one to reduce both running time and memory size requirements of an algorithm.

2.3 User Involvement

Lesson 13: User involvement / interaction is a bottleneck

Availability of labelled data is an important assumption for many data mining tasks associated with supervised or semi-supervised learning. In many practical applications, unlabeled instances are abundantly available, while obtaining labeled data is a very costly step, in particular when instance pairs have to be manually labeled by a user. However, incorporation of user feedback in different stages of data mining process enables one to improve significantly the quality of results. For instance, obtaining of relevance feedback is a common practice in information retrieval. The idea of relevance feedback is to involve the user in the retrieval process so as to improve the final result set. In particular, the user gives feedback on the relevance of documents in an initial set of results[25].

---

[22] http://www.kdnuggets.com/polls/2010/data-mining-analytics-tools.html
[23] http://www.bioconductor.org/
[24] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00). ACM, New York, NY, USA, 1-12. http://doi.acm.org/10.1145/342009.335372
[25] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. Information Retrieval: Implementing and Evaluating Search Engines. MIT Press, Cambridge, Mass., 2010

The Dicode services which intend to support decision making, such as Similarity Learning service, requires a certain amount of user interaction in order to deliver good results. Our research concerning the problem of how to perform distance metric learning accurately at minimal labeling costs is a reliable solution to avoid the 'user interactions' bottleneck. In Dicode, we proposed a sampling method for selecting a fixed number of 'interesting' instance pairs to label that enabled us to learn a good distance metric[26].

Lesson 14: The organization of workshops greatly helps in collecting user feedback

The descriptive nature of local patterns makes them useful as a source of information for decision making. Therefore, the understanding and interestingness of the patterns that are retrieved by the Dicode services are the key requirements for their successful usage. Workshops provide a good opportunity to bring together researchers and practitioners from biology, medicine and bioinformatics domains in order to identify gaps between research and practice and to clarify the user needs in an interactive way. Key questions investigated in the related workshop organized in the context of Dicode (namely, "The Interpretable Pattern Workshop", see deliverable D7.2.2 – Section 2.4.1) were: an appropriate pattern language and inclusion of user feedback in order to improve the interpretation of patterns. Additionally, in the context of the workshop, we organized a challenge where we show the patterns discovered by several algorithms to real experts from BRF (Greece). We used a questionnaire to obtain the feedback. A detailed analysis of the expert's feedback enabled us to clarify the user needs regarding the interpretation, novelty and interestingness of the discovered patterns. The outcome of the analysis can be summarized as follows:

- The way the results are presented was very important to the user. Very large pattern descriptions are hard to understand.
- Very general patterns are not likely to be useful. Generality of the discovered patterns is not always a characteristic of the data, but include domain knowledge, so the possibility to interactively include the user feedback into discovering process, e.g. to remove very general attributes, is a very helpful function. Moreover, the existing mining algorithms have to be optimized to discover more specific patterns
- Expert knowledge plays an important role - even the best mining methods have to be optimized including the expert feedback.

The expert's feedback provided a significant contribution to the quality and usability of the Dicode services for decision support, such as the Subgroup Discovery Service.

Lesson 15: Visualisation is important

Topic models based on the LDA (Latent Dirichlet Allocation)[27] algorithm have been around for several years, but only their visualization in a graph structure was able to bridge the gap

[26] Natalja Friesen and Stefan Rüping. Distance Metric Learning for Recommender Systems in Complex Domains Mastering Data-Intensive Collaboration through the Synergy of Human and Machine Reasoning (dicoSyn 2012). A workshop at CSCW 2012, February 12, 2012, Seattle, WA.
[27] Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022.

between the data-mining expert and the user. In Use Case 3, the brand watch application greatly benefitted from the visual text collection overview provided in a topic graph.

3 Exploratory Search over Linked Data

3.1 Introduction

There are growing arguments that Linked Data technologies can be utilised to enable user-oriented exploratory search systems for the future Internet. Recently, search over Linked Data has been studied in different domains and contexts. Although the technological platforms for exploring Linked Data are growing, enabling citizen users to explore the interconnectable links to make sense of structured data is still a key challenge[28]. There is still limited insight into how conventional semantic browsers over Linked Data can be extended to empower exploratory search, which is open-ended, multi-faceted and iterative in nature. Empirical user studies in representative domains can identify problems and elicit requirements for innovative functionality to assist user exploration.

Anticipating these new tools for knowledge workers to exploit Linked Data in the future, emerging research begin to identify major issues with user exploration of Linked Data, derive requirements for new methods, and engineer solutions to implement these methods utilising semantic technologies and tools. As part of the Dicode project, an exploratory search system, called "Pinta", has been developed for browsing over linked semantic data. Further details on the architecture of Pinta are given in deliverable D5.3.2 (see Section 3: "Semantic Support for Exploratory Search"). Pinta is piloted in experimental studies[29] in a music domain (instantiation called "MusicPinta") with users to elicit requirements and learn lessons about intelligent assistance. These lessons are derived based on observations of challenges users faced while interacting with MusicPinta. The following section provides a brief description of the main elements of these user studies.

---

[28] M. C. Schraefel, Keynote at the ISWC 2010, "What does It Look Like, Really? Imagining how Citizens might Effectively, Usefully and Easily Find, Explore, Query and Re-present Open/Linked Data," in Proceedings of the 9th international semantic web conference on The semantic web, 2010, vol. 6497, pp. 356–369.

[29] Reported in:

- Thakker, D.A, Dimitrova V., Lau, L.,Yang-Turner, F. Despotakis, D. (2013) Assisting User Browsing over Linked Data: Requirements Elicitation with a User Study. To appear in the proceedings of the International Conference on Web Enginnering (ICWE 2013), Aalborg, Denmark, July 8-12.
- Yang-Turner, F., Lau, L., Dimitrova, V., Thakker, D. (2013). Profiling Exploratory Browsing Behaviour with a Semantic Data Browser. Proceedings of the Nineteenth Americas Conference on Information Systems (AMCIS 2013), Chicago, Illinois, August 15-17, 2013.
- Dimitrova V., Lau, L., Thakker, D.A, Yang-Turner, F. Despotakis, D. (2013) Exploring Exploratory Search: A User Study with Linked Semantic Data. To appear in the proceedings of the ACM workshop on Intelligent Exploration of Semantic Data(IESD 2013). In conjunction with the ACM conference on Hypertext 2013, Paris, France, May 1-3.

3.2 User Study to learn lessons about exploratory search over linked data

The study involved 12 participants recruited on voluntary basis. All participants had IT background and good experience in web search. Each participant attended an individual session, conducted and observed by an experimenter for an hour:

(i) using a pre-study questionnaire [5 min] for collecting information about the user and test his/her domain awareness;
(ii) introducing MusicPinta [10 min];
(iii) conducting Task 1 [15 min] aiming at identifying distinctive characteristics of the musical instrument "bouzouki";
(iv) conducting Task 2 [15 min] for identifying usage and features of the musical instrument "electric guitar";
(v) a post-study questionnaire [10 min] for testing again the participant's domain awareness and gathering usability feedback; and,
(vi) briefly interviewing [5 min] for eliciting the overall impression of using MusicPinta for exploratory search. After each task, the users were asked to fill-out a short questionnaire to assess cognitive load using the NASA-TLX questionnaire[30].

The study required participants to complete two tasks (see Table 1) related to exploring musical instruments and was positioned within an advertising scenario for a fictitious UK music shop (see Table 1). In both tasks, the participants were given an entry point for browsing and asked to fill in their answers in a provided template. The tasks exhibit the characteristics of exploratory search tasks[31]: the main goal is learning and/or investigation of a musical instrument; there is a low level of specificity about the information needed and how to find it; search is open ended, requires finding several items and involves a degree of uncertainty; tasks are 'not too easy' and include multiple facets.

Two musical instrument experts (one for Bouzouki, one for Electric guitar) have marked the outcome of participants for the two tasks. The marking is to measure how successful the participants have been in completing the tasks using MusicPinta. Participant achieved 70% average score for Task 1 and 48% for Task 2. In the following section, we focus on the lessons learned related to the task outcome and browsing behaviour that allow us to elicit requirements for supporting exploratory environments.

[30] Hart, S.G. and Staveland, L.E.: Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Human Mental Workload, 11, pp. 139–183, 1988.
[31] Summarised in: B. M. Wildemuth and L. Freund, "Assigning search tasks designed to elicit exploratory search behaviors," in Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval - HCIR '12, 2012, pp. 1–10.

Table 1. User tasks in the experimental study.

Task 1: Characteristics of a musical instrument [bouzouki]	**Task 2: Usage and features of a musical instrument [electrical guitar]**
The music shop is extending its collection of instruments with international musical instruments. You work in an advertising agency which has been asked to prepare an advertisement script for some of the new instruments that will appear in the shop. A key part of the preparation of the advertisement script is the research of the product. You have been asked to conduct a research of one of the new instruments, called bouzouki, using the information available in MusicPinta. You have to identify: • the main characteristics of bouzouki; • up to five similar instruments to bouzouki; • features that make bouzouki distinctive from the similar ones you have chosen. Go to 'Semantic Search' in MusicPinta and type bouzouki. Browse the content and follow links. Complete the provided form.	The music shop wants to increase the sales of its traditional musical instruments, such as electrical guitars. It intends to do this by adding links to creative commons album recordings with electric guitars, together with some interesting information about these albums to inspire customers to play/buy electric guitars or other musical instruments. Furthermore, when displaying its electric guitar items, the shop wants to highlight key features people look for when purchasing electric guitars. You are asked is to conduct the research to address the above requirements by using information provided in MusicPinta. You have to review the information about electric guitar and identify: • three interesting album recordings that include electric guitars and specify what is interesting; • key features that people look for when purchasing an electric guitar. Go to 'Semantic Search' in MusicPinta and type electric guitar. Browse the content and follow links. Complete the provided form.

3.3 Lessons Learned and Requirements for Assisting User Browsing Over Linked Data

The following observations are based on study considering the user interaction while browsing the linked semantic data. Each observation was assessed to elicit requirements for supporting exploratory environments.

Observation 1: Abstraction conundrum. While browsing specific performances and performers, two participants clicked on abstract concepts, such as instrument, performance and performer, from the Music Ontology. In both cases, the participants were looking for concrete information (e.g. participant-12 clicked on instrument in Task 1 when seeking for more detail about a musical instrument, while participant-05 clicked on performer and performance in Task 2 when seeking more detail about an album). The aggregated datasets in MusicPinta have large number of instances for the abstract concepts (which is typical of linked datasets), which led to confusion as the result was a long list of performers, performance and instruments, and the participants quickly pressed the back button on their browsers.

Requirement 1: Offer semantic links at an appropriate level of abstraction. The above observation motivates consideration on identifying what can be algorithmically offered as the right level of abstraction on various browsing junctures. This is especially important when the abstract concepts have large amount of concrete instantiations. The main challenge here

is what to suppress and what to display to the user; e.g. how to decide which performances out of 71k to display when a user is on the entity page of the abstract concept performance.

Observation 2: Exploring entities/content with insufficient information. Another interesting case is the high number of 'empty clicks' - the user clicks on a link and is taken to a page with no information, sees that this link is not helpful and quickly returns to the previous page. In Task 1, such clicks concerned similar instruments, e.g. there was no information about bajitar, xalam, rebab. In Task 2, such clicks concerned performances (music albums) and happened quite often. 'Empty clicks' leading to pages with no information was seen as one of the main reasons for user's frustration. At the same time, may be due to their experience of links that lead to dead-ends, some links were perceived as empty without exploring them further and the users missed to click on important for the tasks information (e.g. pages about musical instruments were abandoned, although there was useful information about relevant instruments; or interesting facts about an album artist were overlooked as the users did not click on the corresponding link). With linked datasets, it is typical to find entities that do not have much explanation or links to other entities.

Similar issues were observed with content (Amazon Reviews in our user study). Textual content in semantic data browsers are semantically tagged and made available via one of the facets. Users clicked to view some of the Amazon reviews to find out more information about an instrument and its review. However, some of the reviews were deemed to have insufficient information to be useful. This observation is in line with relevant research conducted which concludes that not all reviews are equally helpful[32]. One can extrapolate such observations to be generic enough to be applied to social content and conclude that social content has a variety of usefulness levels, while being possible to find content that has insufficient information to be of help in browsing.

Requirement 2: Reduce entity link options. Avoid showing entity links that do not lead to any new information. Reduce number of entity links shown to the user based on their browsing value; allowing reduction of clutter and confusion. The challenge here is to define what 'browsing value' is and how to calculate it for an entity with respect to other entities from the same entity page.

Requirement 3: Reduce content link options. Avoid showing content links that do not lead to any new information. Reduce number of content links shown to the user based on their helpfulness/usefulness. The challenge here is to decide the parameters of helpfulness/usefulness of content.

Observation 3. Varied selection strategies while facing too many choices. Both tasks (deliberately) put the users in situations where they had too many choices. This means that the users had a large number of links to review while in a focus entity page. For example, the bouzouki page included 12 different links in the 'facts facet' (11 links to concepts in the middle classification level and 1 link to the abstract concept instrument) and 51 links in the 'terms facet' (43 links to musical instruments and 8 links to performances). The entry point in Task 2, the electric guitar page, included 18 links in the 'facts facet' (to concepts in the middle

[32] For example, useful reviews have considerable review depth compared to non-useful reviews have been identified in: S. M. Mudambi and D. Schuff, "What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com," MIS Quarterly, vol. 34, no. 1, pp. 185–200, 2010.

and upper classification levels), 78 links to albums in the 'terms facet', and 8 links to Amazon reviews in the 'content facet'. This is a typical situation with the datasets from Linked Data. For example, for the DBpedia dataset, which has 3.5M entities and 627M triples, on average, a user might have to review 192 links while exploring a focus entity.

We observed users following different strategies when presented with too many choices in the browsing interface:

(i) clicking on the nearest classification link from the 'facts facet' (e.g. plucked string instruments or string instruments) to see general characteristics in the case of bouzouki as part of Task 1. However, users rarely clicked on links from the 'facts facet' as part of Task 2, as the task did not require this;
(ii) clicking on instruments mentioned in the 'related terms facet'– (e.g. lute and mandolin mainly in Task 2;
(iii) clicking on something (e.g. an instrument) that sounds familiar (e.g. sitar, banjo, pipa in Task 1);
(iv) click on something (e.g. an instrument or an album) that sounds interesting or unusual (e.g. oud, xalam in Task 1 and noticing a woman artist or something interesting in the album name in Task 2);
(v) clicking on something that looks important (e.g. an artist has several albums in Task 2); and,
(vi) clicking randomly (after exhausting other strategies).

This observation is in line with the latest research in search engines and HCI; increasing numbers of options can make designers and users feel less confident when deciding and less happy with the results[33]. To support the varied level of selection strategies, two requirements were derived.

Requirement 4. Make facet selection process dynamic and intuitive. The use of different types of facets is useful but dependent on tasks. The challenge here is to give control to users and help them to decide or make it easier for them to decide the facet and when they would like to use it. For example, providing guideline on the utility of different types of facets in the system and allowing facets to be added or removed as required by the user.

Requirement 5. Take into account context to cater for interests and importance. People when faced with many choices do select what they find useful/familiar/interesting/unusual/important. Hence, there is a merit in making it easier for users to decide/spot easily these values. The challenge here is how to measure and decide these values from the available options for a specific user or holistically.

Observation 4. Text and Media information influence user experience and performance. For Task 1, a great deal of performance owed to the use of textual description and images while identifying characteristics of an instrument. Hence, there is value in offering unstruc-

[33] See: A. Oulasvirta, J. Hukkinen, and B. Schwartz, "When More Is Less: The Paradox of Choice in Search Engine Use," In Proc. of SIGIR'09, July 19–23, 2009, Boston, MA, pp. 516-523; Schwartz Barry, The paradox of choice : why more is less / Barry Schwartz, 1st ed. ECCO, New York, 2004.

tured (textual and multimedia information) in conjunction with structured data (semantics) for exploration.

Requirement 6. Offer relevant multimedia or textual information. The exploration tools developers shall carefully select multimedia or textual information for the domain and make them available as part of the focus entity pages. For example, in MusicPinta instruments and performances related pages can contain YouTube videos of instruments or performances involving instruments.

3.4 Semantic Signposting in Semantic Data Browsers

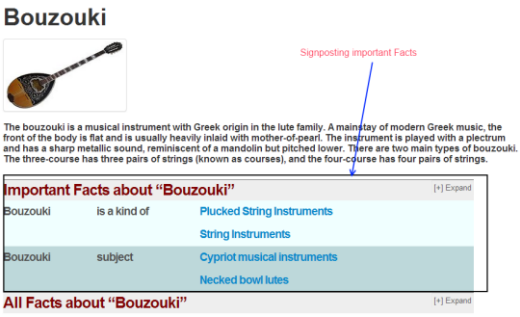


Figure 1. Showing important facts about "Bouzouki" as signposts.

From these requirements, we have identified the need for further algorithmic support to realise the exploratory search potential of semantic data browsers. There can be many possible ways to address these requirements. One of such possible novel approach is ***semantic signposting*** that we have implemented as part of the Dicode project.

In uni-focal exploration, a user focuses on one entity at a time represented in a page. This entity page contains links to various descriptions, images and links to other entities.

Such entity page can be treated as a juncture in the journey where the explorer has to make few choices (through the links which takes her to different paths). Semantic Signposting provides different types of signposts guiding the explorer in making a choice about paths she can take. An example is provided in Figure 1, where important facts are calculated based on the semantic graph and richness of content, and then signposted for user to view (instead of reviewing all the possible facts), hence improving usability and reducing cognitive overload.

4 Future Work Directions

Clearly, the focus of the Dicode project concerning data mining in data-intensive setting has been on batch processing and on improving batch processing using Hadoop. As reported in this deliverable, batch processing should be complemented by real-time analysis. This particularly holds for use cases such as public opinion monitoring where a batch cycle of a day or even half a day may be too slow to react immediately, for instance to curtail a "shitstorm". Successful experiments in this area have been conducted and reported in deliverable D7.2.2. More applications in this area and a deeper understanding of the necessary architecture need to be developed, complemented by an analysis of how to map common data mining tasks and algorithms for this architecture.

5 Conclusion

There are quite a number of lessons learned in the context of Dicode. The reported lessons have a different scope and importance. Some are quite project specific, while others go far beyond. Future projects may benefit by taking these into account, as is the case for the project partners concerning their individual research and working procedures.

Dicode partners already took advantage of some lessons learned, as discussed in the exploitation report appearing in deliverable D7.2.2 (Section 3). For instance, FHG demonstrated the Dicode Text Mining technology at CeBit 2013 in Hannover; NEO provides Big Data consultancy to SMEs and develops components for data mining services that are already used (e.g., in DBpediaSpotlight[34]).

[34] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki