High School Students Facing SynerScope

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

Fernando Núñez Serrano

June 2014
# Contents

1 Introduction ................................................. 4

2 SynerScope .................................................. 5
   2.1 "Big Data" ............................................. 5
   2.2 SynerScope’s Solution ................................. 5
   2.3 SynerScope Overview .................................... 6
      2.3.1 Terms and Definitions ............................ 6
      2.3.2 User Interface ..................................... 10
      2.3.3 SynerScope Views ................................. 11
      2.3.4 Supported Platforms ............................... 15

3 Pre-processing Data ......................................... 16
   3.1 Wikipedia ................................................. 16
      3.1.1 DBpedia ........................................... 18
   3.2 Hadoop .................................................. 18
      3.2.1 MapReduce and Hadoop File System ................. 19
   3.3 Apache Pig .............................................. 22
      3.3.1 Pig Latin ........................................... 23
   3.4 Wikipedia Scripts ....................................... 25
      3.4.1 Selected Nodes and Links ........................... 25
      3.4.2 Scripts .............................................. 26
      3.4.3 Wikipedia in bird’s eye view with SynerScope ....... 27

4 Experiment ................................................... 32
   4.1 Overview Sessions ....................................... 32
   4.2 Explanation .............................................. 33
   4.3 Evaluation Test ......................................... 33
      4.3.1 Using Wikipedia Website and Google ............... 34
   4.4 Feedback and Discussion ............................... 34
   4.5 Location and Execution ................................. 34
   4.6 Schools .................................................. 34

5 Results ........................................................ 36
   5.1 Using Wikipedia Website and Google ..................... 36
   5.2 Descriptive Model ....................................... 37
      5.2.1 Analyse Questions and Students ..................... 37
   5.3 Statistic Model .......................................... 44
      5.3.1 Model Answered Questions .......................... 47
      5.3.2 Model Correct Questions ............................ 48
   5.4 Analyse Observations and Feedback ...................... 51
      5.4.1 Observations ....................................... 51
      5.4.2 Feedback ............................................ 55
Chapter 1

Introduction

SynerScope\(^1\) allows users to show, analyse and understand relational “Big Data” visually by using edge-bundled and time line-enhanced visualizations. SynerScope visualization looks interesting, but complicated. The first contact with the tool can cause insecurity for people who are less experienced with computers. We think that SynerScope can be used smoothly by everyone, no matter the background. With this research we want to prove that. 63 students in 6 sessions between 12 and 18 years old from different high-schools in Amsterdam, used the program after a short explanation.

The main question we used to analyse the sessions is:

How can we make and explain SynerScope easier to use for people who are less experienced with computers?

To answer this question, we have to answer the following questions first:

1. What is the opinion of the students about SynerScope?
2. What are the most chosen strategies to search for the answer? Why?
3. Are the students’ capacities and age important understanding Synerscope?

In this thesis, we start explaining SynerScope and the way we pre-processed the data for this experiment. Then we will show how the experiment works. After that we will give the results and analyse them. In the conclusion, we will answer my main question.

\(^1\)http://whatissynerscope.com/, http://synerscope.com/
Chapter 2

SynerScope

SynerScope\(^1\) was founded in 2011 as a high-tech spin-off company of the Information Visualization group of Eindhoven University of Technology\(^2\). Synerscope is a data analysis company that provides advanced BI/”Big Data” analysis software that directly allows actual domain experts and analysts to make sense of their ”Big Data”.

2.1 ”Big Data”

We live in the data age. It is not easy to measure the total volume of data stored electronically, but an IDC\(^3\) estimate, from 2005 to 2020, the digital universe will grow by a factor of 300, from 130 exabytes\(^4\) to 40,000 exabytes (see Figure 2.1) or 40 trillion gigabytes (more than 5,200 gigabytes for every man, woman, and child in 2020). From now until 2020, the digital universe will about double every two years \cite{4}.

Thus, ”Big Data” is creating large and growing files and this data is measured in Terabytes and Petabytes. One a big issue of this data is unstructured. It is not an organized data in a relation database in nicely created tables that has columns and you exactly know which kind of value is going to go in a column, it is unstructured and that creates a challenges.

Where is this data coming from? It is coming from users like me and you, applications like Facebook, systems, sensors like in factories, banking data swiped in an ATM, security footage at airports, subatomic collisions recorded by the Large Hadron Collider at CERN, the astronomy records by the ASTRON Institute, transponders recording highway tolls, voice calls zipping through digital phone lines, and text as a widespread means of communications (see Figure 2.2). Therefore, all these things plus other are creating ”Big Data”, in other words, they are creating large and growing files.

2.2 Synerscope’s Solution

Much of this data builds networks or relationships. SynerScope is about discovering and understanding those networks, these can be networks of social media content, articles, phone calls and e-mails, but also payments, computers and biological networks. Synerscope effectively allows users to identify and diagnose deviant processes by visual analysis of massive amounts of data. For example, this solution can help insurers with workflow analysis, process improvement, network utilization, fraud diagnosis and Cyber-security

\(^1\)http://whatissynerscope.com/, http://synerscope.com/
\(^2\)http://www.tue.nl/
\(^3\)International Data Corporation, https://www.idc.com/
\(^4\)The exabyte (symbol EB) is \(10^{18}\) bytes of digital information
The human brain has far more processing power available for making sense of visual stimuli than tables of numbers. SynerScope uses sophisticated software to present massive amounts of data in a way that rapidly becomes intuitive with only a few days of training. As a telescope or microscope extend human vision, Synerscope extends human analytic capability. Another valid analogy lies in medical imaging, where an MRI or CAT scan takes digital output and presents it as a picture. SynerScope is not a mere presentation tool; the visuals are fully interactive so users can "drill down" (see Subsection 2.3.1) into their data with a few mouse clicks to carry out detailed analysis of abnormalities. Unlike normal data-mining approaches, Synerscope is not rule-based, but uses the innate power of the human perceptual system to analyze visual cues much more flexibly than a computer can [9].

2.3 SynerScope Overview

In this section, we explained some terms and definitions from SynerScope, the user interface and an overview from the most important visualizations used in this experiment.

2.3.1 Terms and Definitions

Node

Data entities under investigation, such as persons, bank accounts, IP addresses, insurance policies, URIs, or any other kind of entity that can be related to another entity by means of a link. Nodes are also referred to as actor. Nodes are static in nature, i.e., they do not
change over time. The example data set in Figure 2.3, displays persons as nodes:

![Figure 2.3: Source: SynerScope Help. Example with nodes representing persons.](image)

**Figure 2.3:** Source: SynerScope Help. Example with nodes representing persons.

**Link**

A link connects two nodes and indicates a directional event or relationship from one node to another. Links are also referred to as relationship, transaction, event, or hyperlink on a webpage. Examples are phone calls, bank transactions and network communication messages. Links often have a temporal attribute that indicates a link’s occurrence in time. SynerScope supports links with one or two different node types; a person can, for instance, be linked to a computer. In Figure 2.4 the relations between persons is shown, representing phone calls between persons.

![Figure 2.4: Source: SynerScope Help. Example with links between persons representing phone calls.](image)

**Figure 2.4:** Source: SynerScope Help. Example with links between persons representing phone calls.
2.3. SYNERSCOPE OVERVIEW

Hierarchy

A hierarchy is a classification of nodes by repeatedly dividing nodes into subcategories based on the values or bins of a single node attribute. In case of geographical data - where nodes represent cities, a valid hierarchy (shown from top to bottom) would be: continent - country - city. A city node is correctly placed within its hierarchy based on its continent and country attributes. The image 2.5 shows a hierarchy where persons are grouped using gender and institute on the first and second level respectively.

![Figure 2.5: Source: SynerScope Help. Example hierarchy on top of nodes representing persons, with gender at the first level and institute at the second level.](image)

Attribute

A property, quantity, or feature belonging to a node or link. For example, a node type person might have the following attributes: first name, surname, age, and city. A link type message might have the following attributes: content, time stamp, and duration.

Node Type

The structural description of a node with respect to attributes and possibly hierarchies. For example: a node can be a car, a node type named car can have as attributes weight, length, and color. As opposed to links, the values of these attributes do not change over time.

Link Type

The structural description of a link with respect to attributes. For example: a money transfer can be a link, a link type named money transfer can have as attributes amount, time stamp, and currency type.

Key

A key of a node type (2.3.1) is a set of attributes (2.3.1) that uniquely identifies each node (2.3) of that particular node type. Thus the combination of attribute values for a particular node is different for all other nodes of the same node type.

Comma Separated Value (CSV) file

Comma Separated Value files, commonly referred to as CSV files are text files containing a table where each value separated by a separation character. The separator does not have to be a comma, and can be any other character. Typically, a new line denotes a new row of the table. The number of values per row define the columns and should be the same throughout the document. Optionally, the first row contains the header (name) of the corresponding column.
Interaction Mode

The interaction mode defines the behaviour of mouse gestures and commands. SynerScope differentiates between normal, additive, and subtractive selection modes as well as a drill-down mode.

Highlight

Visually emphasizing nodes and their related links or links and their related nodes by moving the mouse cursor over them.

Indirect Highlight

The indirect highlight is an expansion from links to nodes for highlighted links or from nodes to links and links to nodes for highlighted nodes.

Selection

Selecting is the process of visually emphasizing nodes or links by persistently marking them by clicking (nodes and links) or click-dragging (links only). Selecting can be normal, additive (add to selection), or subtractive (subtract from selection) depending on the interaction mode.

Drilling (drill-down, drill-up)

Systematic manipulation of the visibility of links in the views. A drill-down hides links and/or nodes so the user can focus on specific parts of the data in isolation to reveal and inspect detailed patterns. A drill-up (re)adds links and/or nodes to show the data at an overview level to reveal summary patterns, trends, and outliers. The image 2.6 shows the drilling bottoms.

![Drilling](image)

**Figure 2.6:** drill-down and drill-up bottoms in the Interaction bar.

Project

A SynerScope project file contains settings for a single project. Project files can be directly opened by SynerScope.
Investigation

Typically, an analysis is performed by investigating the data from different perspectives. SynerScope supports this approach by allowing the user to have investigations. Investigations contain the following information:

- Definitions of node and link types.
- Prefiltering (2.3.1) steps to focus the investigation on a particular part of the data.
- Notes about the investigation and any findings already found.

Snapshot

Contains the current state of the analysis in the form of a data bookmark. Snapshots belong to a specific project; by activating a snapshot SynerScope depicts the state of the analysis (selection state, drill-down state, etc.) recorded within the snapshot.

Prefiltering

The process of applying filters to the data, even before it is shown in SynerScope.

Scoop

A local copy of the database with a reduced amount of data due to prefiltering.

Node-Link Diagram

A node-link diagram is a common graphical depiction of a graph or network in which entities (nodes) are represented as circles and relations (links) are represented as lines connecting at most two circles. Node-link diagrams are intuitive to users, but they do not scale well visually. These diagrams quickly become cluttered when more than a couple of hundred nodes and links are displayed.

![Node-link Diagram](image)

**Figure 2.7:** Source: SynerScope Help. An example of a node-link diagram.

2.3.2 User Interface

The SynerScope interface is divided into multiple re-sizable tiles. Each tile contains one or multiple views. If a tile contains multiple views, different views can be selected by clicking one of the tabs displayed below the currently visible view of the tile. The interaction bar is displayed on the left-hand side of the application. This toolbar is used for interactive exploration of the data. Please have a look at layout management of views and interactions for interactive exploration to see more details.
2.3.3 SynerScope Views

SynerScope uses multiple linked views to provide different graphical representations of the data.

Bundling View

The Bundling View shows nodes (see Subsection 2.3.1) and links (see Subsection 2.3.1) using the current hierarchy (see Subsection 2.3.1). Nodes are organized in multiple rings. The inner ring is composed of individual nodes. Each larger ring is composed of a decreasing number of aggregated nodes. Each ring corresponds to a layer in the currently selected hierarchy, and hence matches with a (group of) attribute values. The most outer ring shows the node type(s).

Nodes are connected using links. Instead of straight lines, links are bundled together using the currently selected hierarchy. The bundling of links is similar to bundling numerous cables together using a tie wrap.

Links are colored by direction: from green to red. Highlighted as well as selected nodes and links are colored using blue and orange respectively. Selected nodes and links that are highlighted are colored purple. In Figure 2.9 you can see the Bundling View.

Sequence View

The Sequence View shows nodes using the current hierarchy and links sorted on an attribute of the selected link type.

Nodes are placed at the top of the view and organized using the currently selected hierarchy. Links are placed below the nodes. The end-points of a link are determined by the horizontal position of the “From” and “To” node in the hierarchy. The vertical position is determined by the value of the selected link type attribute.

Links are colored by direction: from green to red. Highlighted as well as selected nodes and links are colored using blue and orange respectively. Selected nodes and links that are highlighted are colored purple. In Figure 2.10 you can see the Sequence View.
Figure 2.9: Source: SynerScope Help. The Bundling View shows the topology in which nodes are connect by links.

Figure 2.10: Source: SynerScope Help. The Sequence View shows links ordered by an attribute related to a link.
CHAPTER 2. SYNERSCOPE

2.3. SYNERSCOPE OVERVIEW

Table
The Table shows a textual representation of the data. Individual nodes or links of a particular node or link type are shown. Each column of the table displays a different attribute of the selected node type or link type. Only entries for a single type, either node type or link type are shown. Change the displayed type by clicking one of the buttons at the bottom of the view. Select All, Highlighted, Selected, or Highlighted/Selected to switch between displaying all entries or a particular subset: selected, highlighted or both highlighted and selected nodes or links. In Figure 2.11 you can see the Table view.

![Table view](image)

**Figure 2.11:** Source: SynerScope Help. The Table shows a textual representation of the data.

Hierarchy Editor
The hierarchy editor is used to add and modify hierarchies. This view is also used to change the current hierarchy that is used throughout the other views, such as the Bundling View and Sequence View.
The hierarchy editor consist of two parts. The top part is used to manage (create, select, remove, and rename) hierarchies. The bottom part is used to edit the selected hierarchy. All views use the same hierarchy (if applicable), by selecting a different hierarchy, all views will use the newly selected hierarchy. Only the selected hierarchy can be changed.

Search and Filter View
This view allows to set the current selection based on a search query. In Figure 2.12 you can see the Search and Filter View.

![Search and Filter View](image)

**Figure 2.12:** Source: SynerScope Help. The Search and Filter View allows to query an attribute for a specific value.

Scatter Plot
The Scatter Plot is used to relate two scalar attributes of either a node or a link type by drawing the nodes or links as dots on a two dimensional chart. The position on the
horizontal and vertical axis is determined by the attribute values. In Figure 2.13 you can see the Scatter Plot View.

![Figure 2.13](image)

**Figure 2.13:** Source: SynerScope Help. The Scatter Plot allows for multi-dimensional analysis.

### Web View

The Web View allows to see a web page from a URI attribute of a data node. It is possible to select all the links in that web page, search text inside the page or navigate like a normal browser. This is a first session and for this experiment it was only possible to open a URI from a node selecting a link. In Figure 2.14 you can see the Web View.

![Figure 2.14](image)

**Figure 2.14:** Own Source. The Web View.
2.3.4 Supported Platforms

SynerScope is available for the operating systems: Windows, Linux and OS X. Any machine with an Intel Core i5/i7 (or equivalent), at least 4GB RAM, and a dedicated NVidia graphics accelerator should be able to run SynerScope fine for datasets under 100K nodes (see Subsection 2.3) and 1M links (see Subsection 2.4).
Chapter 3

Pre-processing Data

In this section, we show the selected data set to be analysed in SynerScope and trained with the high-school students as well as all the programs used in the whole process in order to have the proper source (CSV file, see Subsection 2.3.1) to load in SynerScope. In Figure 3.1 you can see the system architecture.

We first explain about the Wikipedia data set. Then we talk about the software framework Hadoop. Then we explain one of the tools running under Hadoop, Apache Pig and we finally explain the realized work using the Pig Hadoop preprocessor scripts to load, analyze and transform the Wikipedia, DBpedia and Wikidata databases.

3.1 Wikipedia

What is a data set? A data set is really abstract, but really easy. We can explain better this matter with an intuitive example. If you are in the cash register of a supermarket,
each checked in product has one registered amount of money, so one amount is one data point and one receive with all the amounts together is a data set. There are so many kind of receives.

Like we said in the Introduction (see Chapter 1) the data set selected to be analysed with SynerScope is the English version of Wikipedia. In the case of the Wikipedia data set, one data point is a page (one node in SynerScope, see 2.3.1).

Wikipedia\(^1\) is the fifth most important web of the world. It is a collaboratively edited, multilingual, free Internet encyclopedia supported by the non-profit Wikimedia Foundation\(^2\).

Volunteers worldwide collaboratively write Wikipedia’s 30 million articles in 287 languages. In the English Wikipedia there are currently 4,479,857 articles and around 100 millions of links between them. The size of the 13 February 2014 dump is approximately 9.85 GB compressed, 44 GB uncompressed.\(^3\)

Wikipedia articles consist mostly of free text, but also include structured information embedded in the articles, such as "infobox" tables (the pull-out panels that appear in the top right of the default view of many Wikipedia articles, see Figure 3.2), categorisation information, images, geo-coordinates and links to external Web pages. This structured information is extracted and put in a uniform database (DBpedia) which can be queried.

The Wikipedia receives between 25,000 and 60,000 page requests per second, depending on time of day, from a lot of computers somewhere in the world. All the Wikipedia pages are all together in one database, the Wikipedia Software/Hardware deliver the page requested from the database.

It is possible to download a whole copy of the database in the website of Wikimedia (see Figure 3.3). Because everybody can change the Wikipedia pages just with an account, some people pay Wikimedia, the organization behind Wikipedia, to check that is correct and keep the pages updated. One tenth of Wikipedia is through away every year, there are some pages that you have to remove because they do not meet the criteria of relevance or quality encyclopaedic for Wikipedia and if nobody is protesting about it, they are removed. Like this we can get a good fill database and you can download in the website shown in Figure 3.3. You can see that there are different versions, they make copies from the older versions and you can also download them [11].

In this research we downloaded the English Wikipedia dump in XML (9.85GB)\(^4\), it con-

---


---

**Figure 3.2:** Source: [http://en.wikipedia.org/wiki/Dog](http://en.wikipedia.org/wiki/Dog)
3.2 Hadoop

Apache Hadoop is an open source software framework of tools that support applications on "Big Data". Hadoop dramatically simplifies writing distributed data-intensive applications.

So what is this framework use for? The objective of this tool is to support running on
3.2. HADOOP

Figure 3.4: Source: http://dbpedia.org/page/Dog

applications of "Big Data" (see Subsection 2.1). Hadoop is an open source set of tools and it is distributed under Apache License\(^9\), so this guarantees that not particular company is controlling the direction of Hadoop and it is maintained by Apache.

So the keyword behind Hadoop is "Big Data". "Big Data" is creating challenges that Hadoop is addressing. Challenges are created at three levels, lot of data is coming in a very high speed, **Velocity**, a big **Volume** of data has been get there and is growing and growing and this data is also too **Variety**. It is not an organized data. It has audio’s, videos, files, log files, etc.

In a traditional approach, an enterprise would get a very powerful computer and the "Big Data" would be processed by this powerful computer. This computer will do a good job, but until a certain point, a point that will come when that computer will not be able to processing any more because that is not scalable and the "Big Data" is growing, so this traditional Enterprise Approach does have its limitations when it comes to "Big data".

How does Hadoop solve this limitation? Hadoop breaks the data into smaller pieces and that is way it is able to deal with the "Big Data". Then the perform of the computation is as well break down into smaller pieces and it senses each piece of the computation with each piece of data. So the data is break down in equal pieces and each computation can be finished in equal amount of time. Once all the computations are finished, the results are combine together and that combined result is sent back to the applications as a combined overall result (see Figure 3.6).

### 3.2.1 MapReduce and Hadoop File System

How does Hadoop break the data and the computation into pieces? As a very high level you have a simple architecture in Figure 3.5. You can say that Hadoop has too main components: **MapReduce** and the **File System**. The File System is called HDFS (Hadoop File System). As we already said, Hadoop is a set of tools, we can call those tools projects, there are many projects that are been started and managed by Apache. The objective of these projects is to write the systems in tasks that are related to Hadoop.

One important feature from Hadoop is that works as a Distributed Model (see Figure 3.6). We are not talking about a big powerful computer, we are talking about numerous low cost computers. Hadoop is a Linux based, so we have Linux in all of those low cost

\(^9\)http://www.apache.org/foundation/
3.2. HADOOP

Figure 3.5: Source: http://hortonworks.com/hadoop/yarn/. High Level Hadoop Architecture.

computers. All these computer will have two components of Hadoop: The Task Tracker and the Data Node. The job of the Task Tracker component is to process the smaller piece of task that is been given to this particular node and the job of the Data Node is to manage this piece of data that has been given to this particular node. And all these computer are called slaves. They are called slave because we have a Master. The main difference between a Master and a Slave is that the Master has to additional components running on the computer. It will have also a Job Tracker and a Name node (see Figure 3.6).

The Job Tracker and the Task Trackers are part of the High level component MapReduce and the Name Node and the Data Nodes follow under the umbrella of Hadoop File System (HDFS). The applications that are running on Hadoop will contact the Master Node. One of the attributes of Hadoop it is that there is a batch processing set of tools. So applications would assign or provide a task for Hadoop to perform and is going to go in a queue. Once the task is completed the application will be informed and the results will be given back to the application (see Figure 3.6).

Figure 3.6: Own Source. Master-Slave, MapReduce, HDFS, Master Backup

The roll of the Job Tracker component running on the Master node is to break the higher bigger task into smaller pieces and to send each small piece of computation to the Task Tracker component. This component is in charge of breaking down the task into smaller pieces and then sending them to the Task Tracker components running on the slaves. These components will then process the data and return the results back to the Job Tracker component, which will then aggregate the results and send them back to the application.
Tracker. So the slaves will perform that small piece and it will be sent the result back to the Job Tracker, then the Job Tracker will combine all the small results together and it will send back the final result to the application (see Figure 3.6).

The Name node running in the Master node is responsible to keep an index of which data is resident and which data not. So when the application contacts the Name node, it tells the application: Go to this particular to get the data. So it has all the index. So the Name Node says to the application where is the data residing and goes directly to that node and get the data from that node (see Figure 3.6).

The hardware failures are bound to happen, but the good thing about Hadoop is that it is built keeping hardware failures in mind, it has a fault Tolerance for Data. By default Hadoop maintenances three copies from each file. Copies are scatter along different computers. So when a computer fails, the system keeps on running, data is available from different nodes, once you fix the fail node Hadoop will take care of that and it will copy some other files to that node. The fault tolerance is also applicable to the task record services that are running on slave computers, if any of the computers fail or even just the service fails, Job Tracker will detect that failure and ask to other Task Trackers to perform the same job.

If Master computer dies that it could be a single point of computer failure. Hadood has taken care of that as well, the tables that are maintain by the Name Node that has all the index where the data is residing, all those tables are backup and the backup copies are copied over to different computers (see Figure 3.6). Enterprise version of Hadoop also keeps two Masters, one is the main master and one is the backup master in case of master dies so that is also a single point of failure.

Thus, we have seen an overview about how Hadoop has solved those challenges providing by the ”Big Data” and about the architecture of Hadoop. The programming now is very easy for the programmers. The programmers do not have to worry about where the files are located, the Master node is there to manage that and you also they do not have to worry about how to manage the failures, Hadoop is taking care of that and also they do not have to worry about how to break the big computations into smaller pieces, that is a task from Hadoop. Also they do not have to do a programming for the scaling, wherever program written by for 1 MB of file will work for 100 GB file as well. So the programmers could focus on writing scale free programs.

Therefore, we know the two main components from Hadoop are the MapReduce and the HDFS. We also said that Hadoop is set of tools. In Figure 3.7 you can see all the tools that are made by Apache. Each of this piece of Software is providing a value added at certain area related to Hadoop.

Some of the areas where Hadoop is being used:

- Social Media: Yahoo, Facebook.
- Retail: Amazon, Ebay.
- Financial Services: Heavily use Hadoop.
- Search tools: Google and in other search tools as well.
- Government: Federal Reserve Board.
- Intelligence Agencies.
Thus, any area where we are dealing with "Big Data" is likely to use Hadoop. Some examples of the applications:

- Advertisement: Mining of users behaviour to generate recommendations.
- Searches: group related documents, indexing.
- Security: Search for uncommon patterns to detect fraud behaviour.

How does the future look for Hadoop? Yahoo says that by 2015 50% of all Enterprise data will be processed by Hadoop. This is clearly because of the limitation of the Enterprise Model, no matter how big the computer is, it will have limitations when it comes to "Big Data" because the data will come to one point where the computer can not deal with that size.

### 3.3 Apache Pig

We explained the general functioning of the two main components from Hadoop: MapReduce and HDFS. But we also said that Hadoop is a set of tools managed by Apache and they are under the umbrella of Hadoop (see Figure 3.7). These tools do add certain value to the functionality at multiple levels.

One of the tools that runs on Hadoop is Apache Pig (see Figure 3.8). Apache Pig uses both components: HDFS to read and write files and MapReduce to execute jobs. It has its own high-level scripting language called Pig latin. Pig is extensible through user-defined functions (UDFs) than can be written in java, python and other languages. Pig scripts provide a high-level language to create the MapReduce jobs needed to process in a Hadoop cluster [6].

Pig consists in three elements:

1. A language called Pig Latin, it is a high-level scripting language, it does not require metadata or schema and the statements are translated into a series of MapReduce jobs.
2. An interactive shell: **Grunt**.

3. A server repository for User Defined Functions (UDFs) created and maintained by the community called **Piggybank**.

Pig can also be extended with jar files that are created by the developers themselves.

### 3.3.1 Pig Latin

Pig Latin is a language for expressing data analysis and infrastructure processes. It supports many traditional data operations like join, sort, filter, etc. Simplifies the joining of data and the chaining of jobs.

Pig Latin statements (see Figure 3.9) are translated into one or more MapReduce jobs, with HCatalog traditional Pig data flow paradigm changes slightly. HCat holds the location and metadata information. Therefore, there is not needed to Pig script to handle the
3.3. APACHE PIG

CHAPTER 3. PRE-PROCESSING DATA

data flow. As you can see in Figure 3.9, we can read data to be manipulated from the file system with HCat. We manipulate the data with Pig and output the data to the screen or store for processing also using Pig.

Pig Latin script describes a directed acyclic graph (DAG). Looking at Figure 3.10, the edges are data flows and the nodes are operators that process the data. Pig goes through several steps to transform to a Pig Latin script to a set of MapReduce jobs (see Figure 3.11). The steps are as follows:

1. The Pig interpreter immediately processes each entry.
2. If a statement is valid, it gets added to a logical plan (built by the interpreter).
3. The steps in the plan do not execute in MapReduce until a DUMP or STORE command.

![Figure 3.10: Source: http://hortonworks.com/. Pig Latin Structured Processing Flow.](http://hortonworks.com/)

<table>
<thead>
<tr>
<th>Interprets</th>
<th>Each entry</th>
<th>Executes in MapReduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_selection = LOAD '$LINKS' USING PigStorage('r') AS (from:chararray, to:chararray);</td>
<td>sources = FOREACH link_selection GENERATE from AS uri; targets = FOREACH link_selection GENERATE to AS uri; node_selection0 = UNION sources, targets; node_selection = DISTINCT node_selection0;</td>
<td>STORE node_selection INTO 'SNODES' USING PigStorage('w');</td>
</tr>
</tbody>
</table>

![Figure 3.11: Source: Own source. Pig Latin Script to MapReduce job.](http://ownsource.com/)

Pig Latin statements work with relations. A relation can be defined as follows (see Figure 3.12):

- A relation is a bag (more specifically, an outer bag).
- A bag is a collection of tuples.
- A tuple is an ordered set of fields.

24
• A field is a piece of data.

![Image of data types: Tuple, Bag, Map]

**Figure 3.12:** Source: Own source. Pig Latin Relations.

Pig has three complex data types:

- **Tuple**: ordered set of values:

- **Bag**: Unordered collection of tuples:
  {("25-11-1986", ”ERROR”, 404, ”Page not found”),
   ("25-11-1986", ”ERROR”, 200, ”OK")}

- **Map**: Collection of key value pairs:
  [firstName#Cary, lastName#Grant, id#123].

### 3.4 Wikipedia Scripts

Using a Pig preporcessor script and python UDFs, the Wikipedia pages and the DBpedia records are loaded, analyzed and transformed in order to have the proper source information in SynerScope (see Figure 3.1).

#### 3.4.1 Selected Nodes and Links

The data entities (nodes) in this Wikipedia investigation are **pages** and the relationships between the nodes are **links**.

The size of the English Wikipedia XML dump is approximately 9.85 GB compressed, 44 GB uncompressed. This dump contains 4.479.857 pages and around 100 millions of links between them. A computer that conforms the Hardware/Software requirements to use SynerScope (see Subsection 2.3.4) can run smoothly for datasets under 100K nodes and 1 million links. So we select a reduce part of Wikipedia to run the project without problems in normal computers with the schools. The three selected investigations with interesting contents for the High-School students are:

1. **Soccer Players**: Soccer Player’s pages and soccer manager’s pages from: AFC Ajax, F.C Barcelona, Manchester United FC and Manchester City FC.

2. **Scientists**: Scientists’ pages from the field Physics.
Musical Artist: Musician’s pages from the Pop music genre.

In addition to these pages, the dataset also includes pages from which are being linked to the selected pages or are linked to (In and Out links). Furthermore, the dataset contains all the links between all the pages listed (Interrelations) and an augmented random number of pages out from the selection. In Figure 3.13, you can see an intuitive reduced example from this selection. The two pages from the selection are the F.C Barcelona soccer players Ronaldo and Leonel Messi, we take all the in and out links from these two pages (in Figure 3.13, you can only see a few links with grey arrows) and also the interrelations between all the pages listed, that is the red arrows in Figure 3.13.

![Node-Link Diagram](image)

**Figure 3.13:** Source: Own source. Node-Link diagram from Barcelona Soccer Players: Leonel Messi and Ronaldo.

### 3.4.2 Scripts

In Figure 3.14, you can see a diagram with all the scripts from the project.

First of all, the Wikipedia XML English dump file is split into 100MB XML files using Mahout\(^\text{10}\).

We use PigNLProc\(^\text{11}\), a collection of Pig Latin scripts and utilities focused on Wikipedia and DBpedia to load and extract the data from the English Wikipedia XML dump [7].

With a total of five scripts (see the program codes in Appendix D), we extract all the wikipedia pages plus several other attributes like the DBpedia categories from each page (Person, PopulatedPlace, Event, etc.). Below you have a sort number list with all the operations done with each Pig Script, the numbers are referenced in Figure 3.14.

1. We extract the URIs and their DBpedia types (hierarchies), the output is stored in *dbpedia_types*.
2. We extract the whole internal links graph from Wikipedia. We define the links between the pages with two fields, one “from” field and one “to” field. The output is stored in *wikipedia_links*.

---


\(^{11}\)//github.com/ogrisel/pignlproc
3. We load the selected links from each investigation (SoccerPlayers, Scientists and Musical Artists, see subsection 3.4.1) and we join with all the links from the Wikipedia XML dump, *wikipedia* links (number 2). We finally extract all the in and out links from the selection and the interrelations links. The output is stored in *inter*$_{FILTER}$.

4. We load the links calculated in the point 3 and it is added an augmented sample of links out from the selection. The output is stored in *inter*$_{aug}$$_{FILTER}$. This output is used to defined the relations (links) in SynerScope.

5. In this last script, we join everything together to define what is a node (a page in this case). We calculate and load different attributes from the nodes. We load the links selection calculated in point 4, *inter*$_{aug}$$_{FILTER}$, and we join to all these nodes selected the following:

- The *title* from the Wikipedia article, a field called *unique*, it is the number of words in the Wikipedia article and a field called *length*, it is the total number of words from the Wikipedia article.
- *in_degree*: Number of in links from the page and *out_degree*: Number of out links from the page.
- We add the DBpedia hierarchies. We only take a hierarchy depth of three: *hierarchy_one*, *hierarchy_two* and *hierarchy_three*.
- We add the *birthdate* and *deathdate* field available for some pages with a *hierarchy_one* equal to *Person*. Additional to this attributes we join some more attributes interesting for each investigation, you can see all the selected attributes from each investigation in Appendix B.

The nodes with all their columns are stored in *nodes*$_{FILTER}$.

### 3.4.3 Wikipeda in bird’s eye view with Synerscope

We finally have two text files containing all the selected nodes and links with their attributes. Now we can start to use SynerScope. In Figure 3.15, you can see an overview of the SynerScope workflow.
3.4. WIKIPEDIA SCRIPTS

CHAPTER 3. PRE-PROCESSING DATA

Figure 3.15: Source: SynerScope Help. Overview of the SynerScope workflow.

Source Selection

We use the data from Comma Separated Value (CVS) files (see Subsection 2.3.1). One file for the nodes and one file for the links.

Data Definition

We define the node type (see Subsection 2.3.1) importing the CSV file \textit{nodes\_FILTER}. We select the attributes of interest and we define the \textit{key} attribute that uniquely identifies each node (see Subsection 2.3.1), in this case is the \textit{uri} field (see Figure 3.16).

We define the link type (see Subsection 2.3.1) importing the CSV file \textit{inter\_aug\_FILTER}.

Figure 3.16: Source: Own Source. Node type Definition.

We select the only two attributes, the \textit{from} field and the \textit{to} field (see Figure 3.17).

We do the same process for the three different investigations: \textit{Soccer Players}, \textit{Scientists} and \textit{Musical Artists}. Therefore, we have three investigations as you can see in Figure 3.18.

Exploration and Analysis

Once we defined the data we can start using SynerScope to visualize and interact with the application to analyse the Wikipedia data set (see Figure 3.19).

28
CHAPTER 3. PRE-PROCESSING DATA

3.4. WIKIPEDIA SCRIPTS

Figure 3.17: Source: Own Source. Link type Definition.

Figure 3.18: Source: Own Source. The Three Investigations.
The first thing we have to do is to define the selected hierarchy. We have three attributes called: \textit{Hierarchy\_one}, \textit{Hierarchy\_two} and \textit{Hierarchy\_three}. The inner ring is composed of individual nodes, pages in this case. Each larger ring is composed of a decreasing number of aggregated nodes, since \textit{Hierarchy\_one} until \textit{Hierarchy\_three}. For example in Figure 3.20, the hierarchy of the page \textit{Lionel Messi} is: Person-Athlete-SoccerPlayer.

In Figure 3.20, you can see the network visualization from the example explained in the node-link diagram in Figure 3.13. That contains all the in and out links from the two Wikipedia pages, Ronaldo and Leonel Messi, and also all the interrelations.

\textbf{Figure 3.19:} Source: Own Source. Soccer Players Data Set in Wikipedia.
Figure 3.20: Source: Own Source. Ronaldo and Leonel Messi Network with in, out links and interrelations in the Bundling View.
Chapter 4

Experiment

For this research we want to find out how we can make and explain SynerScope easier to use for people who are less experienced with computers. 63 students in 6 sessions between 12 and 18 years old from different high schools in Amsterdam, used the program after a short explanation.

We have chosen high school students because they are not contaminated by the business world, so they are not apprehensive to try out new software.

We have a varied group of students; from different ages, levels and capacities/experiences. We will analyse whether the age is a matter understanding Synerscope and we chose high schools students from different courses. Thus, they had different levels in the course computing. We will check its dependence in the results.

After an explanation about the program the students made a test with 10 questions from three investigations, to test if they understood the program and how they interact with SynerScope. While the students were making the test, we were observing them. After the evaluation test we spent approximately half an hour on feedback and discussion with the students about the program.

![School and SynerScope]

Figure 4.1: Own Source. SynerScope Experiment with the Schools

4.1 Overview Sessions

Here we present an overview of the sessions. The duration of the experiment was not the same in all the sessions due to the particularities with the schedule from the different school groups.

We made the following schedule:

1. Welcome and Explanation: 35 minutes.
2. Evaluation test: 75 minutes.
3. Feedback and Discussion: 30 minutes.

4.2 Explanation

The first part of the experiment is a presentation of 35 minutes. During this presentation, we explained the following:

- What is SynerScope?
- The Wikipedia data sets selected (SoccerPlayers, Scientists and MusicalArtists)
- Demonstration with SynerScope showing the interface. We showed them the different visualizations, how to interact with the tool and we solved an example question similar to the questions in the Evaluation Test.
- Explanation about the experiment.

It is emphasised that the most important is not about answering the questions correctly, but how they interact with the program.

4.3 Evaluation Test

After the explanation the students started the evaluation test. The test consists of ten questions: 4 questions about the Soccer Players investigation, 3 questions about the Scientists investigation and 3 questions about the Musical Artists investigation. With the test, they received a hand-out where we explained from each investigation the selected links and the nodes with all the attributes of interest in the investigation (see Appendix B).

We scale the difficulty of each question with a value from 1 until 5. This scale is based on the number of steps to get the answer and their complexity. In the table below, you can see the number of steps to get the answer from each question and another column that represents whether there is an extra complexity in each question and why. With this two attributes, we decided the final scale for the complexity of each question from 1 until 5. In Appendix C, you can see the questions solutions and all the steps to get the answer.

<table>
<thead>
<tr>
<th>Question</th>
<th>Number Steps</th>
<th>Extra Difficulty</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>NO</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>NO</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>YES. Relate two visualizations.</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>NO</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>NO</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>NO</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>NO</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>YES. Search and filter two times in a difficult way.</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>NO</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>NO</td>
<td>1</td>
</tr>
</tbody>
</table>

During the test, the students were allowed to ask questions. We did not have a limitation of time, but the average duration was 75 minutes.
4.3.1 Using Wikipedia Website and Google

In the first session there were 15 students. They were working in 6 couples and 1 trio. Because only 4 OTOY instances were running Synerscope in the Intertain Lab (see Section 4.5), we split the students in 2 groups. In one group the half was working with SynerScope and in the other group the other half was working with Wikipedia and Google. We will show the results using Wikipedia in the Chapter 5.

4.4 Feedback and Discussion

After the Final Evaluation we spent half an hour on feedback and discussion with the students. We asked each student to tell us about:

- What do they think about the program?
- What do they think about the way of interacting and organizing the program?
- What was easy? What was difficult?
- What was clear? What was unclear?
- If they have any other comments.

4.5 Location and Execution

There were two ways and two places to show Synerscope to the schools (see Figure 4.2):

- Intertain lab at the VU: The Intertain lab at the VU has 12 computers. All the sessions in this lab were done with the Amazon Cloud OTOY + AWS system. We could have different instances running, a maximum of nine due to the limitation of IP addresses, in the server and they can be used by Internet in a computer. The performance with the cloud instance was unstable and some of the instances worked slowly. This issue will be taken into account in the results. In some session in this lab, some couples with problems in the computer were using a MacBook with a perfect performance, this will be taken into account in the statistics results.

- Game Cella at the VU: We installed locally a version of SynerScope in five computers from the Game Cella. The performance in this case was very acceptable.

4.6 Schools

To get the participants of this research, we sent a motivation letter to Informatics teachers of 10 different high schools in Amsterdam (see Appendix A). We decided to use the teachers who answered the first. The three schools we worked with were:

- Damstede Amsterdam (Amsterdam-North):
  Number students: 33
  Courses: 5VWO and 4VWO.
  Dates: 12th March and 2nd April.

---

1http://www.networkinstitute.org/tech-labs/intertain-lab/
2https://aws.amazon.com/marketplace/seller-profile?id=795808b7-f99d-426d-bb03-8aa79ff5b65e
3https://aws.amazon.com/
4http://www.networkinstitute.org/tech-labs/game-cella/
5http://damstede.mwp.nl/Home.aspx
Figure 4.2: Own Source. Locations from the Experiment: Intertain Lab and Game Cella

- **Pieter Nieuwland College**\(^6\) (Amsterdam-East):
  Number students: 20
  Courses: 5VWO, 4VWO, 4HAVO.
  Dates: 15th April and 17th April.

- **Cygnus Gymnasium**\(^7\) (Amsterdam-East):
  Number students: 14
  Courses: 1GYM.
  Dates: 6th May.

\(^6\)http://www.pieternieuwland.nl/
\(^7\)http://www.cygnusgymnasium.nl/
Chapter 5

Results

In this chapter, first we will show the results from the session where the students tried to answer the same evaluation test with Wikipedia and Google. After that we will give and analyse a descriptive model of the results from the evaluation tests and from the different groups of students. Then we will do a statistic model to test the obtained results from the descriptive model. Then we will give and analyse the observations and the feedback from all the sessions with the high school students using SynerScope. We will finally give a summarize where we answer the research sub questions.

5.1 Using Wikipedia Website and Google

Like we explained in Chapter 4, there was one session where the students were answering the evaluation test the half of the session with Wikipedia and Google. The most of the students left all the answers empty or they just wrote: “Impossible, not found, impossible to find in Wikipedia, etc.”.

Looking at Figure 5.1, only one group from the seven groups of that session answered correctly one question (question 7). This question is about doing yourself a new question about the Scientists Investigation, the answer was: “Question: Which Continent has the most links from Scientists?, answer: Europe”.

\[\text{Figure 5.1: } \% \text{ Answered Correctly, Incorrectly and Not Answered using Wikipedia, Google, etc.}\]
In regard to all the sessions, we did not do more sessions where the students were using Wikipedia and Google, but we discovered that some groups did not have query knowledge and they believed that it was possible to answer the questions with Google, which it is not possible, at least not as fast as with SynerScope.

5.2 Descriptive Model

5.2.1 Analyse Questions and Students

In total 63 students used SynerScope in 6 sessions. In 5 sessions the students worked in groups (couples or trios) on one computer. Only in one session 5 students worked alone. In total, 32 evaluation tests were done (see Section 4.5).

In the following paragraphs, we will give a descriptive analyse of every part from the evaluation tests. First the questions of the tests, after that the groups of students who did the tests.

Analyse Questions

In Figure 5.2, you can see the percentage of answered questions from all the sessions. From the different investigations, the investigation of Soccer Players was the most answered (75%). We expected this result because 66% (see Figure 5.3) of the students started with the Soccer Players investigation and in the beginning they had more energy and motivation. Just like we expected the last investigation, Musical Artists, is the least answered (25%). Question 8 from this investigation was the least answered by students (13%), only 4 groups tried it. There were only four chosen orders as you can see in Figure 5.3.

In Figure 5.4, you can see a general description with the percentage of answered correctly, incorrectly of the questions answered. The students answered about 60% of the
5.2. DESCRIPTIVE MODEL

CHAPTER 5. RESULTS

Figure 5.3: % Started by each Combination and each Investigation.

Figure 5.4: % Answered Correctly, Incorrectly from the Questions Answered.
questions correctly.
In Figure 5.5, you can see percentage of answered correctly, incorrectly and not answered from each question and each investigation. In Figure 5.6, you can see the difficulty class from 1 until 5 of each question (see table in Section 4.3) next to the percentage answered correctly, the percentage answered per question and the percentage answered correctly from the questions answered.

Question 3 was answered correctly by 70% of the students (see Figure 5.4). We expected that would be answered incorrectly by the most of the students because it is the only question where they had to relate two scalar attributes in the Scatter Plot View (see Subsection 2.3.3) and resolve which category had a higher distribution from one of the attributes. It is a high level question, because you have to interact with two visualizations at the same time. Figure 5.6 shows that question 8 was answered only by 13% of the groups and only 25% was answered correctly. This result is not very remarkable, because there is only one way to answer this question and you have to search and filter two times in a difficult way.

The 80% of the students answered correctly the question 6 and 9. We expected this result. Those questions are the same kind of questions where you have to relate two categories of nodes and extract the node with the major number of in-coming or out-going links. Question 5 is also the same kind of questions but only 50% of the answers were correctly. This percentage was lower because we asked for a list of the three countries with the most links from Scientists and a lot of students only typed one. The easiest questions were 4, 7 and 10. The students had to make a new question themselves. They had to investigate and find a question. Because of that the percentage answered from these questions was low. From the students who answered them almost nobody thought about a different type of question. They used the same kind of questions, like the questions in the test and as simple as possible, for example: "Which city is the
5.2. DESCRIPTIVE MODEL

CHAPTER 5. RESULTS

Figure 5.6: Difficulty class (1-5), % Answered Correctly, % Answered per question and % Answered Correctly from Questions Answered.

most linked from Scientists?”, ”Which VideoGame is the second most linked to Musical Artists?” We only saw one student thinking about a more difficult question with the Scatter Plot, linking up the Birthdate attribute and the length attribute.

Why did the students make mistakes in those questions? The most errors in questions 1 and 5 were due to select the outgoing or incoming links from the wrong category. Thus, for example in question 1 they had to search the youngest Soccer Player who is playing or played in Ajax. So many students got the youngest Soccer Player who linked out from the page of Ajax. Question 2 was made incorrectly many times, because the most of the students just typed in all the Soccer Players who died in Paramaribo in a plane crush, but they did not think about whether they played in Ajax.

Time

In Figure 5.7, you can see the average of the time the students needed to find the answer per question next to the difficulty class from 1 until 5 of each question. The students took on average 7 minutes to answer one question. It shows that in the first four questions the average time was higher. This happened with almost all the students started from the question 1 (see Figure 5.3) and in the beginning they needed a warm up time to get used to work with the program. Actually, the question 1 took the longest mean time.

In Figure 5.7 also shows that the questions 1, 2, 3 and 8 with a higher difficulty value took more time. The questions with a difficulty value of 2 took less time and the same kind of questions: 4, 7 and 10 with a difficulty value of 1 did not have a similar duration. Question 4 from Soccer Players investigation took more time than the question 7. This is
probably because question 7 was made after two easier questions than question 4. In Figure 5.8, you can see the mean time and the percentage answered per question for the four orders of questions. It looks that the students who started with investigation 3 (order 3-1-2) answered less questions. They started with the difficult question 8 and nobody was able to answer it. We will test in the statistic model whether the chosen order is depended for the percentage answered and answered correctly of the questions.
Analyse Groups of Students

In the following table you can see each level, age and computing experience per group. This is important for the results. The higher the group, the more years they have had the course computing. But is that necessary to understand SynerScope? You will read that in the results.

<table>
<thead>
<tr>
<th>Session</th>
<th>School</th>
<th>Level</th>
<th>Age</th>
<th>Years Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Damstede</td>
<td>5VWO</td>
<td>16-18</td>
<td>2.5</td>
</tr>
<tr>
<td>2</td>
<td>Damstede</td>
<td>4VWO</td>
<td>15-16</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>Pieter Nieuwland College</td>
<td>4HAVO/5VWO</td>
<td>15-18</td>
<td>1.5/2.5</td>
</tr>
<tr>
<td>4</td>
<td>Pieter Nieuwland College</td>
<td>4HAVO/5VWO</td>
<td>16-17</td>
<td>1.5/2.5</td>
</tr>
<tr>
<td>5</td>
<td>Pieter Nieuwland College</td>
<td>4HAVO/4VWO</td>
<td>15-16</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>Cygnus Gymnasium</td>
<td>1GYM</td>
<td>12</td>
<td>0.5</td>
</tr>
</tbody>
</table>

In Figure 5.9, you can see the percentages from the 32 tests that were made. Only group 30 did not answer any question correctly. This group was a trio of 12 years old students who answered three questions in total which were incorrect. With the three answers they gave, we could see that they knew how to work with the program, but apparently they did not have motivation to try to answer more questions.

**Figure 5.9:** % Answered Correctly, Incorrectly and % Not Answered from the 32 groups tested.
Observing the students, we realized that once they found something they thought that was the right answer, they were fairly quick to accept it. They did not try to do any verification, but they just accepted the first thing they found as their answer and as fast as possible. For example in question 2, where the students needed to find out how many Ajax players died in Paramaribo, they took all the soccer players who died in Paramaribo without checking if they played in Ajax.

In Figure 5.10, you can see a graph with the percentage answered and answered correctly from the answered questions plus a line age graph (from 12 until 18 years old). It is very remarkable that one of the best groups tested was one trio of three girls of 12 years old. They answered the 60% of the questions with the 100% correct. It is also remarkable that the session with the most percentage answered was the session 5 where the students worked individually.

The table below shows the number of students from each age from all the participants:

<table>
<thead>
<tr>
<th>Age</th>
<th>Number of Students</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>3</td>
<td>5%</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>16%</td>
</tr>
<tr>
<td>16</td>
<td>21</td>
<td>33%</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>24%</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>22%</td>
</tr>
</tbody>
</table>

The average age is 15.2 years old. 16 years old was the most abundant (33%) age from the 63 students. In figure 5.11, you can see the percentages answered correctly, incorrectly and not answered by the age of the students. It looks like as older the students are they answered more questions. It also looks that there is not a difference in the percentage of
Figure 5.11: % Answered Correctly, Incorrectly and Not Answered by the Age of the 32 groups tested.

correct answers between the age. We will test these estimations in the statistic model.

The following table shows the number of tests from each level, from the highest (5VWO) until the lowest class (1GYM):

<table>
<thead>
<tr>
<th>Level</th>
<th>Tests</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>5VWO</td>
<td>7</td>
<td>22%</td>
</tr>
<tr>
<td>4VWO</td>
<td>12</td>
<td>38%</td>
</tr>
<tr>
<td>4HAVO</td>
<td>7</td>
<td>22%</td>
</tr>
<tr>
<td>1GYM</td>
<td>6</td>
<td>19%</td>
</tr>
</tbody>
</table>

In Figure 5.12, you can see the percentages answered correctly, incorrectly and not answered by the level of the students. It looks like as lower the students level is they answered less questions. It also looks that there is not a big difference in the percentage correct between the different levels. We will test these estimations in the statistic model.

A general problem we found out during the sessions was that the students did not read the explanation we added in the hand-out. In the hand-out, all the attributes from each investigation were explained. It was easier to answer some questions knowing all the attributes (see Appendix B).

5.3 Statistic Model

We want to estimate with a statistic model the variance of two response variables (dependent variables): answered questions and answered correctly questions with respect to other predictor variables (independent variables) like: Age, course (level), members number doing the experiment, location where the experiment was done, questions order chosen and time length of the experiment.
CHAPTER 5. RESULTS

5.3. STATISTIC MODEL

Figure 5.12: % Answered Correctly, Incorrectly and Not Answered by Students Level.

The statistic analyse chosen is a simple linear regression for the variances of the response variables of interest. Specifically, we use Ordinary Least Squares (OLS). It is a method for estimating the unknown parameters in a linear regression model. This method minimizes the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation (see Figure 5.13)[12].

We have a number of observations \((n)\) \((y_i, x_i)_{i=1}^n\). Each observation contains a response \(y_i\) and a vector of \(p\) predictors (or regressors) \(x_i\). In a linear regression model the response variable is a linear function of the regressors:

\[
y = X\beta + \xi
\]

(5.1)

Where \(\beta\) is a vector of the regression coefficients that represents the rate of change of one variable \(y_i\) as a function of changes in the \(x_i\); it is the slope of the regression line.

After we have estimated \(\beta\), the predicted values from the regression will be

\[
\hat{y} = X\hat{\beta}
\]

(5.2)

Then we estimate the standard error of the regression (SER). It is common to assess the goodness-of-fit of the OLS regression by comparing how much the initial variation in the sample can be reduced by regressing onto \(X\). The coefficient of determination \(R^2\) is defined as a ratio of ”explained” variance to the ”total” variance of the dependent variable \(y\):

\[
R^2 = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}
\]

(5.3)

It is the percentage of variance of the response variable explained by the predictor variables used in the model. \(R^2\) will be a number between 0 and 1, with values close to 1 indicating a good degree of fit.

After the calculation of the OLS coefficients \(\beta\) and the standard error of the regression (SER), we do a t-test. A t-test is any statistical hypothesis test in which the test statistic
follows a Student’s $t$ distribution if the null hypothesis is supported. It can be used to determine if two sets of data are significantly different from each other. Assuming the model fits the slope of a regression line (see equation 5.1)\cite{13}. Then:

$$t_{\text{score}} = \frac{\hat{\beta}}{SE_{\hat{\beta}}} \tag{5.4}$$

We chose two different response variables ($y$) to match the regression model. The variable \texttt{Questions\_Correct} and the variable \texttt{Question\_Answered}. This variables have a value 0 or 1. Therefore, the OLS coefficients calculated may not be interpretable, but we can explain the sign ($+$, $-$) of the variance respect to the response variables. We will only use the coefficients values to compare them with other predictor variables.

We also highlight the statistical significance from each predictor variable. That is, the low probability that an observed effect would have occurred due to chance. If the probability of such chance effects is less than a pre-determined threshold (e.g. 5% chance), then an investigator can conclude that the observed effect actually reflects the characteristics of the population rather than just sampling error. The selected significance levels in this research are 90%, 95% and 99%.

The chosen predictor variables ($X$) will be shown in the definition of each model in the following subsections. The predictor variables \textit{Age} and \textit{Course} are highly correlated. Thus, we will do one model with each variable in order to have a robust design and avoid the multicollinearity.

You can see a spreadsheet with all the variables and the calculated data in the footnote link\footnote{https://drive.google.com/?usp=chrome_app#folders/0Bz0doRAU3cW3WWJnSXZSQWhZbms}. 

---

\cite{13}
5.3.1 Model Answered Questions

In this model, we used as response variable $y = \text{Questions}_{\text{Answered}}$. We have one model with the age predictor variable and other with the course predictor variable. We have 63 students by 10 questions = 630 observations. We also expand the sample by 10 (6300 observations). In Tables 5.14 and 5.15 we show the results obtained for the Questions Answered model. In the table 5.14 we introduce age as dependent variable and in table 5.15 this variable is replaced by course. It is not possible to introduce this two variables jointly because they show multicollinearity.

### Age

In the table of Figure 5.14, you can see all the predictor variables and their values of the OLS regression coefficients.

Based on the obtained results in Table 5.14, we can say that the raise of members reduces the number of answered questions. It is totally robust because it is always negative and 99% significant in the most of the cases. We can say that an increase of age, also increases the number of answered questions. All the orders increase significantly the number of questions answered with respect to 3-1-2. The variable $\text{length}_{\text{experiment}}$ is always positive and 99% significant. Nevertheless, the value is always $0.000^{***}$ because this variable is measured in seconds and the increase of one second in the length does not increase the number of answered questions. Of course, we can say that with an increase of time the

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>members</td>
<td>-0.124***</td>
<td>-0.124***</td>
<td>-0.048</td>
<td>-0.048***</td>
<td>-0.124***</td>
<td>-0.124***</td>
<td>-0.048</td>
<td>-0.048***</td>
</tr>
<tr>
<td>age</td>
<td>0.000</td>
<td>0.000</td>
<td>0.036**</td>
<td>0.036**</td>
<td>0.000</td>
<td>0.000</td>
<td>0.036**</td>
<td>0.036**</td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-2-3</td>
<td>0.310***</td>
<td>0.310***</td>
<td>0.239*</td>
<td>0.239***</td>
<td>0.310***</td>
<td>0.310***</td>
<td>0.239*</td>
<td>0.239***</td>
</tr>
<tr>
<td>2-3-1</td>
<td>0.267**</td>
<td>0.267**</td>
<td>0.206*</td>
<td>0.206***</td>
<td>0.267**</td>
<td>0.267**</td>
<td>0.206*</td>
<td>0.206***</td>
</tr>
<tr>
<td>3-1-2</td>
<td>0.398***</td>
<td>0.398***</td>
<td>0.241</td>
<td>0.241***</td>
<td>0.398***</td>
<td>0.398***</td>
<td>0.241</td>
<td>0.241***</td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game Cell lab</td>
<td>0.118</td>
<td>0.118***</td>
<td>-0.129</td>
<td>-0.129**</td>
<td>0.118</td>
<td>0.118***</td>
<td>-0.129</td>
<td>-0.129***</td>
</tr>
<tr>
<td>Intertain lab</td>
<td>-0.156**</td>
<td>-0.156***</td>
<td>-0.121</td>
<td>-0.121***</td>
<td>-0.156**</td>
<td>-0.156***</td>
<td>-0.121</td>
<td>-0.121***</td>
</tr>
<tr>
<td>MacBook</td>
<td>-0.096***</td>
<td>-0.096***</td>
<td>-0.096</td>
<td>-0.096***</td>
<td>-0.096***</td>
<td>-0.096***</td>
<td>-0.096</td>
<td>-0.096***</td>
</tr>
<tr>
<td>Answer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>answer_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>answer_2</td>
<td>-0.096</td>
<td>-0.096**</td>
<td>-0.096</td>
<td>-0.096***</td>
<td>-0.096</td>
<td>-0.096**</td>
<td>-0.096</td>
<td>-0.096***</td>
</tr>
<tr>
<td>answer_3</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254</td>
<td>-0.254***</td>
</tr>
<tr>
<td>answer_4</td>
<td>-0.567***</td>
<td>-0.567***</td>
<td>-0.567</td>
<td>-0.567***</td>
<td>-0.567***</td>
<td>-0.567***</td>
<td>-0.567</td>
<td>-0.567***</td>
</tr>
<tr>
<td>answer_5</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317</td>
<td>-0.317***</td>
</tr>
<tr>
<td>answer_6</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381</td>
<td>-0.381***</td>
</tr>
<tr>
<td>answer_7</td>
<td>-0.409***</td>
<td>-0.409***</td>
<td>-0.409</td>
<td>-0.409***</td>
<td>-0.409***</td>
<td>-0.409***</td>
<td>-0.409</td>
<td>-0.409***</td>
</tr>
<tr>
<td>answer_8</td>
<td>-0.899***</td>
<td>-0.899***</td>
<td>-0.899</td>
<td>-0.899***</td>
<td>-0.899***</td>
<td>-0.899***</td>
<td>-0.899</td>
<td>-0.899***</td>
</tr>
<tr>
<td>answer_9</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714</td>
<td>-0.714***</td>
</tr>
<tr>
<td>answer_10</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714</td>
<td>-0.714***</td>
</tr>
<tr>
<td>constant</td>
<td>0.649***</td>
<td>0.649***</td>
<td>0.612*</td>
<td>0.612***</td>
<td>0.649***</td>
<td>0.649***</td>
<td>0.612</td>
<td>0.612***</td>
</tr>
<tr>
<td>R2</td>
<td>0.01%</td>
<td>0.01%</td>
<td>12.20%</td>
<td>12.20%</td>
<td>38.69%</td>
<td>38.69%</td>
<td>41.88%</td>
<td>41.88%</td>
</tr>
<tr>
<td>n. observations</td>
<td>630</td>
<td>6300</td>
<td>630</td>
<td>6300</td>
<td>630</td>
<td>6300</td>
<td>630</td>
<td>6300</td>
</tr>
</tbody>
</table>

**Figure 5.14:** Table with the OLS coefficients ($\hat{\beta}$) from the response variable Questions Answered and a number of predictor variables. It is only interpretable the sign of these coefficients. It is also highlighted in these coefficients the significance between each predictor variable and the variable Questions Answered by the calculation of the $t$-score.

The value $t_{score} = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$
number of answered questions increase. We can also see that the likelihood to answer the rest of the questions with respect to question 1 is lower. Question 8 looks to be the lowest likelihood with respect to question 1 (see cells yellow highlighted in Figure 5.14). It is also remarkable that the $R^2$ increases when we introduce more variables.

Course

In the table of Figure 5.15, you can see all the predictor variables and their values of the OLS regression coefficients. Based on the results in Table 5.15, we can add that the $4HAVO$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>members</td>
<td>0.010</td>
<td>0.010</td>
<td>0.000</td>
<td>0.000</td>
<td>0.010</td>
<td>0.010</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Course</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1GYM</td>
<td>0.048</td>
<td>0.048**</td>
<td>-0.138</td>
<td>-0.138***</td>
<td>0.046</td>
<td>0.048***</td>
<td>-0.138</td>
<td>-0.138***</td>
</tr>
<tr>
<td>4HAVO</td>
<td>-0.209***</td>
<td>-0.209***</td>
<td>-0.219***</td>
<td>-0.219***</td>
<td>-0.209***</td>
<td>-0.209***</td>
<td>-0.219***</td>
<td>-0.219***</td>
</tr>
<tr>
<td>4WBO</td>
<td>0.168***</td>
<td>0.168***</td>
<td>0.009</td>
<td>0.000</td>
<td>0.168***</td>
<td>0.168***</td>
<td>0.009</td>
<td>0.000</td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-2-3</td>
<td>0.198</td>
<td>0.198***</td>
<td>0.198</td>
<td>0.198***</td>
<td>0.198*</td>
<td>0.198***</td>
<td>0.198*</td>
<td>0.198***</td>
</tr>
<tr>
<td>2-3-1</td>
<td>0.235***</td>
<td>0.235***</td>
<td>0.202***</td>
<td>0.239***</td>
<td>0.239***</td>
<td>0.239***</td>
<td>0.239***</td>
<td>0.239***</td>
</tr>
<tr>
<td>3-1-2</td>
<td>0.066</td>
<td>0.066</td>
<td>0.092</td>
<td>0.092</td>
<td>0.066</td>
<td>0.066</td>
<td>0.092</td>
<td>0.092</td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game_Cella_Lab</td>
<td>0.366***</td>
<td>0.366***</td>
<td>0.007</td>
<td>0.007</td>
<td>0.366***</td>
<td>0.366***</td>
<td>0.007</td>
<td>0.007</td>
</tr>
<tr>
<td>Intertain_Lab</td>
<td>-0.135***</td>
<td>-0.135***</td>
<td>-0.112***</td>
<td>-0.112***</td>
<td>-0.135***</td>
<td>-0.135***</td>
<td>-0.112***</td>
<td>-0.112***</td>
</tr>
<tr>
<td>MacBook</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>length_experiment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>answer_1</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
</tr>
<tr>
<td>answer_2</td>
<td>-0.095</td>
<td>-0.095***</td>
<td>-0.095</td>
<td>-0.095***</td>
<td>-0.095</td>
<td>-0.095***</td>
<td>-0.095</td>
<td>-0.095***</td>
</tr>
<tr>
<td>answer_3</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
<td>-0.254***</td>
</tr>
<tr>
<td>answer_4</td>
<td>-0.587***</td>
<td>-0.587***</td>
<td>-0.587***</td>
<td>-0.587***</td>
<td>-0.587***</td>
<td>-0.587***</td>
<td>-0.587***</td>
<td>-0.587***</td>
</tr>
<tr>
<td>answer_5</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
<td>-0.317***</td>
</tr>
<tr>
<td>answer_6</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
<td>-0.381***</td>
</tr>
<tr>
<td>answer_7</td>
<td>-0.429***</td>
<td>-0.429***</td>
<td>-0.429***</td>
<td>-0.429***</td>
<td>-0.429***</td>
<td>-0.429***</td>
<td>-0.429***</td>
<td>-0.429***</td>
</tr>
<tr>
<td>answer_8</td>
<td>-0.569***</td>
<td>-0.569***</td>
<td>-0.569***</td>
<td>-0.569***</td>
<td>-0.569***</td>
<td>-0.569***</td>
<td>-0.569***</td>
<td>-0.569***</td>
</tr>
<tr>
<td>answer_9</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
</tr>
<tr>
<td>answer_10</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
<td>-0.714***</td>
</tr>
<tr>
<td>constant</td>
<td>0.231</td>
<td>0.231***</td>
<td>-0.032</td>
<td>-0.032</td>
<td>0.689***</td>
<td>0.689***</td>
<td>0.406**</td>
<td>0.406**</td>
</tr>
<tr>
<td>n observations</td>
<td>630</td>
<td>630</td>
<td>630</td>
<td>630</td>
<td>630</td>
<td>630</td>
<td>630</td>
<td>630</td>
</tr>
<tr>
<td>$R^2$</td>
<td>11.88%</td>
<td>11.88%</td>
<td>12.85%</td>
<td>12.85%</td>
<td>41.56%</td>
<td>41.56%</td>
<td>42.53%</td>
<td>42.53%</td>
</tr>
</tbody>
</table>

$99\%$ significant $***$
$95\%$ significant $**$
$90\%$ significant $*$

$t_{score} = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$

Figure 5.15: Table with the OLS coefficients ($\hat{\beta}$) from the response variable Questions_Answered and a number of predictor variables. It is only interpretable the sign of these coefficients. It is also highlighted in these coefficients the significance between each predictor variable and the variable Questions_Answered by the calculation of the $t_{score}$.

course is lower in answering questions than the course $5VWO$. Doing the experiment at the Game Cella Lab increased the number of answered questions with respect to the students who used the MacBook in the Intertain Lab. Doing the experiment at the Intertain Lab decreases the number of answered questions with respect to using the MacBook (see cells in yellow highlighted in Figure 5.14). With the variable course the $R^2$ is a little higher than with the age variable. Therefore, this model explains the variance of the response variable better, Questions_Answered, by the predictor variables used in the model.

5.3.2 Model Correct Questions

In this model, we used as response variable $y = \text{Questions\_Correct}$. We have one model with the age predictor variable and other with the course predictor variable. We have in
total 334 observations. We also expand the sample by 10 (3340 observations). In Tables 5.16 and 5.17 we show the results obtained for the Questions Answered Correctly model. In the table 5.16 we introduce age as dependent variable and in table 5.17 this variable is replaced by course. It is not possible to introduce this two variables jointly because they show multicollinearity.

**Age**

In the table of Table 5.16, you can see all the predictor variables and their values of the the OLS regression coefficients. Based on the results in Table 5.16, the variable length \_experiment is always positive and 99% significant. Because this variable is measured in seconds, the coefficient is always $0.000***$. Of course, we can say that with an increase of time the number of correct answered questions increases. The likelihood to answer the questions 5,8 and 10 correctly with respect to question 1 is lower. Question 8 is the lowest likelihood with respect to question 1.

**Course**

In Table 5.17, you can see all the predictor variables and their values of the the OLS regression coefficients. Based on the results in Table 5.17, we can add that the increase of members decreases the number of correct answered questions. The number of questions answered correctly are significantly lower for the courses 1GYM and 4VWO with respect
5.3. STATISTIC MODEL

CHAPTER 5. RESULTS

Figure 5.17: Table with the OLS coefficients ($\hat{\beta}$) from the response variable Questions_Correct and a number of predictor variables. It is only interpretable the sign of these coefficients. It is also highlighted in these coefficients the significance between each predictor variable and the variable Questions_Correct by the calculation of the $t_{score}$.

<table>
<thead>
<tr>
<th>Predictor Variables (X)</th>
<th>Conf (1α)</th>
<th>Conf (3α)</th>
<th>Conf (2α)</th>
<th>Conf (3β)</th>
<th>Conf (3α)</th>
<th>Conf (4α)</th>
<th>Conf (4β)</th>
</tr>
</thead>
<tbody>
<tr>
<td>members</td>
<td>-0.150**</td>
<td>-0.152***</td>
<td>-0.153***</td>
<td>-0.150***</td>
<td>-0.148**</td>
<td>-0.148**</td>
<td>-0.142**</td>
</tr>
<tr>
<td>1GVM</td>
<td>-0.113***</td>
<td>-0.113***</td>
<td>-0.195***</td>
<td>-0.189**</td>
<td>-0.145**</td>
<td>-0.145**</td>
<td>-0.212**</td>
</tr>
<tr>
<td>4HAVO</td>
<td>0.067</td>
<td>0.067</td>
<td>0.049</td>
<td>0.049**</td>
<td>0.039</td>
<td>0.039</td>
<td>0.029</td>
</tr>
<tr>
<td>4VWO</td>
<td>-0.250***</td>
<td>-0.250***</td>
<td>-0.305***</td>
<td>-0.305***</td>
<td>-0.234***</td>
<td>-0.234***</td>
<td>-0.303***</td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-3</td>
<td>0.092</td>
<td>0.092</td>
<td>0.088</td>
<td>0.088</td>
<td>0.057</td>
<td>0.057</td>
<td>0.046</td>
</tr>
<tr>
<td>2-3</td>
<td>0.045</td>
<td>0.045</td>
<td>0.029</td>
<td>0.029</td>
<td>0.036</td>
<td>0.036</td>
<td>0.008</td>
</tr>
<tr>
<td>3-3</td>
<td>0.105</td>
<td>0.105</td>
<td>0.111</td>
<td>0.111</td>
<td>0.099</td>
<td>0.099</td>
<td>0.096</td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game_Cells</td>
<td>-0.409***</td>
<td>-0.409***</td>
<td>-0.557***</td>
<td>-0.557***</td>
<td>-0.444***</td>
<td>-0.444***</td>
<td>-0.560***</td>
</tr>
<tr>
<td>InterCell (lab)</td>
<td>-0.214***</td>
<td>-0.214***</td>
<td>-0.208**</td>
<td>-0.208**</td>
<td>-0.213**</td>
<td>-0.213**</td>
<td>-0.262**</td>
</tr>
<tr>
<td>Macbook</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>length experiment</td>
<td>0.000</td>
<td>0.000**</td>
<td>0.000</td>
<td>0.000**</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>99% significant</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>95% significant</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>90% significant</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

$t_{score} = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$
to the course 5VWO. With the predictor variable course the $R^2$ is higher than with the age variable. Therefore, this model explains better the variance of the response variable, Questions Correct, by the predictor variables used in the model. It is remarkable that the value $R^2$ in this model with the response variable Questions Correct is lower than with the dependant variable Questions Answered. That is because we have less observations (334 against 630).

5.4 Analyse Observations and Feedback

In this section we will analyse the observations we had during all the sessions. Then, we will analyse the feedback we had from the students after each session.

5.4.1 Observations

General Observations

The general observations we made during the experiment were the following:

1. In general the sessions went very well. The first session was an experiment to learn about doing the rest of the sessions. The major issue was the slow performance with the OTOY instances in the cloud. The computers crashed very often, which was annoying for the students. We only had this problem during the three sessions at the Intertain Lab. We saw that a lot of students became frustrated because of this slow performance and that is one of the reasons why some students had a lower motivation. Our experience with that says that if your response is 24 milliseconds slower than the videos that you normally see in a screen, people start to get annoyed and during those sessions it was sometimes much more than 25 milliseconds.

2. The older students get bored less fast but they were frustrated much faster. The students of 12 years old were distracted quite fast, but they were less faster frustrated.

3. A few students will actually happily scroll through a table view with more than 500 rows until they find the node(s) they were looking for, instead of searching or drilling down. Actually they tried to search in the table view, but they did not know how. Another problem with the table view was that they were confused when you cannot see all the columns attributes together.

4. Almost all the students understood the drill-down feature to zoom in a specific part.

5. Although we explained it in the presentation, many students asked about how to drill up.

6. Some students had problems selecting something in the ring.

7. Expand links to nodes was more difficult. They got stuck when they needed to expand links to nodes to select a drill down selection of nodes in the inner ring (see Figure 5.18). Many students asked how to do that.

8. Some students were often confused after expanding internal links between two categories. They expected to see only the links between the 2 (or more) sets of nodes they had in their selection, but you also see links within the node sets. It is formally correct, but they thought it was an error because the Bundling View (see Figure 5.19) looks confusing. Once you said that they only had to select again the category and expand again to incoming or outgoing links they understood it.
5.4. ANALYSE OBSERVATIONS AND FEEDBACK

CHAPTER 5. RESULTS

Figure 5.18: Expand links to nodes.

Figure 5.19: Expanding to internal links between countries and musical artists.
9. Adjust selection table view, nodes/links and "Selection", "All" and "Highlight" (see Figure 5.20) was another problem for the students. We found out this problem in other visualizations like the scatter plot (see page 14) or in the selection of the correct attribute in search and filter. Sometimes they searched by the default attribute and they did not know you can search by nodes or by links. Actually it was impressing that the 12 years old group had less problems in this matter. This problem affected at the moment to visualize in the table view the current search made with the search and filter visualization (see page 13).

Figure 5.20: Adjust Selection Table View (nodes/links and "Selection", "All" and "Highlight").

10. Some students had difficulties changing tiles size to see bigger one visualization (see Figure 5.21).

11. They asked a lot of times how to open a new view, it was clicking the bottom New in one of the tiles, see figure 5.22.

12. It is also remarkable that many students did not know so clear the difference between incoming and outgoing links.

13. How to relate and interact with two visualizations was not clear and new for the students.

Technical Problems

During the observations we found out some technical problems:

1. In the first session the OTOY was not possible for more than 4 instances due to lack of IP numbers.

2. In the second session OTOY worked with 9 instances, but it was not really stable. A few instances are fast and the rest quite slow.

3. In Game Cella lab where we were using a local version installed in the computers, the drill-up sometimes was fast and sometimes slow.
Figure 5.21: Change tiles size.

Figure 5.22: Open a New View bottom.
5.4.2 Feedback

After the evaluation test we started a conversation with the students (see page: 34). We asked all the students what they thought about Synerscope. In the number list below I show the most interesting comments from the students.

Opinion Students

1. The most of students were surprised about the visualization. They thought it was very interesting to see how the information is linked in Wikipedia. Some students felt how efficient the visualization was, but they missed a faster and handier way to navigate between the categories. They also thought that it was weird that a program like this was not invented before.

2. Many students thought that it was quite complicated in the beginning but after a while it was more fun and interesting and they became faster. Some students appreciated all the difficult procedures behind the program. They could see how complex the things are.

3. Some students thought it was not clear that they could not see all the data set, which is not possible because there are around 100.000 nodes and 250.000 links between them in the data set. Actually SynerScope is solving that, it enables to visualize ”Big Data” and drill-down and focus on specific parts of the data in isolation to reveal and overview level to reveal patterns, trends and outliers, but some students did not appreciate it.

4. Some students thought that the questions were very specific for a wide data set, but that is actually what we wanted to do, to be able to search something specific in a big data set.

5. One student asked why the program did not have just a line to represent the connections. He thought it was so inefficient to have a ”knot” with many links together. We responded him that if you only have one line per connection the distinction would be gone and he totally understood.

SynerScope Improvements and Better Explanation

6. Some students said that if they had to search for something very detailed, you need another function and they did not know where that function was. Thus, they thought you needed to know the program and it has to be clearer what each bottom does. Some students thought it would be interesting to hide some bottoms with different modes like one ”advanced mode”, ”normal mode”, etc. and you do not get distracted by all the other ones. It would be a good idea to change some buttons to make it clearer.

7. Some students also thought that it was very handy that you have the web view. It was more recognisable, but they missed to see the web page selecting the nodes and not only the links. This matter was already in the to-do list, but we did not have time to arrange it for this experiment.

8. Some students thought that it would be useful to be able to zoom straight in the ring and not only in the connections. So if you select any node in the ring you can drill down without selecting its links. SynerScope is already trying to fix this problem.
9. Many students thought that it was annoying that if you searched something and it was not written like in Wikipedia you did not get any result. They would like to have a search engine more efficient and when you are wrong it corrects you.

10. Some students said that it would be nice after searching something to get it automatically in the Table View.

11. Some students thought it was a problem to do not see all the columns in the table view. It took so much time to use it.

12. A lot of students did not like that the drill up operation took so long what it is normal because it has to load 200,000 links, but even in the Game Cella it was weird that sometimes was fast and other times not.

13. Many students from different sessions missed to have a partial drill up. So if for example you do three drill downs and you want to go backward one drill up or two steps and not only general overview again.

14. It was confused for many students that if you zoom in you can just search in that part and not in others, they thought it had to be more clear. So if you have the general part always, you can always search faster.

15. Some students thought it was not possible to search for a value of one attribute in the table view and it is actually possible clicking the right button of the mouse. (see figure 5.23).

![Figure 5.23: Searching for a value in Table View.](image)

**Other Comments**

16. Some students were surprised because there were so many columns empty from the data of Wikipedia, that it is problem of Wikipedia and we have to make it better.

17. Some students were interesting about who is using SynerScope.
5.5 Answer Sub questions

After the quantitative and qualitative analysis from the tests, the observations and the feedback of the students, we are going to summarize them answering the sub questions.

5.5.1 What is the opinion of the students about SynerScope?

The most of the students thought it was really interesting to try a program like SynerScope and visualize Wikipedia in a way they never saw before. So many students did not understand the program very well in the beginning but once they understood, they became faster and faster. That means that there was a learning process during the session. You can also see that in the time they used answering each question. For the first questions they took longer than the last questions. After using the program they became handier and faster with SynerScope. This conclusion is based on the feedback observations 1,2 and 3 and the measurements shown in Figure 5.8 and 5.7.

Way of interacting with SynerScope.

One general thought from the students was that it is not clear because they could not find everything immediately. They would like to have all the data that is in the ring in a row so it shows automatically where you can search on. Thus, they have to appreciate that what they are looking on is ”Big Data” and they can not find a specific information immediately with the rows of data. This conclusion is based on the feedback observation 4.

Way of organising SynerScope.

The most of the students were surprised about the view of the program and the first reaction was that it looked very difficult, but it was because they never saw a visualization like that. It was difficult for some students to change the size of the tiles in the Interface to zoom in one visualization. This conclusion is based on the feedback observation 10.

Difficulties with SynerScope.

Below you have a list with the most repeated difficulties from all the sessions:

- Difference ring selection.
- Expanding links to nodes to select a drill down selection of nodes.
- Adjust Selection Table View (see Figure 5.20), and the settings from the different visualizations.
- Some students did not know how to see the selection nodes searched in the ”Search and filter” in the Bundling View. They can not appreciate them when they are in the drill up visualization of the bundling view and they did not think about expand nodes to links to make the selection.
- Students often got confused after expanding internal links between two categories of nodes.
5.5. ANSWER SUB QUESTIONS

CHAPTER 5. RESULTS

Suggested Changes SynerScope.

Below you have a list with the most repeated suggested changes from the students for SynerScope in all the sessions:

- Zoom in straight on the ring of the bunling view.
- Drill up faster.
- Partial drill up.
- Have two HEB views simultaneously that act independently of each other, so you can look at multiple drill levels side-by-side.
- Articles should open in the Web view only when selecting an actual node and not only a link.
- Hide some of the functions in an advanced panel and only show the most elementary buttons.

5.5.2 What are the most chosen strategies to search for the answer? Why?

The most of the mistakes answering the questions were due to use wrong one button and because they wanted to answer as fast as possible. This conclusion is based on the comments from Figure 5.5.

The most of the students and what everybody does is to follow the intuition. We have all so much experience searching with ordinal search engines as Google. In a ordinal search engine you can search something typing a query in the Google search box and depending on the terms you used, you can be closer or further to your information need. When we have a better technique and we want to teach using it, one of the major problems is to forget what it is intuitive.

Thus, very often the students were just trying to search with the Search and Filter view like it was Google. They did not pay that much attention to the most powerful visualization, the Bundling view and the advantages to have so many different visualizations interacting with one another.

Difference of time between the questions.

The students took on average 7 minutes to answer one question. We realized that the high school students wanted to search something and answer the question as fast as possible. They did not wonder why and when it was not working, they just went on. This validates that the lower your education, the faster you are with the program.

In general the most difficult questions took more time to answer than the easier questions. This conclusion is based on Figure 5.7.

5.5.3 Are the students’ capacities an age important for using SynerScope?

The results say that it is possible after an explanation of the program in 35 minutes that all the students from different age, school course, level, genre and motivation can use Sy-
erScope and answer at least one question correctly. This conclusion is based on Figure 5.9.

We demonstrated that older students can answer more questions. Naturally, because they have more experience. This conclusion is based on table from Figure 5.14.
Chapter 6

Conclusion

After all the results obtained from the Chapter 5, some important conclusions should be remarked that will answer the main research question proposed in the first chapter. There have been six sessions with students between 12 and 18 years old from three different high-schools in Amsterdam. Thus, settling on the obtained results exhaustively some ideas can be figured out.

How can we make and explain SynerScope easier to use for people who are less experienced with computers?

With this experiment we proved that high-school students between 12 and 18 years old are able to use SynerScope, after a short explanation. All the students have the course informatics at school, but none of them ever worked with a program like SynerScope. Our participants were students from different ages and different levels. We could see that experience comes with the age. The older the students were, the more questions were answered. Those students had more experience working with computers, because they had the course informatics longer than the younger students. Anyhow, all the students were able to work with the program. Concluding: it is an advantage to be more experienced with computers, but it is not indispensable.

SynerScope was quite complicated to work with for our participants. There are some possible improvements to make it easier to work with:

1. Zoom straight in the ring and not only in the connections from the bundling view.
2. Auto-Correcting search engine.
3. Search automatically table view.
4. Partial drill-up in the bundling view.
5. Possibility to always have the general overview.

With the 40 minutes explanation we gave, the students were already able to work with the program. Observing the students during the sessions, we found out that they missed some part of explanation. To improve the explanation we should explain the following parts clearer:

1. SynerScope is not about searching like we do in conventional search engines as Google. Moreover, the bundling view is not as intuitive as it could be, it requires further explanations of each features. Thus, the users can easily navigate and interact with SynerScope.
2. The students had a lot of problems understanding how to use the *In-Coming* and *Out-Going* buttons. This matter has to be explained clearer, otherwise, the results are not appropriate.

3. Another usual issue was to use the button *Expand links to Nodes*. If you do not know how to use that button, you can not visualize a drill-down selection of nodes.

4. Adjust selection *Table View* and select properly settings in all the visualizations.

THE FUTURE OF BIG DATA IS HERE AND EVERYBODY CAN LOOK INSIDE OF IT WITH SYNERSCOPE.
Chapter 7

Discussion

In general the experiments went very well. We got the results we wanted to have. Young students are able to use SynerScope and we discovered many improvements in explaining the program and make it easier to use.

The next time we would do this experiment, we should change some procedures:

1. The performance of SynerScope was much better in the Game Cella Lab computers in the VU. Because of the problems with the OTOY instances in the Intertain Lab, the students became frustrated very fast. It would be nice to avoid that next time and always do the sessions at the Game Cella.

2. We did one session with five students working individually with a computer at the Game Cella lab. During this session the students were much more motivated and concentrated.

3. We found out that high school students do not like to put much effort in something they have to do for school. Some of the groups turned out to be annoyed or unmotivated. The next time we can add one kind of challenge. For example: the students who can interact the best with SynerScope, deserved a price. Or the teacher of informatics could give a mark on the work they did.

4. With more students, we would have had more observations and the statistic analyse could be more robust.
Chapter 8

Future Work

This experiment was only the first step. Things that would be interesting to investigate in the future would be:

1. Test the same groups of students again and make learning effect study.
2. Use lower level high school students (MAVO).
3. Use other groups, like university students from Computer Science and from totally different studies like languages, law, etc.
Appendix A

Appendix Title
Appendix A

Motivation Letter School

Dear Mr. ..., 

My name is Fernando Nuñez Serrano and I am doing an internship at the Dutch company Synerscope. We would like to do an experiment with the Synerscope program, software for visually analysing massive amounts of data, together with students from your school.

I got your contact information from ...

Our software is the latest in visual analytics software that can show networks of hundreds of thousands of relations, for example, parts of the World Wide Web. It allows people to interact with data by pointing and clicking. Usually, you would use your Web browser to look at one Website at a time and to click on links to go to another page. With our software you can also look at and interact with the shape of the entire network of links between the pages. This allows you to see which groups of pages link to each other and which don’t, and select pages to zoom in on with a regular browser.

Most people, even analysts and data scientists, have never seen software like this before. Our goal with the experiment is to find the best way to explain the program. Some of our clients have been working with old analysis software for a long time and are apprehensive to try out our new visual approach. If we can show that we can explain it to high school students that might take away some of their insecurity. We also want to see where the students have difficulties in understanding the program so that we can make it easier to understand.

Concretely, the students from your course Informatica could learn what all of Wikipedia looks like from a bird’s eye view and how they can visualise, analyse and understand Wikipedia in our program. We think it could be a nice opportunity for the students to get to know what the latest computer software outside of the school curriculum looks like, and it would give them a totally different perspective on Wikipedia, which they are probably already familiar with.
If we could do the experiment with one or two of your fourth or fifth year classes, we would be very grateful. Please let me know if you are interested by mail or telephone and we can arrange a meeting to discuss the experiment and how we could make it work in practice.

I am looking forward to hearing from you.
Thank you in advance for your help.

Sincerely,

Fernando Núñez Serrano
+31 681665220
fenunser@gmail.com

http://www.synerscope.com
Appendix B

Evaluation test and Hand-out

SynerScope Experiment

Date:

School:

Class:

Name Person 1:

Age Person 1:

Name Person 2:

Age Person 2:

Computer Name:
Soccer Players Explanation

This dataset of Wikipedia is a selection of the Soccer Player’s pages and soccer manager’s pages from: AFC Ajax, F.C Barcelona, Manchester United FC and Manchester City FC.
In addition to these pages, the dataset also includes pages from which is being linked to

![Diagram Links Selection](image)

**Figure B.1**: Diagram Links Selection.

the selected pages or we link to. Furthermore, the dataset contains all the links between all the pages listed (see figure B.1).
<table>
<thead>
<tr>
<th>Column</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Wikipedia Title.</td>
<td>Ronald Koeman.</td>
</tr>
<tr>
<td>Team</td>
<td>List of all the teams where the player is been playing. (in alphabetical order)</td>
<td>AFC Ajax/FC Barcelona/FC Groningen/Feyenoord/Netherlands national football team/PSV Eindhoven</td>
</tr>
<tr>
<td>Position</td>
<td>The place where the player plays in the field.</td>
<td>Defender / Midfielder</td>
</tr>
<tr>
<td>Birth Date</td>
<td>YYY-MM-DD</td>
<td>1963-03-21</td>
</tr>
<tr>
<td>Birth Place</td>
<td>Place where the person was born.</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>meters</td>
<td>1.83</td>
</tr>
<tr>
<td>Weight</td>
<td>Kg.</td>
<td></td>
</tr>
<tr>
<td>Death Date</td>
<td>YYY-MM-DD</td>
<td></td>
</tr>
<tr>
<td>Death Place</td>
<td>Place where the person died.</td>
<td></td>
</tr>
<tr>
<td>Hierarchy one</td>
<td>First Hierarchy</td>
<td>Person</td>
</tr>
<tr>
<td>Hierarchy two</td>
<td>Second Hierarchy</td>
<td>Soccer Manager</td>
</tr>
<tr>
<td>Hierarchy three</td>
<td>Third Hierarchy</td>
<td>misc</td>
</tr>
<tr>
<td>Length</td>
<td>Number of works in the Wikipedia article (integer).</td>
<td>1383</td>
</tr>
<tr>
<td>Unique</td>
<td>Number of unique words in the Wikipedia article (integer).</td>
<td>529</td>
</tr>
<tr>
<td>In degree</td>
<td>Number of in LINKS (integer).</td>
<td>57</td>
</tr>
<tr>
<td>Out degree</td>
<td>Number of out LINKS (integer).</td>
<td>98</td>
</tr>
</tbody>
</table>

**Look out!** The hierarchy SoccerPlayer is assigned to the persons that are still soccer players at this moment or soccer player was their last job. Other soccer players from the past have assigned other hierarchy from their current/last status. So many of the ex-soccer players were/are soccer managers too. For instance, Diego Maradona’s hierarchy is: Person-SoccerManager-misc, but he was a soccer player at FC Barcelona.
Soccer Players Questions

1. Which SoccerPlayer/ex-SoccerPlayer from AFC Ajax is the youngest?

   Time:

   Answer:

2. How many ex-soccer players from AFC Ajax died in Paramaribo Suriname? Why?

   Time:

   Answer:

3. Relation the scalar attributes Length and Unique using the Scatter Plot. Which hierarchy Athlete or Artist has more unique terms in its pages?

   Time:

   Answer:

4. Make one question yourself about something interesting that you can find in this dataset.

   Question:

   Time:

   Answer:
Scientists Explanation

This dataset of Wikipedia is a selection of the Scientists’ pages from the field: Physics. In addition to these pages, the dataset also includes pages from which is being linked to the selected pages or we link to. Furthermore, the dataset contains all the links between all the pages listed (see figure B.2).

Figure B.2: Diagram Links Selection.
<table>
<thead>
<tr>
<th>Column</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Wikipedia Title.</td>
<td>Albert Einstein.</td>
</tr>
<tr>
<td>Field</td>
<td>Field in Science.</td>
<td>Physics</td>
</tr>
<tr>
<td>Birth Date</td>
<td>YYYY-MM-DD</td>
<td>1879-03-14</td>
</tr>
<tr>
<td>Birth Place</td>
<td>Place where the person was born.</td>
<td>German Empire/Kingdom of Wurttemberg/Ulm</td>
</tr>
<tr>
<td>Residence</td>
<td>List of residence places.(in alphabetical order)</td>
<td>Switzerland/United States</td>
</tr>
<tr>
<td>Death Date</td>
<td>YYYY-MM-DD</td>
<td>1955-04-18</td>
</tr>
<tr>
<td>Death Place</td>
<td>Place where the person died.</td>
<td>Princeton New Jersey/United States</td>
</tr>
<tr>
<td>Hierarchy one</td>
<td>First Hierarchy</td>
<td>Person</td>
</tr>
<tr>
<td>Hierarchy two</td>
<td>Second Hierarchy</td>
<td>Scientist</td>
</tr>
<tr>
<td>Hierarchy three</td>
<td>Third Hierarchy</td>
<td>misc</td>
</tr>
<tr>
<td>Length</td>
<td>Number of works in the Wikipedia article (integer).</td>
<td>7711</td>
</tr>
<tr>
<td>Unique</td>
<td>Number of unique words in the Wikipedia article (integer).</td>
<td>2166</td>
</tr>
<tr>
<td>In degree</td>
<td>Number of in LINKS (integer).</td>
<td>1017</td>
</tr>
<tr>
<td>Out degree</td>
<td>Number of out LINKS (integer).</td>
<td>309</td>
</tr>
<tr>
<td>uri</td>
<td>Uniform Resource Identifier</td>
<td>\textit{<a href="http://en.wikipedia.org/wiki/AlbertEinstein%7D">http://en.wikipedia.org/wiki/AlbertEinstein}</a></td>
</tr>
</tbody>
</table>
Scientists Questions

5. List the three countries with the most links from scientists:

Time:

Answer:

6. Which disease has the most links from scientists?

Time:

Answer:

7. Make one question yourself about something interesting that you can find in this dataset.

Question:

Time:

Answer:
Musical Artists Explanation

This dataset of wikipedia is a selection of the musician’s pages from the Pop music genre. In addition to these pages, the dataset also includes pages from which is being linked to the selected pages or we link to. Furthermore, the dataset contains all the links between all the pages listed (see figure B.3).

Figure B.3: Diagram Links Selection.
<table>
<thead>
<tr>
<th>Column</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Wikipedia Title.</td>
<td>Michael Jackson.</td>
</tr>
<tr>
<td>Occupation</td>
<td>Activities, jobs. (in alphabetical order)</td>
<td>Actor/Arrangement/Businessperson/Choreography/Dance/Entertainment/Musician/Record producer/Singer-songwriter.</td>
</tr>
<tr>
<td>Associated Band</td>
<td>Bands and associated bands from the musical artist. (in alphabetical order)</td>
<td>The Jackson 5</td>
</tr>
<tr>
<td>Birth Date</td>
<td>YYY-MM-DD</td>
<td></td>
</tr>
<tr>
<td>Birth Place</td>
<td>Place where the person was born.</td>
<td>Gary Indiana.</td>
</tr>
<tr>
<td>Death Date</td>
<td>YYY-MM-DD</td>
<td></td>
</tr>
<tr>
<td>Death Place</td>
<td>Place where the person died.</td>
<td></td>
</tr>
<tr>
<td>Hierarchy one</td>
<td>First Hierarchy</td>
<td>Person</td>
</tr>
<tr>
<td>Hierarchy two</td>
<td>Second Hierarchy</td>
<td>Artist</td>
</tr>
<tr>
<td>Hierarchy three</td>
<td>Third Hierarchy</td>
<td>MusicalArtist</td>
</tr>
<tr>
<td>Length</td>
<td>Number of works in the Wikipedia article (integer).</td>
<td>15205</td>
</tr>
<tr>
<td>Unique</td>
<td>Number of unique words in the Wikipedia article (integer).</td>
<td>3201</td>
</tr>
<tr>
<td>In degree</td>
<td>Number of in LINKS (integer).</td>
<td>828</td>
</tr>
<tr>
<td>Out degree</td>
<td>Number of out LINKS (integer).</td>
<td>586</td>
</tr>
<tr>
<td>uri</td>
<td>Uniform Resource Identifier</td>
<td><a href="http://en.wikipedia.org/wiki/Michael">http://en.wikipedia.org/wiki/Michael</a> Jackson</td>
</tr>
</tbody>
</table>
Musical Artists Questions

8. Which musician from the Netherlands has two links from two ex-Beatles musicians?

Time:

Answer:

9. Which software VideoGame has the most links from Musical Artists?

Time:

Answer:

10. Make one question yourself about something interesting that you can find in this dataset.

Question:

Time:

Answer:
Appendix C

Solutions Test and Steps Per Question

Soccer Players Questions

1. Which SoccerPlayer/ex- SoccerPlayer from AFC Ajax is the youngest?
Steps:
   1. Search and filter on links by attribute Title (to:pages) “AFC Ajax” drill down to selection.
   2. Select SoccerPlayer ring and expand nodes to links, drill down to selection.
   3. Select all the links and expand links to nodes.
   4. Table view Nodes sort by date, first node should be the answer.

2. How many ex-soccer players from AFC Ajax died in Paramaribo Suriname? Why?
Answer: 4 (or 7), Surinam Airways Flight PY764 air crash in Paramaribo.
Steps:
   1. Search and filter on links by attribute Title (to:pages) “AFC Ajax” drill down to selection.
   2. Select SoccerPlayer ring and expand nodes to links, drill down to selection.
   3. Select all the links and expand links to nodes.
   4. Table view Nodes sort by deathPlace, total 7 nodes.
   5. Check in attribute Team how many players played in Ajax, total 4.

3. Relation the scalar attributes Length and Unique using the Scatter Plot. Which hierarchy Athlete or Artist has more unique terms in its pages?
Answer: Artists.
Steps:
   2. Select category ring Artist.
   3. Highlight the other category ring Athlete.
   4. Choose the more Unique category of pages.
4. Make one question yourself about something interesting that you can find in this dataset.
Answer: Free question.
Steps: As many as the participant wants.

Scientists Questions

5. List the three countries with the most links from scientists:
Answer: United States, 362, Germany, 177, United Kingdom, 115.
Steps:
1. Select Scientists ring and Country ring, expand to internal links, drill down selection.
2. Two Choices.
   (a) Select Scientists and expand nodes to outgoing links, drill down selection.
   (b) Select Country and expand nodes to incoming links, drill down selection.
3. Select the three biggest nodes from the inner ring.

6. Which disease has the most links from scientists?
Answer: Cancer, 15.
Steps:
1. Search and filter on nodes by attribute Hierarchy-one “Disease”, expand to incoming links, drill down to selection.
2. Select Scientists ring and expand nodes to links, drill down selection.
3. Select the biggest node from the inner ring.

7. Make one question yourself about something interesting that you can find in this dataset.
Answer: Free question.
Steps: As many as the participant wants.

Musical Artists Questions

8. Which musician from the Netherlands has two links from two ex-Beatles musicians?
Answer: Eddie Van Halen.
Steps:
1. Select Musical Artist ring, expand nodes to incoming links, drill down selection.
2. Search and filter on nodes by attribute BirthPlace “Netherlands”, expand nodes to links, drill down selection.
3. Search and filter on links by attribute Asociated band (to:pages) “Beatles”.
4. Select the node in the inner ring highlighted.

9. Which software VideoGame has the most links from Musical Artists?
Answer: Gand Thef Auto IV.
Steps:
1. Search and filter on nodes by attribute Hierarchy-three “VideoGame”, expand nodes to links, drill down selection.

2. Select Musical Artist ring, expand nodes to outgoing links, drill down selection.

3. Select the biggest node in the inner ring.

10. Make one question yourself about something interesting that you can find in this dataset.

Answer: Free question.

Steps: As many as the participant wants.
Appendix D

Program code

D.0.4 Apache Pig Scripts
dbpedia_preprocessor.pig

register 'uri.py' using jython as uri

dbpediaLink0 = LOAD 'wikipedia_links_en.ttl.bz2' USING PigStorage(' ') AS
(wiki:chararray, predicate:chararray, db:chararray);
dbpediaLink1 = FILTER dbpediaLink0 BY predicate == '<http://xmlns.com/foaf/0.1/primaryTopic>';
dbpediaLink2 = FOREACH dbpediaLink1 GENERATE
uri.nobrackets(wiki) AS wiki, db;

dbpediatypes0 = LOAD 'instance_types_en.ttl.bz2' USING PigStorage(' ') AS
(instance:chararray, predicate:chararray, class:chararray);
dbpediatypes1 = FILTER dbpediatypes0 BY class != '<http://www.w3.org/2002/07/owl#Thing>';
typesjoin0 = JOIN dbpediaLink2 BY db,
         dbpediatypes1 BY instance;
types = FOREACH typesjoin0 GENERATE
$0 AS uri, $4 AS type;
STORE types INTO 'dbpedia_types' using PigStorage('	');

types_group = GROUP types BY uri;
hierarchy1 = FOREACH types_group GENERATE
FLATTEN(types.type) AS type;
hierarchy1_group = GROUP hierarchy1 BY type;
type_count = FOREACH hierarchy1_group GENERATE
group AS type, COUNT($1) AS type_count;
ordered_count = ORDER type_count BY type_count;
STORE ordered_count INTO 'dbpedia_type_count' using PigStorage('	');

linkgraph.pig

REGISTER 'pignlproc-0.1.0-SNAPSHOT.jar';
%default INPUT '/mnt/hgfs/share/sample';  
%default OUTPUT 'wikipedia_links';

parsed = LOAD '$INPUT' USING pignlpstorage.ParsingWikipediaLoader('en')  AS (title, uri, text, redirect, links, headers, paragraphs);
links0 = FOREACH parsed GENERATE uri, FLATTEN(links);
links = FOREACH links0 GENERATE uri AS from, links::target AS to;
STORE links INTO '$OUTPUT' USING PigStorage('\t');

one_hop_plus_inter.pig

%default SELECTION 'filter_Athlete';  
%default LINKS 'wikipedia_links';  
%default INOUT 'inout_Athlete';  
%default INTER 'inter_Athlete';

selection = LOAD '$SELECTION' using PigStorage('\t')  AS (uri:chararray);

links = LOAD '$LINKS' using PigStorage('\t')  AS (from:chararray, to:chararray);

# 1 hop expand
selection_in_links0 = JOIN selection BY uri, links BY to;
selection_in_links = FOREACH selection_in_links0 GENERATE  
    links::from AS from, links::to AS to;
selection_out_links0 = JOIN selection BY uri, links BY from;
selection_out_links = FOREACH selection_out_links0 GENERATE  
    links::from AS from, links::to AS to;
selection_in_out_links0 = UNION  
    selection_in_links, selection_out_links;
selection_in_out_links = DISTINCT selection_in_out_links0;
STORE selection_in_out_links INTO '$INOUT' USING PigStorage('\t');

sources = FOREACH selection_in_out_links  
    GENERATE from AS uri;
targets = FOREACH selection_in_out_links  
    GENERATE to AS uri;
nodes0 = UNION sources, targets;
nodes = DISTINCT nodes0;

# interrelations
constrained_by_in_links0 = JOIN nodes BY uri, links BY to;
constrained_by_in_links = FOREACH constrained_by_in_links0  
    GENERATE links::from AS from, links::to AS to;

81
APPENDIX D. PROGRAM CODE

constrained_by_out_links0 = JOIN nodes BY uri,
constrained_by_in_links BY from;
constrained_by_out_links = FOREACH constrained_by_out_links0
GENERATE constrained_by_in_links::from AS from,
constrained_by_in_links::to AS to;

-- also contains 1 hop expand links
interrelation_links = DISTINCT constrained_by_out_links;

STORE interrelation_links INTO '$INTER' USING PigStorage('\t');

one_hop_augment.pig

%default SELECTION 'inter_Athlete';
%default LINKS 'wikipedia_links';
%default AUGMENTED 'inter_aug_Athlete';
%default SCOOP '50000';

link_selection = LOAD '$SELECTION' using PigStorage('\t')
AS (from:chararray, to:chararray);

links = LOAD '$LINKS' using PigStorage('\t')
AS (from:chararray, to:chararray);

from_nodes = FOREACH link_selection
GENERATE from AS uri;
to_nodes = FOREACH link_selection
GENERATE to AS uri;
nodes = UNION from_nodes, to_nodes;
selection = DISTINCT nodes;

-- 1 hop expand
selection_in_links0 = JOIN selection BY uri, links BY to;
selection_in_links = FOREACH selection_in_links0
GENERATE links::from AS from, links::to AS to;
selection_out_links0 = JOIN selection BY uri, links BY from;
selection_out_links = FOREACH selection_out_links0
GENERATE links::from AS from, links::to AS to;
selection_in_out_links0 = UNION
selection_in_links, selection_out_links;
selection_in_out_links = DISTINCT selection_in_out_links0;


cogr = COGROUP selection_in_out_links BY (from, to),
link_selection BY (from, to);
minus = FILTER cogr BY IsEmpty(link_selection);
links_non_selection = FOREACH minus
GENERATE FLATTEN(selection_in_out_links);

-- COUNT on link_selection
link_group = GROUP link_selection ALL;
link_selection_n = FOREACH link_group GENERATE
COUNT(link_selection) AS cnt;

-- COUNT on links
links_group = GROUP links_non_selection ALL;
links_non_selection_n = FOREACH links_group GENERATE
COUNT(links_non_selection) AS cnt;

additional_sample = SAMPLE links_non_selection
(((double)"$SCOOP"-(double)link_selection.n.cnt)/
(doubler(link_selection.n.cnt));
augmented_selection = UNION additional_sample, link_selection;

STORE augmented_selection INTO '$AUGMENTED' USING PigStorage('t');

decorate_Soccer.pig

REGISTER 'pignlproc-0.1.0-SNAPSHOT.jar';
REGISTER 'dbpedia_hierarchy.py' using jython AS dbp;
REGISTER 'bag_of_words.py' using jython AS bow;
REGISTER 'autoinc.py' using jython AS autoinc;

%default WIKI '/mnt/hgfs/share/sample';
%default LINKS 'inter_aug_Soccer';
%default NODES 'nodes_Soccer';

link_selection = LOAD '$LINKS' USING PigStorage('t')
AS (from:chararray, to:chararray);

sources = FOREACH link_selection GENERATE from AS uri;
targets = FOREACH link_selection GENERATE to AS uri;
nodeselection0 = UNION sources, targets;
nodeselection = DISTINCT nodeselection0;

parsed = LOAD '$WIKI'
USING pignlproc.storage.ParsingWikipediaLoader('en')
AS (title, uri, text, redirect, links, headers, paragraphs);

nodeselection_parsed0 = JOIN nodeselection BY uri, parsed BY uri;
nodeselection_parsed = FOREACH nodeselection_parsed0 GENERATE
  parsed::title AS title,
  parsed::uri AS uri,
  parsed::text AS text,
  parsed::redirect AS redirect,
  parsed::links AS links,
  parsed::headers AS headers,
  parsed::paragraphs AS paragraphs;

bags = FOREACH nodeselection_parsed GENERATE
  uri, title, bow.bag_of_words(text) AS tfs;
totals = FOREACH bags GENERATE
  uri, title, FLATTEN(bow.totals(tfs)), bow.top(tfs, 10) AS top10;
terminfo = FOREACH totals GENERATE
uri, REPLACE(REPLACE(title,'\n',''),'\t',''))
AS title, $2 AS unique, $3 AS length, top10;

−− In and out degree −−
edgelist0 = FOREACH
node_selection_parsed GENERATE uri, FLATTEN(links);
edgelist1 = FOREACH edgelist0 GENERATE uri, links::target;
out_group = GROUP edgelist1 BY uri;
out_degree = FOREACH out_group GENERATE
$0 AS uri, COUNT($1) AS out_degree;
in_group = GROUP edgelist1 BY links::target;
in_degree = FOREACH in_group GENERATE
$0 AS uri, COUNT($1) AS in_degree;

nodeinfo0 = COGROUP in_degree BY uri, out_degree BY uri;
nodeinfo1 = FOREACH nodeinfo0 GENERATE group AS uri,
SUM(in_degree.in_degree) AS in_degree,
SUM(out_degree.out_degree) AS out_degree;

−− DBpedia hierarchy −−
types = LOAD 'dbpedia_types' using PigStorage('
')
AS (uri:chararray, type:chararray);
types_group = GROUP types BY uri;
hierarchy0 = FOREACH types_group GENERATE
group AS uri, FLATTEN(dbp.hierarchy(types.type));
hierarchy = FOREACH hierarchy0 GENERATE
uri, $1 AS one, $2 AS two, $3 AS three;

−− Extra columns −−
columns = LOAD 'columns' USING PigStorage (',')
AS (uri:chararray, associatedBand:chararray,
associatedMusicalArtist:chararray, birthPlace:chararray,
deathPlace:chararray, height:double, occupation:chararray,
posi t i t i o n: chararray, team:chararray, weight:double);
dates = LOAD 'dates' USING PigStorage (',')
AS (uri:chararray, birthdate:datetime, deathdate:datetime);

nodes0 = COGROUP
terminfo BY uri, nodeinfo1 BY uri, hierarchy BY uri;
nodes1 = FOREACH nodes0 GENERATE
FLATTEN(terminfo.uri) AS uri,
FLATTEN(terminfo.title) AS title,
FLATTEN(terminfo.unique) AS unique,
APPENDIX D. PROGRAM CODE

```pig
FLATTEN(terminfo.length) AS length,
FLATTEN(nodeinfo1.in_degree) AS in_degree,
FLATTEN(nodeinfo1.out_degree) AS out_degree,
FLATTEN(hierarchy.one) AS hierarchy_one,
FLATTEN(hierarchy.two) AS hierarchy_two,
FLATTEN(hierarchy.three) AS hierarchy_three;

join_columns = JOIN nodes1 by uri LEFT, columns BY uri;
nodes2 = FOREACH join_columns GENERATE
    nodes1::uri AS uri,
    nodes1::title AS title,
    nodes1::unique AS unique,
    nodes1::length AS length,
    nodes1::in_degree AS in_degree,
    nodes1::out_degree AS out_degree,
    nodes1::hierarchy_one AS hierarchy_one,
    nodes1::hierarchy_two AS hierarchy_two,
    nodes1::hierarchy_three AS hierarchy_three,
    columns::birthPlace AS birthPlace,
    columns::deathPlace AS deathPlace,
    (double)columns::height AS height,
    columns::occupation AS occupation,
    columns::position AS position,
    columns::team AS team,
    (double)columns::weight AS weight;

/*STORE nodes2 INTO 'soccer_columns' using PigStorage('\t');*/

join_dates = JOIN nodes2 by uri LEFT, dates BY uri;
nodes3 = FOREACH join_dates GENERATE
    nodes2::uri AS uri,
    nodes2::title AS title,
    nodes2::unique AS unique,
    nodes2::length AS length,
    nodes2::in_degree AS in_degree,
    nodes2::out_degree AS out_degree,
    nodes2::hierarchy_one AS hierarchy_one,
    nodes2::hierarchy_two AS hierarchy_two,
    nodes2::hierarchy_three AS hierarchy_three,
    nodes2::birthPlace AS birthPlace,
    nodes2::deathPlace AS deathPlace,
    (double)nodes2::height AS height,
    nodes2::occupation AS occupation,
    nodes2::position AS position,
    nodes2::team AS team,
    (double)nodes2::weight AS weight,
    dates::birthdate AS birthdate,
    dates::deathdate AS deathdate;

nodes4 = DISTINCT nodes3;

nodes = FOREACH nodes4 GENERATE
```
APPENDIX D. PROGRAM CODE

uri, title, unique, length, in_degree, out_degree,

hierarchy_one, hierarchy_two, hierarchy_three,

birthPlace, deathPlace, height, occupation,

position, team, weight, birthdate, deathdate,

autoinc.auto_increment_id() AS sequence_id;

STORE nodes INTO '$NODES' using PigStorage('t ');

decorate_Scientist.pig

REGISTER 'pignlproc −0.1.0−SNAPSHOT.jar ';
REGISTER 'dbpedia_hierarchy.py' using jython AS dbp;
REGISTER 'bag_of_words.py' using jython AS bow;
REGISTER 'autoinc.py' using jython AS autoinc;

%default WIKI '/home/fenunser/PROJECT/Wikipedia_school/sample '/;
%default LINKS 'inter_aug_Soccer ';
%default NODES 'nodes_Soccer ';

link_selection = LOAD '$LINKS' USING PigStorage('t ')

AS (from:chararray, to:chararray);

sources = FOREACH link_selection GENERATE from AS uri;
targets = FOREACH link_selection GENERATE to AS uri;
nodes_selection0 = UNION sources, targets;
nodes_selection = DISTINCT nodes_selection0;

parsed = LOAD '$WIKI'

USING pignlproc.storage.ParsingWikipediaLoader('en')

AS (title, uri, text, redirect, links, headers, paragraphs);

nodes_selection_parsed0 = JOIN nodes_selection BY uri, parsed BY uri;
nodes_selection_parsed = FOREACH nodes_selection_parsed0 GENERATE

parsed::title AS title,
parsed::uri AS uri,
parsed::text AS text,
parsed::redirect AS redirect,
parsed::links AS links,
parsed::headers AS headers,
parsed::paragraphs AS paragraphs;

bags = FOREACH nodes_selection_parsed GENERATE uri, title,

bow.bag_of_words(text) AS tfs;
totals = FOREACH bags GENERATE uri, title,

FLATTEN(bow.totals(tfs)), bow.top(tfs,10) AS top10;
terminfo = FOREACH totals GENERATE uri,

REPLACE(REPLACE(title ,'
', ' '),'	', '') AS title,

$2 AS unique, $3 AS length, top10;

—

— In and out degree
Appendix D. Program Code

__
edgelist0 = FOREACH node_selection_parsed GENERATE uri, FLATTEN(links);
edgelist1 = FOREACH edgelist0 GENERATE uri, links::target;
out_group = GROUP edgelist1 BY uri;
out_degree = FOREACH out_group GENERATE $0 AS uri, COUNT($1) AS out_degree;
in_group = GROUP edgelist1 BY links::target;
in_degree = FOREACH in_group GENERATE $0 AS uri,
COUNT($1) AS in_degree;

nodeinfo0 = COGROUP in_degree BY uri, out_degree BY uri;
nodeinfo1 = FOREACH nodeinfo0 GENERATE group AS uri,
SUM(in_degree.in_degree) AS in_degree,
SUM(out_degree.out_degree) AS out_degree;

__

DBpedia hierarchy
__
types = LOAD 'dbpedia_types' using PigStorage('\t')
AS (uri:chararray, type:chararray);
types_group = GROUP types BY uri;
hierarchy0 = FOREACH types_group GENERATE group
AS uri, FLATTEN(dbp.hierarchy(types.type));
hierarchy = FOREACH hierarchy0 GENERATE uri,
$1 AS one, $2 AS two, $3 AS three;

__

Extra columns
__
columns = LOAD 'columns' USING PigStorage ('',')
AS (uri:chararray, birthPlace:chararray,
deathPlace:chararray, field:chararray, residence:chararray);
dates = LOAD 'dates' USING PigStorage ('',')
AS (uri:chararray, birthdate:datetime, deathdate:datetime);

nodes0 = COGROUP terminfo BY uri, nodeinfo1 BY uri, hierarchy BY uri;
nodes1 = FOREACH nodes0 GENERATE
FLATTEN(terminfo.uri) AS uri,
FLATTEN(terminfo.title) AS title,
FLATTEN(terminfo.unique) AS unique,
FLATTEN(terminfo.length) AS length,
FLATTEN(nodeinfo1.in_degree) AS in_degree,
FLATTEN(nodeinfo1.out_degree) AS out_degree,
FLATTEN(hierarchy.one) AS hierarchy_one,
FLATTEN(hierarchy.two) AS hierarchy_two,
FLATTEN(hierarchy.three) AS hierarchy_three;

join_columns = JOIN nodes1 by uri LEFT , columns BY uri;
nodes2 = FOREACH join_columns GENERATE
nodes1::uri AS uri,
nodes1::title AS title,
nodes1::unique AS unique,
nodes1::length AS length,
nodes1::in_degree AS in_degree,
nodes1::out_degree AS out_degree,
nodes1::hierarchy_one AS hierarchy_one,
nodes1::hierarchy_two AS hierarchy_two,
nodes1::hierarchy_three AS hierarchy_three,
columns::birthPlace AS birthPlace,
columns::deathPlace AS deathPlace,
columns::field AS field,
columns::residence AS residence;

STORE nodes2 INTO 'soccer_columns' using PigStorage('t');

join_dates = JOIN nodes2 by uri LEFT, dates BY uri;
nodes3 = FOREACH join_dates GENERATE
    nodes2::uri AS uri,
nodes2::title AS title,
nodes2::unique AS unique,
nodes2::length AS length,
nodes2::in_degree AS in_degree,
nodes2::out_degree AS out_degree,
nodes2::hierarchy_one AS hierarchy_one,
nodes2::hierarchy_two AS hierarchy_two,
nodes2::hierarchy_three AS hierarchy_three,
nodes2::birthPlace AS birthPlace,
nodes2::deathPlace AS deathPlace,
nodes2::field AS field,
nodes2::residence AS residence,
dates::birthdate AS birthdate,
dates::deathdate AS deathdate;

nodes4 = DISTINCT nodes3;
nodes = FOREACH nodes4 GENERATE
    uri, title, unique, length, in_degree, out_degree, hierarchy_one, hierarchy_two, hierarchy_three, birthPlace, deathPlace, field, residence, birthdate, deathdate,
    autoinc.auto_increment_id() AS sequence_id;

STORE nodes INTO '$NODES' using PigStorage('t');

decorate_MusicalArtist.pig

REGISTER 'pignlproc−0.1.0−SNAPSHOT.jar';
REGISTER 'dbpedia_hierarchy.py' using jython AS dbp;
REGISTER 'bag_of_words.py' using jython AS bow;
REGISTER 'autoinc.py' using jython AS autoinc;

%default WIKI '/mnt/hgfs/share/sample';
%default LINKS 'inter_aug_Soccer';
%default NODES 'nodes_Soccer';
link_selection = LOAD '$LINKS' USING PigStorage('\t')
AS (from:chararray,to:chararray);

sources = FOREACH link_selection GENERATE from AS uri;
targets = FOREACH link_selection GENERATE to AS uri;
node_selection0 = UNION sources, targets;
node_selection = DISTINCT node_selection0;

parsed = LOAD '$WIKI'
USING pignlproc.storage.ParsingWikipediaLoader ('en')
AS (title, uri, text, redirect, links, headers, paragraphs);

node_selection_parsed0 = JOIN node_selection BY uri, parsed BY uri;
node_selection_parsed = FOREACH node_selection_parsed0 GENERATE
    parsed::title AS title,
parsed::uri AS uri,
parsed::text AS text,
parsed::redirect AS redirect,
parsed::links AS links,
parsed::headers AS headers,
parsed::paragraphs AS paragraphs;

bags = FOREACH node_selection_parsed GENERATE
    uri, title, bow.bag_of_words(text) AS tfs;
totals = FOREACH bags GENERATE
    uri, title, FLATTEN(bow.totals(tfs)), bow.top(tfs,10) AS top10;
terminfo = FOREACH totals GENERATE
    uri, REPLACE(REPLACE(title, '\n', ' '), '\t', '') AS title,
$2 AS unique, $3 AS length, top10;

---
--- In and out degree
---
edgelist0 = FOREACH
    node_selection_parsed GENERATE uri, FLATTEN(links);
edgelist1 = FOREACH edgelist0 GENERATE uri, links::target;
out_group = GROUP edgelist1 BY uri;
out_degree = FOREACH out_group GENERATE
    $0 AS uri, COUNT($1) AS out_degree;
in_group = GROUP edgelist1 BY links::target;
in_degree = FOREACH in_group GENERATE
    $0 AS uri, COUNT($1) AS in_degree;

nodeinfo0 = COGROUP in_degree BY uri, out_degree BY uri;
nodeinfo1 = FOREACH nodeinfo0 GENERATE group
    AS uri, SUM(in_degree.in_degree) AS in_degree,
    SUM(out_degree.out_degree) AS out_degree;

---

89
APPENDIX D. PROGRAM CODE

-- DBpedia hierarchy
--
types = LOAD 'dbpedia_types' using PigStorage('\t')
  AS (uri:chararray, type:chararray);
types_group = GROUP types BY uri;
hierarchy0 = FOREACH types_group GENERATE group
  AS uri, FLATTEN(dbp.hierarchy(types.type));
hierarchy = FOREACH hierarchy0 GENERATE
  uri, $1 AS one, $2 AS two, $3 AS three;

-- Extra columns
--
columns = LOAD 'columns_somus' USING PigStorage (',')
  AS (uri:chararray, associatedBand:chararray,
      associatedMusicalArtist:chararray, birthPlace:chararray,
      deathPlace:chararray, height:double, occupation:chararray,
      position:chararray, team:chararray, weight:double);
dates = LOAD 'dates' USING PigStorage ('','
  AS (uri:chararray, birthdate:datetime, deathdate:datetime);

nodes0 = COGROUP
  terminfo BY uri, nodeinfo1 BY uri, hierarchy BY uri;
nodes1 = FOREACH nodes0 GENERATE
    FLATTEN(terminfo.uri) AS uri,
    FLATTEN(terminfo.title) AS title,
    FLATTEN(terminfo.unique) AS unique,
    FLATTEN(terminfo.length) AS length,
    FLATTEN(nodeinfo1.in_degree) AS in_degree,
    FLATTEN(nodeinfo1.out_degree) AS out_degree,
    FLATTEN(hierarchy.one) AS hierarchy_one,
    FLATTEN(hierarchy.two) AS hierarchy_two,
    FLATTEN(hierarchy.three) AS hierarchy_three;

join_columns = JOIN nodes1 by uri LEFT , columns BY uri;
nodes2 = FOREACH join_columns GENERATE
  nodes1::uri AS uri,
  nodes1::title AS title,
  nodes1::unique AS unique,
  nodes1::length AS length,
  nodes1::in_degree AS in_degree,
  nodes1::out_degree AS out_degree,
  nodes1::hierarchy_one AS hierarchy_one,
  nodes1::hierarchy_two AS hierarchy_two,
  nodes1::hierarchy_three AS hierarchy_three,
  columns::associatedBand AS associatedBand,
  columns::associatedMusicalArtist AS associatedMusicalArtist,
  columns::birthPlace AS birthPlace,
  columns::deathPlace AS deathPlace,
  columns::occupation AS occupation;
APPENDIX D. PROGRAM CODE

```class
/*STORE nodes2 INTO 'soccer_columns' using PigStorage('\t');*/

join_dates = JOIN nodes2 by uri LEFT, dates BY uri;
nodes3 = FOREACH join_dates GENERATE
    nodes2::uri AS uri,
    nodes2::title AS title,
    nodes2::unique AS unique,
    nodes2::length AS length,
    nodes2::in_degree AS in_degree,
    nodes2::out_degree AS out_degree,
    nodes2::hierarchy_one AS hierarchy_one,
    nodes2::hierarchy_two AS hierarchy_two,
    nodes2::hierarchy_three AS hierarchy_three,
    nodes2::associatedBand AS associatedBand,
    nodes2::associatedMusicalArtist AS associatedMusicalArtist,
    nodes2::birthPlace AS birthPlace,
    nodes2::deathPlace AS deathPlace,
    nodes2::occupation AS occupation,
    dates::birthdate AS birthdate,
    dates::deathdate AS deathdate;
nodes4 = DISTINCT nodes3;
nodes = FOREACH nodes4 GENERATE
    uri, title, unique, length, in_degree, out_degree,
    hierarchy_one, hierarchy_two, hierarchy_three,
    associatedBand, associatedMusicalArtist, birthPlace,
    deathPlace, occupation, birthdate, deathdate,
    autoinc.auto_increment_id() AS sequence_id;

STORE nodes INTO '$NODES' using PigStorage('\t');

D.0.5 Python UDFs

```

```class
class bag_of_words:
    def __init__(self, string):
        self.string = string

    @outputSchema("bow:bag{t:tuple(term:chararray,freq:long)}")
    def bag_of_words(self):
        return sorted(dict((key, len(list(group)))
            for key, group in
                itertools.groupby([term for term in
                    sorted(re.split("\W+", string.lower())
            if term not in [' ']),
                key=lambda t: t[1], reverse=True))

    @outputSchema("bow:bag{t:tuple(term:chararray,freq:long)}")
    def top(self, bag, n):
        return bag[0:n]

    @outputSchema("t:tuple(unique:long,tokens:long)")
    def most_frequent(self, bag):
        return
```

91
APPENDIX D. PROGRAM CODE

```python
def totals(bag):
    unique_terms = 0
    total_tokens = 0
    for term, freq in bag:
        unique_terms = unique_terms + 1
        total_tokens = total_tokens + freq
    return (unique_terms, total_tokens)

autoinc.py

COUNT = 0

@outputSchema('auto_increment_id:int')
def auto_increment_id():
    global COUNT
    COUNT += 1
    return COUNT

dbpedia_hierarchy.py

@outputSchema("hierarchy:tuple(one:chararray,two:chararray,three:chararray)")
def hierarchy(bag):
    dbp = "http://dbpedia.org/ontology/"
    one = 'misc'
    two = 'misc'
    three = 'misc'
    bag = sorted(bag)
    if (dbp + 'Activity',) in bag:
        one = 'Activity'
    if (dbp + 'Game',) in bag and two == 'misc':
        two = 'Game'
    if (dbp + 'Sport',) in bag and two == 'misc':
        two = 'Sport'
    if (dbp + 'Agent',) in bag:
        one = 'Agent'
    if (dbp + 'Deity',) in bag and two == 'misc':
        two = 'Deity'
    if (dbp + 'Family',) in bag and two == 'misc':
        two = 'Family'
    if (dbp + 'Organisation',) in bag and two == 'misc':
        two = 'Organisation'
    if (dbp + 'Band',) in bag and two == 'Organisation':
        three = 'Band'
    if (dbp + 'ComedyGroup',) in bag and two == 'Organisation':
        three = 'ComedyGroup'
    if (dbp + 'Company',) in bag and two == 'Organisation':
        three = 'Company'
    if (dbp + 'Library',) in bag and two == 'Organisation':
        three = 'Library'
    if (dbp + 'School',) in bag and two == 'Organisation':
        three = 'School'
```

92
if (dbp + 'University>',) in bag and two == 'Organisation':
    three = 'University'
if (dbp + 'MilitaryUnit>',) in bag and two == 'Organisation':
    three = 'MilitaryUnit'
if (dbp + 'PoliticalParty>',) in bag and two == 'Organisation':
    three = 'PoliticalParty'
if (dbp + 'SportsTeam>',) in bag and two == 'Organisation':
    three = 'SportsTeam'
if (dbp + 'SportLeague>',) in bag and two == 'Organisation':
    three = 'SportLeague'
if (dbp + 'TradeUnion>',) in bag and two == 'Organisation':
    three = 'TradeUnion'
if (dbp + 'TermOfOffice>',) in bag and two == 'Organisation':
    three = 'TermOfOffice'

if (dbp + 'Person>',) in bag:
    one = 'Person'
if (dbp + 'Ambassador>',) in bag and two == 'misc':
    two = 'Ambassador'
if (dbp + 'Archeologist>',) in bag and two == 'misc':
    two = 'Archeologist'
if (dbp + 'Architect>',) in bag and two == 'misc':
    two = 'Architect'
if (dbp + 'Aristocrat>',) in bag and two == 'misc':
    two = 'Aristocrat'
if (dbp + 'Artist>',) in bag and two == 'misc':
    two = 'Artist'
if (dbp + 'Actor>',) in bag and two == 'Artist':
    three = 'Actor'
if (dbp + 'Comedian>',) in bag and two == 'Artist':
    three = 'Comedian'
if (dbp + 'ComicsCreator>',) in bag and two == 'Artist':
    three = 'ComicsCreator'
if (dbp + 'Humorist>',) in bag and two == 'Artist':
    three = 'Humorist'
if (dbp + 'MusicalArtist>',) in bag and two == 'Artist':
    three = 'MusicalArtist'
if (dbp + 'Painter>',) in bag and two == 'Artist':
    three = 'Painter'
if (dbp + 'Sculptor>',) in bag and two == 'Artist':
    three = 'Sculptor'
if (dbp + 'Writer>',) in bag and two == 'Artist':
    three = 'Writer'
if (dbp + 'Poet>',) in bag and two == 'misc':
    two = 'Poet'
if (dbp + 'Astronaut>',) in bag and two == 'misc':
    two = 'Astronaut'
if (dbp + 'Athlete>',) in bag and two == 'misc':
    two = 'Athlete'
if (dbp + 'BadmintonPlayer>',) in bag and two == 'Athlete':
    three = 'BadmintonPlayer'
APPENDIX D. PROGRAM CODE

```python
if (dbp + 'AmericanFootballPlayer>') in bag and two == 'Athlete':
    three = 'AmericanFootballPlayer'

if (dbp + 'BaseballPlayer>') in bag and two == 'Athlete':
    three = 'BaseballPlayer'

if (dbp + 'BasketballPlayer>') in bag and two == 'Athlete':
    three = 'BasketballPlayer'

if (dbp + 'Boxer>') in bag and two == 'Athlete':
    three = 'Boxer'

if (dbp + 'BullFighter>') in bag and two == 'Athlete':
    three = 'BullFighter'

if (dbp + 'ChessPlayer>') in bag and two == 'Athlete':
    three = 'ChessPlayer'

if (dbp + 'Cricketer>') in bag and two == 'Athlete':
    three = 'Cricketer'

if (dbp + 'Cyclist>') in bag and two == 'Athlete':
    three = 'Cyclist'

if (dbp + 'DartsPlayer>') in bag and two == 'Athlete':
    three = 'DartsPlayer'

if (dbp + 'FigureSkater>') in bag and two == 'Athlete':
    three = 'FigureSkater'

if (dbp + 'GolfPlayer>') in bag and two == 'Athlete':
    three = 'GolfPlayer'

if (dbp + 'Gymnast>') in bag and two == 'Athlete':
    three = 'Gymnast'

if (dbp + 'HorseRider>') in bag and two == 'Athlete':
    three = 'HorseRider'

if (dbp + 'IcehockeyPlayer>') in bag and two == 'Athlete':
    three = 'IcehockeyPlayer'

if (dbp + 'LacrossePlayer>') in bag and two == 'Athlete':
    three = 'LacrossePlayer'

if (dbp + 'MartialArtist>') in bag and two == 'Athlete':
    three = 'MartialArtist'

if (dbp + 'FormulaOneRacer>') in bag and two == 'Athlete':
    three = 'FormulaOneRacer'

if (dbp + 'MotocycleRacer>') in bag and two == 'Athlete':
    three = 'MotocycleRacer'

if (dbp + 'RallyDriver>') in bag and two == 'Athlete':
    three = 'RallyDriver'

if (dbp + 'PokerPlayer>') in bag and two == 'Athlete':
    three = 'PokerPlayer'

if (dbp + 'RugbyPlayer>') in bag and two == 'Athlete':
    three = 'RugbyPlayer'

if (dbp + 'Skater>') in bag and two == 'Athlete':
    three = 'Skater'

if (dbp + 'SnookerPlayer>') in bag and two == 'Athlete':
    three = 'SnookerPlayer'

if (dbp + 'SoccerPlayer>') in bag and two == 'Athlete':
    three = 'SoccerPlayer'

if (dbp + 'Surfer>') in bag and two == 'Athlete':
    three = 'Surfer'

if (dbp + 'Swimmer>') in bag and two == 'Athlete':
    three = 'Swimmer'
```
three = 'Swimmer'
if (dbp + 'TableTennisPlayer>') in bag and two == 'Athlete':
    three = 'TableTennisPlayer'
if (dbp + 'TennisPlayer>') in bag and two == 'Athlete':
    three = 'TennisPlayer'
if (dbp + 'VolleyballPlayer>') in bag and two == 'Athlete':
    three = 'VolleyballPlayer'
if (dbp + 'Wrestler>') in bag and two == 'Athlete':
    three = 'Wrestler'
if (dbp + 'NascarDriver>') in bag and two == 'Athlete':
    three = 'NascarDriver'
if (dbp + 'Bullfighter>') in bag and two == 'misc':
    two = 'Bullfighter'
if (dbp + 'BusinessPerson>') in bag and two == 'misc':
    two = 'BusinessPerson'
if (dbp + 'Celebrety>') in bag and two == 'misc':
    two = 'Celebrety'
if (dbp + 'Chef>') in bag and two == 'misc':
    two = 'Chef'
if (dbp + 'Cleric>') in bag and two == 'misc':
    two = 'Cleric'
if (dbp + 'Saint>') in bag and two == 'Cleric':
    three = 'Saint'
if (dbp + 'Cardinal>') in bag and two == 'Cleric':
    three = 'Cardinal'
if (dbp + 'Pope>') in bag and two == 'misc':
    two = 'Pope'
if (dbp + 'Coach>') in bag and two == 'misc':
    two = 'Coach'
if (dbp + 'Criminal>') in bag and two == 'misc':
    two = 'Criminal'
if (dbp + 'Economist>') in bag and two == 'misc':
    two = 'Economist'
if (dbp + 'Engineer>') in bag and two == 'misc':
    two = 'Engineer'
if (dbp + 'FictionalCharacter>') in bag and two == 'misc':
    two = 'FictionalCharacter'
if (dbp + 'Historian>') in bag and two == 'misc':
    two = 'Historian'
if (dbp + 'Journalist>') in bag and two == 'misc':
    two = 'Journalist'
if (dbp + 'Judge>') in bag and two == 'misc':
    two = 'Judge'
if (dbp + 'MilitaryPerson>') in bag and two == 'misc':
    two = 'MilitaryPerson'
if (dbp + 'Model>') in bag and two == 'misc':
    two = 'Model'
if (dbp + 'Monarch>') in bag and two == 'misc':
    two = 'Monarch'
if (dbp + 'OfficeHolder>') in bag and two == 'misc':
    two = 'OfficeHolder'
if (dbp + 'OrganisationMember>',) in bag and two == 'misc':
    two = 'OrganisationMember'
if (dbp + 'Philosopher>',) in bag and two == 'misc':
    two = 'Philosopher'
if (dbp + 'Politician>',) in bag and two == 'misc':
    two = 'Politician'
if (dbp + 'Presenter>',) in bag and two == 'misc':
    two = 'Presenter'
if (dbp + 'Psychologist>',) in bag and two == 'misc':
    two = 'Psychologist'
if (dbp + 'Royalty>',) in bag and two == 'misc':
    two = 'Royalty'
if (dbp + 'Scientist>',) in bag and two == 'misc':
    two = 'Scientist'
if (dbp + 'ManagerClub>',) in bag and two == 'misc':
    two = 'ManagerClub'
if (dbp + 'SportsManager>',) in bag and two == 'misc':
    two = 'SportsManager'
if (dbp + 'SoccerManager>',) in bag and two == 'misc':
    two = 'SoccerManager'
if (dbp + 'TelevisionPersonality>',) in bag and two == 'misc':
    two = 'TelevisionPersonality'

if (dbp + 'Altitude>',) in bag:
    one = 'Altitude'

if (dbp + 'AnatomicalStructure>',) in bag:
    one = 'AnatomicalStructure'
if (dbp + 'Artery>',) in bag and two == 'misc':
    two = 'Artery'
if (dbp + 'Bone>',) in bag and two == 'misc':
    two = 'Bone'
if (dbp + 'Brain>',) in bag and two == 'misc':
    two = 'Brain'
if (dbp + 'Embryology>',) in bag and two == 'misc':
    two = 'Embryology'
if (dbp + 'Ligament>',) in bag and two == 'misc':
    two = 'Ligament'
if (dbp + 'Lymph>',) in bag and two == 'misc':
    two = 'Lymph'
if (dbp + 'Muscle>',) in bag and two == 'misc':
    two = 'Muscle'
if (dbp + 'Nerve>',) in bag and two == 'misc':
    two = 'Nerve'
if (dbp + 'Vein>',) in bag and two == 'misc':
    two = 'Vein'

if (dbp + 'Area>',) in bag:
    one = 'Area'

if (dbp + 'Award>',) in bag:
APPENDIX D. PROGRAM CODE

one = 'Award'
if (dbp + 'Decoration>,) in bag and two == 'misc':
    two = 'Decoration'
if (dbp + 'NobelPrize>,) in bag and two == 'misc':
    two = 'NobelPrize'
if (dbp + 'Biomolecule>,) in bag:
    one = 'Biomolecule'
if (dbp + 'Protein>,) in bag and two == 'misc':
    two = 'Protein'
if (dbp + 'Gene>,) in bag and two == 'misc':
    two = 'Gene'
if (dbp + 'CelestialBody>,) in bag:
    one = 'CelestialBody'
if (dbp + 'Asteroid>,) in bag and two == 'misc':
    two = 'Asteroid'
if (dbp + 'Constellation>,) in bag and two == 'misc':
    two = 'Constellation'
if (dbp + 'Galaxy>,) in bag and two == 'misc':
    two = 'Galaxy'
if (dbp + 'Planet>,) in bag and two == 'misc':
    two = 'Planet'
if (dbp + 'Star>,) in bag and two == 'misc':
    two = 'Star'
if (dbp + 'ChemicalSubstance>,) in bag:
    one = 'ChemicalSubstance'
if (dbp + 'Colour>,) in bag:
    one = 'Colour'
if (dbp + 'Competition>,) in bag:
    one = 'Competition'
if (dbp + 'Currency>,) in bag:
    one = 'Currency'
if (dbp + 'DataBase>,) in bag:
    one = 'DataBase'
if (dbp + 'Device>,) in bag:
    one = 'Device'
if (dbp + 'AutomobileEngine>,) in bag and two == 'misc':
    two = 'AutomobileEngine'
if (dbp + 'Camera>,) in bag and two == 'misc':
    two = 'Camera'
if (dbp + 'InformationAppliance>,) in bag and two == 'misc':
    two = 'InformationAppliance'
if (dbp + 'Instrument>,) in bag and two == 'misc':
    two = 'Instrument'
if (dbp + 'Weapon>', ) in bag and two == 'misc':
    two = 'Weapon'

if (dbp + 'Diploma>', ) in bag:
    one = 'Diploma'

if (dbp + 'Disease>', ) in bag:
    one = 'Disease'

if (dbp + 'Drug>', ) in bag:
    one = 'Drug'

if (dbp + 'EthnicGroup>', ) in bag:
    one = 'EthnicGroup'

if (dbp + 'Event>', ) in bag:
    one = 'Event'

if (dbp + 'PersonalEvent>', ) in bag and two == 'misc':
    two = 'PersonalEvent'

if (dbp + 'Birth>', ) in bag and two == 'PersonalEvent':
    three = 'Birth'

if (dbp + 'Death>', ) in bag and two == 'PersonalEvent':
    three = 'Death'

if (dbp + 'Divorce>', ) in bag and two == 'PersonalEvent':
    three = 'Divorce'

if (dbp + 'Marriage>', ) in bag and two == 'PersonalEvent':
    three = 'Marriage'

if (dbp + 'SocietalEvent>', ) in bag and two == 'misc':
    two = 'SocietalEvent'

if (dbp + 'Convention>', ) in bag and two == 'SocietalEvent':
    three = 'Convention'

if (dbp + 'Election>', ) in bag and two == 'SocietalEvent':
    three = 'Election'

if (dbp + 'FilmFestival>', ) in bag and two == 'SocietalEvent':
    three = 'FilmFestival'

if (dbp + 'MilitaryConflict>', ) in bag and two == 'SocietalEvent':
    three = 'MilitaryConflict'

if (dbp + 'MusicFestival>', ) in bag and two == 'SocietalEvent':
    three = 'MusicFestival'

if (dbp + 'SpaceMission>', ) in bag and two == 'SocietalEvent':
    three = 'SpaceMission'

if (dbp + 'SportsEvent>', ) in bag and two == 'SocietalEvent':
    three = 'SportsEvent'

if (dbp + 'Food>', ) in bag:
    one = 'Food'

if (dbp + 'Beverage>', ) in bag and two == 'misc':
    two = 'Beverage'

if (dbp + 'Cheese>', ) in bag and two == 'misc':
    two = 'Cheese'
if (dbp + 'Holiday>',) in bag:
    one = 'Holiday'

if (dbp + 'HumanDevelopmentIndex>',) in bag:
    one = 'HumanDevelopmentIndex'

if (dbp + 'Imdb>',) in bag:
    one = 'Imdb'

if (dbp + 'Language>',) in bag:
    one = 'Language'

if (dbp + 'Letter>',) in bag:
    one = 'Letter'

if (dbp + 'List>',) in bag:
    one = 'List'

if (dbp + 'TrackList>',) in bag and two == 'misc':
    two = 'TrackList'

if (dbp + 'Listen>',) in bag:
    one = 'Listen'

if (dbp + 'MeanOfTransportation>',) in bag:
    one = 'MeanOfTransportation'

if (dbp + 'Automobile>',) in bag and two == 'misc':
    two = 'Automobile'

if (dbp + 'Motorcycle>',) in bag and two == 'misc':
    two = 'Motorcycle'

if (dbp + 'Ship>',) in bag and two == 'misc':
    two = 'Ship'

if (dbp + 'Train>',) in bag and two == 'misc':
    two = 'Train'

if (dbp + 'Media>',) in bag:
    one = 'Media'

if (dbp + 'Name>',) in bag:
    one = 'Name'

if (dbp + 'GivenName>',) in bag and two == 'misc':
    two = 'GivenName'

if (dbp + 'Surname>',) in bag and two == 'misc':
    two = 'Surname'

if (dbp + 'PenaltyShootOut>',) in bag:
    one = 'PenaltyShootOut'

if (dbp + 'PersonFunction>',) in bag:
    one = 'PersonFunction'

if (dbp + 'PoliticalFunction>',) in bag and two == 'misc':
    two = 'PoliticalFunction'
APPENDIX D. PROGRAM CODE

if (dbp + 'Profession',) in bag and two == 'misc':
    two = 'Profession'

if (dbp + 'Place',) in bag:
    one = 'Place'
if (dbp + 'ArchitecturalStructure',) in bag and two == 'misc':
    two = 'ArchitecturalStructure'
if (dbp + 'Building',) in bag and two == 'ArchitecturalStructure':
    three = 'Building'
if (dbp + 'Infrastructure',) in bag and two == 'ArchitecturalStructure':
    three = 'Infrastructure'
if (dbp + 'Park',) in bag and two == 'ArchitecturalStructure':
    three = 'Park'
if (dbp + 'Cemetery',) in bag and two == 'misc':
    two = 'Cemetery'
if (dbp + 'Community',) in bag and two == 'misc':
    two = 'Community'
if (dbp + 'HistoricPlace',) in bag and two == 'misc':
    two = 'HistoricPlace'
if (dbp + 'Monument',) in bag and two == 'misc':
    two = 'Monument'
if (dbp + 'NaturalPlace',) in bag and two == 'misc':
    two = 'NaturalPlace'
if (dbp + 'BodyOfWater',) in bag and two == 'NaturalPlace':
    three = 'BodyOfWater'
if (dbp + 'Mountain',) in bag and two == 'NaturalPlace':
    three = 'Mountain'
if (dbp + 'Volcano',) in bag and two == 'NaturalPlace':
    three = 'Volcano'
if (dbp + 'PopulatedPlace',) in bag and two == 'misc':
    two = 'PopulatedPlace'
if (dbp + 'Continent',) in bag and two == 'PopulatedPlace':
    three = 'Continent'
if (dbp + 'Country',) in bag and two == 'PopulatedPlace':
    three = 'Country'
if (dbp + 'Department',) in bag and two == 'PopulatedPlace':
    three = 'Department'
if (dbp + 'Island',) in bag and two == 'PopulatedPlace':
    three = 'Island'
if (dbp + 'Region',) in bag and two == 'PopulatedPlace':
    three = 'Region'
if (dbp + 'City',) in bag and two == 'PopulatedPlace':
    three = 'City'
if (dbp + 'Town',) in bag and two == 'PopulatedPlace':
    three = 'Town'
if (dbp + 'Village',) in bag and two == 'PopulatedPlace':
    three = 'Village'
if (dbp + 'State',) in bag and two == 'PopulatedPlace':
    three = 'State'
if (dbp + 'Street',) in bag and two == 'PopulatedPlace':
    three = 'Street'
i f ( dbp + 'Territory>', ) i n  b a g  a n d  t w o  ==  'PopulatedPlace' :
    t h r e e  =  'Territory'
  i f ( dbp + 'ProtectedArea>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'ProtectedArea'
  i f ( dbp + 'SkiArea>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'SkiArea'
  i f ( dbp + 'SportFacility>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'SportFacility'
  i f ( dbp + 'RaceTrack>', ) i n  b a g  a n d  t w o  ==  'SportFacility':
    t h r e e  =  'RaceTrack'
  i f ( dbp + 'Stadium>', ) i n  b a g  a n d  t w o  ==  'SportFacility':
    t h r e e  =  'Stadium'
  i f ( dbp + 'WineRegion>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'WineRegion'
  i f ( dbp + 'WorldHeritageSite>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'WorldHeritageSite'

  i f ( dbp + 'Population>', ) i n  b a g :
    o n e  =  'Population'

  i f ( dbp + 'PublicService>', ) i n  b a g :
    o n e  =  'PublicService'

  i f ( dbp + 'Sales>', ) i n  b a g :
    o n e  =  'Sales'

  i f ( dbp + 'Species>', ) i n  b a g :
    o n e  =  'Species'
  i f ( dbp + 'Animal>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'Animal'
  i f ( dbp + 'Plant>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'Plant'
  i f ( dbp + 'Fungus>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'Fungus'

  i f ( dbp + 'SportCompetitionResult>', ) i n  b a g :
    o n e  =  'SportCompetitionResult'
  i f ( dbp + 'OlympicResult>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'OlympicResult'

  i f ( dbp + 'SportsSeason>', ) i n  b a g :
    o n e  =  'SportsSeason'
  i f ( dbp + 'MotorsportSeason>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'MotorsportSeason'
  i f ( dbp + 'SportsTeamSeason>', ) i n  b a g  a n d  t w o  ==  'misc':
    t w o  =  'SportsTeamSeason'
  i f ( dbp + 'FootballLeagueSeason>', ) i n  b a g  a n d  t w o  ==  'SportsTeamSeason':
    t h r e e  =  'FootballLeagueSeason'
  i f ( dbp + 'NCAATeamSeason>', ) i n  b a g  a n d  t w o  ==  'SportsTeamSeason':
    t h r e e  =  'NCAATeamSeason'
  i f ( dbp + 'SoccerClubSeason>', ) i n  b a g  a n d  t w o  ==  'SportsTeamSeason':

101
three = 'SoccerClubSeason'
if (dbp + 'SoccerLeagueSeason',) in bag and two == 'SportsTeamSeason'
three = 'SoccerLeagueSeason'

if (dbp + 'Statistic',) in bag:
one = 'Statistic'

if (dbp + 'Swarm',) in bag:
one = 'Swarm'

if (dbp + 'Tax',) in bag:
one = 'Tax'

if (dbp + 'TimePeriod',) in bag:
one = 'TimePeriod'
if (dbp + 'GeologicalPeriod',) in bag and two == 'misc':
two = 'GeologicalPeriod'
if (dbp + 'HistoricalPeriod',) in bag and two == 'misc':
two = 'HistoricalPeriod'
if (dbp + 'PeriodOfArtisticStyle',) in bag and two == 'misc':
two = 'PeriodOfArtisticStyle'
if (dbp + 'Year',) in bag and two == 'misc':
two = 'Year'

if (dbp + 'TopicalConcept',) in bag:
one = 'TopicalConcept'
if (dbp + 'Fashion',) in bag and two == 'misc':
two = 'Fashion'
if (dbp + 'MovieGenre',) in bag and two == 'misc':
two = 'MovieGenre'
if (dbp + 'MusicGenre',) in bag and two == 'misc':
two = 'MusicGenre'
if (dbp + 'Ideology',) in bag and two == 'misc':
two = 'Ideology'

if (dbp + 'UnitOfWork',) in bag:
one = 'UnitOfWork'
if (dbp + 'Case',) in bag and two == 'misc':
two = 'Case'
if (dbp + 'Project',) in bag and two == 'misc':
two = 'Project'

if (dbp + 'Unknown',) in bag:
one = 'Unknown'

if (dbp + 'Work',) in bag:
one = 'Work'
if (dbp + 'Artwork',) in bag and two == 'misc':
two = 'Artwork'
if (dbp + 'Cartoon',) in bag and two == 'misc':
two = 'Cartoon'
if (dbp + 'Film>',) in bag and two == 'misc':
    two = 'Film'
if (dbp + 'Musical>',) in bag and two == 'misc':
    two = 'Musical'
if (dbp + 'MusicalWork>',) in bag and two == 'misc':
    two = 'MusicalWork'
if (dbp + 'Album>',) in bag and two == 'MusicalWork':
    three = 'Album'
if (dbp + 'Song>',) in bag and two == 'MusicalWork':
    three = 'Song'
if (dbp + 'Single>',) in bag and two == 'MusicalWork':
    three = 'Single'
if (dbp + 'RadioProgram>',) in bag and two == 'misc':
    two = 'RadioProgram'
if (dbp + 'Software>',) in bag and two == 'misc':
    two = 'Software'
if (dbp + 'ProgrammingLanguage>',) in bag and two == 'Software':
    three = 'ProgrammingLanguage'
if (dbp + 'VideoGame>',) in bag and two == 'Software':
    three = 'VideoGame'
if (dbp + 'TelevisionShow>',) in bag and two == 'misc':
    two = 'TelevisionShow'
if (dbp + 'Website>',) in bag and two == 'misc':
    two = 'Website'
if (dbp + 'WrittenWork>',) in bag and two == 'misc':
    two = 'WrittenWork'
if (dbp + 'Book>',) in bag and two == 'WrittenWork':
    three = 'Book'
if (dbp + 'Comic>',) in bag and two == 'WrittenWork':
    three = 'Comic'
if (dbp + 'Drama>',) in bag and two == 'WrittenWork':
    three = 'Drama'
if (dbp + 'Newspaper>',) in bag and two == 'WrittenWork':
    three = 'Newspaper'
if (dbp + 'Movie>',) in bag:
    one = 'Movie'

if (dbp + 'Desease>',) in bag:
    one = 'Desease'
return((one, two, three))
Bibliography

[1] CLM Advisors. Synerscope, b.v. retains clm advisors to expand big data visualization solution for insurance industry @ONLINE, May 2013.


