Summer 8-2019

ARTIFICIAL INTELLIGENCE THE USE OF KNOWLEDGE GRAPHS

Besar Mehmeti

Programi për Shkenca Kompjuterike dhe Inxhinieri

ARTIFICIAL INTELLIGENCE
THE USE OF KNOWLEDGE GRAPHS

Shkalla Bachelor

Besar Mehmeti

Gusht / 2019
Prishtinë

Programi për Shkenca Kompjuterike dhe Inxhinieri

Punim Diplome
Viti akademik 2013/2014

Besar Mehmeti

ARTIFICIAL INTELLIGENCE
THE USE OF KNOWLEDGE GRAPHS

Mentori: PhD. Krenare Pireva Nuçi

Gusht / 2019

Ky punim është përpiluar dhe dorëzuar në përmbushjen e kërkesave të pjesshme
për Shkallën Bachelor

ABSTRACT

In this digital age, an overwhelming amount of unstructured data found in the web is increasing at an unprecedented rate. In response, information extraction techniques have been developed to automatically extract information from unstructured text and populate knowledge bases. This thesis will introduce the reader to the idea of Knowledge Graphs, the history of how they were first experimented on and later developed, and in particular, this thesis is going to elaborate how Diffbot, Wikidata and IBM Watson technologies are modeled, stored and queried. By having this information at our fingertips, we believe that we can make smarter decisions, keep up with business trends, and even find ways to further improve our overall operations regarding data analysis. Different businesses demand different types of Business Intelligence Software. To understand well which one suits us, through this study we have evaluated features of Diffbot, Wikidata and IBM Watson by taking into consideration their conditions, features and costs.

ACKNOWLEDGEMENTS

I would like to thank Prof. Krenare Pireva Nuçi for the ongoing support throughout the completion of this thesis, as well as for the continuous motivation and advices given throughout this period. Moreover, I would like to thank my family for the continuous support and all the opportunities they have given to me in order to finish my studies. Lastly, a big thank you also goes to the academic staff of UBT, for the motivation, support and the untired work that they do daily.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION LIST

AI – Artificial Intelligence

ANI - Artificial Narrow Intelligence

AGI - Artificial General Intelligence

GM – General Motors

ACS – American Cancer Society

KG – Knowledge Graph

UI – User Interface

DOM - Document Model Object Parsing

PHP - Hypertext Preprocessor

JSON - JavaScript Object Notation

WWW – World Wide Web

SQL - Structured Query Language

DQL – Diffbot Query Language

1.Introduction

The concept of Artificial Intelligence (from now on referred to as AI) has been in existence since the days of Aristotle, but it was brought into attention in the 1950s by Alan Turing, who is widely known for breaking the famous Nazi Enigma Code, which turned the tides of World War II in favor of the Allied Forces [1]. In 1950 Turing published a paper called "Computing Machinery and Intelligence" in which he proposed the concept of "Thinking Machines" and tried to define what it means for a machine to be percieved as intelligent with his "Turing Test" [2]. Although it has been around as a concept for a long time, it has proved a bit tougher to truly pinpoint a definiton for AI. John McCarthy, who is regarded as the father of AI, first coined the term in 1956 and defined it as "The science and Engineering of Intelligent Machines". He proposed that for a machine to be considered autonomous in its thinking, it would have to do something, or come up with a solution which if done by a person and would have to involve intelligence [3].

Today, we can see hundreds of examples of AI in daily use. From personal assistants like Apple's "Siri" or Microsoft's "Cortana", search engines like Google and Bing, even social networks like Facebook and Twitter which use algorithms to ensure you see the content that is most relevant to you. There is no singular modern definiton of AI that is uniformly agreed upon, mainly beacause the problem with defining "Thinking Machines" is that we first have to define what constitutes as "thinking" or "intelligence." According to a published book by Stanford University, “the goal of work in AI is to build machines that perform tasks normally requiring human intelligence” (Nils J. Nilsson, 1971). Another source defines AI as a “a branch of computer science dealing with the simulation of intelligent behavior in computers” and “the capability of a machine to imitate intelligent human behavior” (Mirriam Webster, n.d).

AI generally tends to fall under two main categories, the first one being Artificial Narrow Intelligence (from now on referred as ANI) and known as the Weak AI (Ben Goertzel, 2010), is designed to perform tasks that are very specific in nature and are merely designed to simulate human intelligence. These machines are generally good at focusing at doing one specific task and doing it as well as possible, like projecting a path, reporting the weather, or playing poker against a human opponent. On the other hand, the second category falls under General Artificial Intelligence (from now on referred as AGI), sometimes also referred to as "Strong AI" is designed

to come up with a solution to completing any task which would previously require the intervention of humans. This type of AI can learn and develop new solutions autonomously as it is faced with obstacles and problems. AGI is still not close to being achieved, but it remains a goal in the field of AI [4].

A definition of AI that this thesis has come up with, is that AI is the study and design of "Intelligent Agents," in which case an intelligent agent is a system or a program which studies and perceives its environment and decides to take the actions which increase its chances of success. For this reason, this thesis will concentrate on expanding the horizon of AI in terms of introducing the notion of *Knowledge Graphs.*

1.2 Organization of Thesis

In order to treat the subject as thoroughly as possible, the remainder of the thesis is organised as follows:

In Chapter 2 is covered the Literature Review which introduces the different methods, techniques, and technologies which are used in order to gather the data which are required to populate the knowledge graphs. The thesis presents the problem declaration in Chapter 3, together with the aim and objectives. Further, in Chapter 4 is presented the methodology, which defines the methods used within this thesis and finally in Chapter 5, is the comparition and evaluation of selected technologies in Knowledge Graphs, and try to understand what sets these technologies apart form one another, and what components they are actually composed of. This thesis will mainly examine and analyse the way that data has evolved from classic databases to the structure of a knowledge graph and the potential of using these products either for individual or for business purposes.

2.Literature Review

In today's world, where virtually every device and every machine are connected via the World Wide Web (referred to as WWW further), having intelligent agents which can navigate and organize through all that data is becoming more and more important. An overwhelming amount of unstructured text, such as newspaper articles or social media, is increasing at an unprecedented rate. With the advances in Internet technologies and increase in smart-phone use, communication has become increasingly fast, and events happening in one part of the world get communicated to the rest of the world in a matter of seconds or minutes. The sheer amount of data in the WWW means that a need has arisen for a way to categorize and arrange it all into pieces of consumable information. In response, information extraction techniques have been developed to automatically extract information from unstructured text and populate knowledge bases [5].

In order to get closer to the concept of how Knowledge Graphs are introduced into the thesis, we will closely introduce and elaborate topics such as: Web Scraping, Web Scraping Techniques, Web Crawling, Web Crawling Techniques, Web Scraping VS Web Crawling, Graph Database and *Knowledge Graphs.*

2.1 Web Scraping

Web Scraping (which is more commonly referred to as data scraping) is a collection of techniques ranging from manual techniques performed by humans, to more automated technologies which are used for extracting information from web pages on the internet. Scraping does not necessarily relate to websites, since scraping can be done on a local machine or a local database (Boeing, G. Waddell, P, 2016). Scraping is a process which can obviously be completed manually, but it is much faster, and a lot less prone to mistakes and errors if it is automated, especially if there are thousands of documents to extract. There are hundreds of ways how data or information can be displayed online, considering that most of the data or text in the internet is not structured or tabbed into organised content [6]. Here comes in the picture the web scraping technique, which fascilitates the process of displaying the unstructured information by converting them into tabular data. The process of transforming the unstructured to structure data is depicted in Figure 1, which emphasizes the web scraping technique as a central point, which during the crawling from the websites takes the data as an input and transformed the output into structured data.

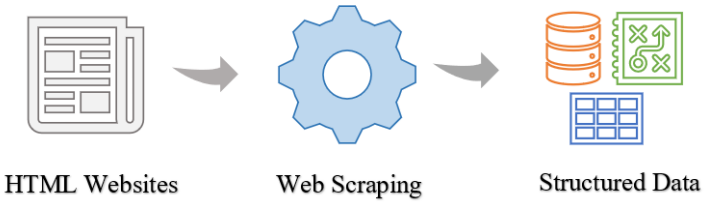

Figure 1.The transformation of data from raw unstructured text form to structured data.

2.1.1 Web Scraping Techniques

There are many different techniques available that are used for web scraping today. Some of the main techniques in use today are:

- **Manual Copy and Paste** - also referred to as Manual Scraping is the act of manually copying and storing every piece of relevant information that is required by the user. The act of clicking "Save As" on an online document can be considered a manual web data extraction. This technique is very monotonous and repetitive, which is why it is usually automated, but it is effective especially on websites with protection against bots;
- **HTML Parsing** - is accomplished using script, and generally targets nested HTML pages. This method is moslty used for text extraction and link extraction;
- **Document Model Object Parsing** (which will be referred to as DOM Parsing further**)** – DOM Parsing is usually used to structure information and contents inside XML files. DOM Parsing is usually employed to get access to an in-depth view of the internal structure of a web page;
- **XPath** - XML Path Language is generally also used on XML documents. This technique is usually used together with DOM Parsing to copy the contents of an entire webpage and publish it somewhere else. The usage of these specific techniques over one another depends mostly on the way the data resides and the way it is represented online. For example, HTML Parsing is employed for classic HTML web pages while something like XPath is generally more friendly to XML documents. Another factor which decides which techniques are employed is the purpose that the data will be used for after the extraction is

completed. Most of the time, more than one of these techniques will be utilized at the same time, and an effective scraper will employ multiple techniques in combination [7].

2.2 Web Crawling

Web Crawling (which sometimes is referred to also as Data Crawling), began as a movement which was concerned with "mapping out the internet" which means finding out about the structure of each website and all the ways in which they are connected to each other. Web crawling is mostly used by search engines, for example Google or Bing, in order to index all the pages for easier search and access in a later date, although it can be used for other purposes like the automation of website maintainance tasks or finding security flaws. Google alone averages billions of searches every day. People rely on search engines for retrieving specific pieces of information relevant to them in the shortest time possible. But with the nature of the WWW being that it is always changing and expanding, scouring through all of the content in the WWW for a single search is not a realistic solution. Because there is virtually an infinite amount of data to search through, search engines employ the services of crawlers to search through, categorize, store and index web pages in order to access them easier later. According to a published paper by Christopher Olston and Marc Najork, a web crawler (also known as a web spider, a web bot, or simply a crawler) "is a program or a robot which is designed to browse through all of the content of the web in an automatic and systematic fashion" [8]. This means that it will browse every single page of a website, searching for links which lead to other pages or other websites, until it has mapped everything out and gathered all of the newest and updated information, which is then saved somewhere where the search engine can access it later.

According to a scientific contribution given by Multimedia University in Malaysia, the simplest crawling methodology is derived as per Figure 2. This figure shows how a Web Crawler has a typical life cycle which starts with set of the seed known as Uniform Resource Locator (URL), which is originally extracted from the web cache or the domain name server, and then digs deeper (Download) into the URL to fetch the sub URLs found into the main page [9]. By this we can already conclude that the web crawler is an essential component of search engine.

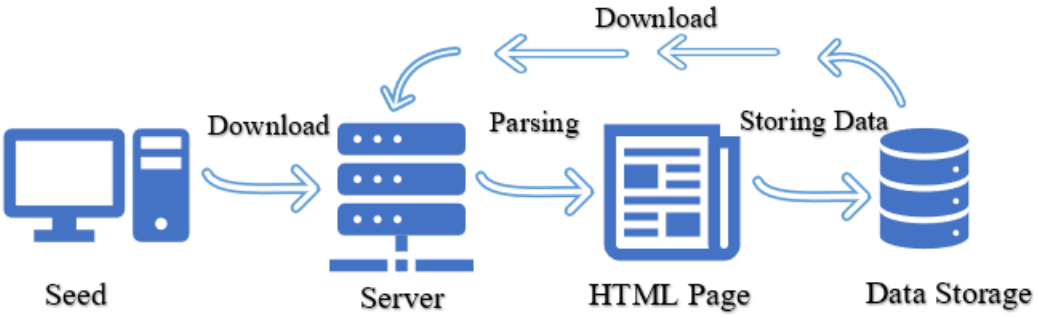

Figure 2. Web Crawler Methodology

2.2.1 Web Crawling Techniques

Further to subsection 2.1 where the life cycle of a web crawler is presented, it is crucial to further understand that the crawling method used by various search engines relies greatly on the crawling technique which it uses [10]. On this section we will briefly examine these techniques and capture what do they cover.

In general, there are four main web crawling techniques as follows:

1. **Focused Crawlers** are mainly tasked with downloading web pages that are related in some way to each other and prioritizing collecting pages usually specific to a certain topic. It calculates how relevant a given page is to the given topic and whether it should download it or not based on the relevancy [11]. The focused crawler determines how far the given page is relevant to the topic and how to proceed forward, hence the name of the crawler, a term coined by Chakrabarti et al. (Chakrabarti, Berg&Dom,1999). Since 1995, this crawler found use by the search engine AltaVista, now a property of Yahoo! [12];

2. **Incremental Crawlers** frequently visits the same set of web pages, in order to determine how often a page generally changes, so it can retrieve the newer updated versions of the pages. It exchanges the older, less important pages with more important ones. The

incremental crawler ensures that only the most valuable data is shown to the user, which means search times are lower and bandwidth is conserved;

3. **Distributed Crawlers** work in groups in order to distribute the process of crawling, to further widen the coverage of the World Wide Web. Apart from Google, Yahoo! and Microsoft Bing, other search engines that support this query are IndieGogo, YaCy and Grub;

4. **Parallel Crawlers** also run in groups, but they usually run in parallel due to the continuous growth of the Web size. This way, two or more crawlers are focusing on splitting the same task in order to perform faster. Parallel crawlers are very important for ensuring download times remain as low as possible [13]. According to a study, "a parallel crawler is implemented in a distributed fashion to crawl educational domains ending in .edu" (Ankur M. Patwa, 2006), and systems like Needle and Mercator use it [14].

2.3 Web Scraping VS Web Crawling

Having the opportunity to study and analyse the work of a number of researchers, the concepts of data scraping and data crawling are sometimes mistaken with one another, although both of them share similarities in the sense that they are exploring the internet for data, there are some distinct differences in what they result to achieve [15].

From one side, data scraping usually involves at least a degree of data crawling, since the scraper has to crawl around the web searching for the information it has been tasked to recover, but while a crawler is usually focused on finding and storing every single part of a web page, a scraper from the other side is looking for specific information within the data provided. Data scraping is usually applied in any scale, starting from a collection of documents to an entire database of information; whereas crawling is generally deployed only on a larger scale.

Further, the difference between data scraping and data crawling from the redundancy or duplication point of view is also interesting. In web scraping, duplicates are welcomed and used for measuring the accuracy of data, (for example in two articles about the same topic published in different websites) whereas in web crawling, deduplication is very important in order to remove unnecessary copies of the data (Arpan Jhan, 2012). Furthermore, Table 1 lists the key differences between Data Scraping and Data Crawling and shows the properties that are most confused with one another when speaking about scraping versus crawling. There are many other differences in both the techniques that are utilised for scraping/crawling and the way that they deal with the information presented, but the concepts presented in the table are the most important ways in which these two technologies are differentiated from one another.

Data Scraping	Data Crawling
Involves extracting data from various sources including web	Refers to downloading pages from the web
Can be done at any scale	Mostly done at a large scale
Deduplication is not necessary a part	Deduplication is an essential part
Needs crawl agent and parser	Needs only crawl agent

Table 1. Key differences between Data Scraping and Data Crawling.

According to the above, Table 1 describes the overall differences from the process point of view.

To conclude, while both web scraping and web crawling begin their process by being fed with a list or a database of websites, the different forms that the data ends up being saved in, and the different ways that data will be utilised is one of the key differences. While the main objective is to transform the data into an organised and structured shape, a crawler will mainly save the data into a giant database for further querying later (mostly for search results) while the structured data of a scraper can be used in a multitude of ways such as XML and SQL which makes it further compatible with different applications and programs which are used online [16].

2.4 Graph Database

In this section we will analyze the concept of storing data in a graph structure, and the evolution of the storage of data from the classical relational database system to a graph system. These concepts are the concepts which led to the discovery of using knowledge graphs for the storage and retrieval of structured data. We will also look at the differences between the classic database model versus the graph model, and the advantages that the graph model has over the relational database model, according to a guide created by the developers of the graph database management system - Neo4j.

A database is a collection of information stored in an organized manner, usually represented and organized in tables, with rows and columns [17]. The data inside a database is meant to have the ability to be accessed, changed and updated by the people who are using the database. Modern databases and managed by Database Management Systems (also referred to as DBMS) which are systems developed to provide tools for manipulating the data inside the database using Queries. As well as the data itself, the relationships between the data are what allows users to make sense of all the information and develop tools and applications in order to use that information for what is required of them, which can be anything from keeping the records of sales/purchases of a business, to analyzing the performance of a sports institution for example. The relationship between the tables brings us to the concept of “relational databases” where the data resides inside rows and columns in the pre-built tables and we can access and change that data without the need to reorganize the tables themselves. In a relational database model, the relationships and the references to other rows and other tables are represented by referring primary keys from foreign key columns (which are both represented using numerical identifiers). These custom-made

relationships are referred to as “Joins”. Joins match the primary key of a table row to a foreign key of a row in another connected table and are usually computed at query time, which means that the more complicated a relationship or a “Join” is, and the more variables it has to connect to, the computing cost will usually increase exponentially. In a relational database schema, the data inside the table is the most important part, and the relationships between the data are secondary.

As established in the Chapter 2, there is a growing movement towards presenting data in a structured, semantic form which is more friendly to being queried with natural, human langage. And such, the concept of storing data in the form of a graph was born. A graph database is mainly based on the concept of a mathematical graph which is composed of nodes which represent the different entities, and the edges (or lines) which represent the acutal relationships between the data. In a graph pattern, the relationships between the different entities are equally as important as the data or entities themselves. In a graph database, every node has its own identifier, and also every edge has its own unique identifier, but every edge may also contain its own set of properties [17].

Figure 3 and 4 below represent the difference in how the storage of information is visualized in a classic relational database versus a graph database.

Sales		
Customer	**Item**	**Time**
0001	1A	20:34
0001	1A	21:15
0003	2A	21:16
0002	1A	21:16
0002	5C	21:34

Inventory	
Description	**SKU**
Pepsi	1A
Club Soda	2A
.	.
.	.
Diet Coke	5C

Customer	
Name	**CustID**
John	0001
Jack	0002
Ted	0003
Ken	0004
Valerie	0005

Figure 3. A visualization of how data is stored in a relational database, and how the relationships between the data are organized using identifiers.

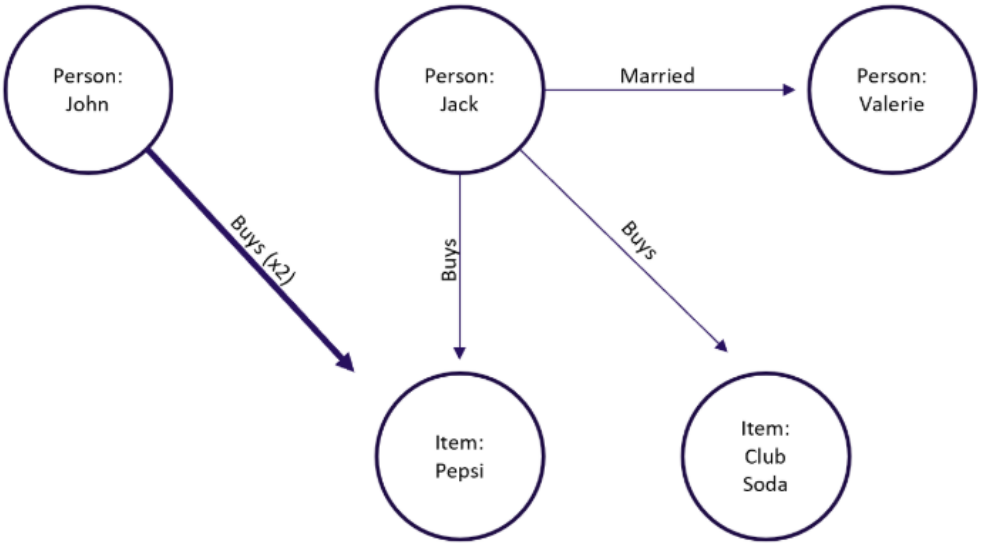

Figure 4. A visualization of that same data being stored in a graph pattern. The entities are represented by circles, and the relationships are represented by the lines between them [18].

As seen in the figures, a graph database has the ability to store the same type and the same amount of data, but it also has the ability to store the relationships between the data as entities of their own, which opens the way for a multitude of functions which a classic database cannot perform without immensely complicated Join functions. The relationship between the data is visibly apparent without having to create any new functionality for it.

While relational databases store all the information in pre-set columns and rows inside tables, graph databases do not have a rigid pre-determined structure. All of the relationships between the data are stored in the edges of between the nodes, and with the ability for the edges and the ends of the edges to have their own properties, this structure opens the way for representing very complex relationships between completely unrelated data sets.

Some other aspects to note when it comes to the differences between a relational database and a graph database will be analysed below. The flexible nature of the graph schema allows it to change over time and adapt to new data and even new types of data without being constricted by pre-set tables. The fact that the relationships between the data are also entities themselves and the semantic nature that the data is stored in, allows the data to be queried with natural human language rather than different programing languages like SQL (Structured Query Language), which is the most popular database query language. Even when looking at a visualization of a graph, it is much more instinctive to the way humans think using relations, as opposed to tables of rows and columns. Another important distinction is the computing cost of querying the information. In a classic relational database, the computing cost increases exponentially the more complicated a query is, while the cost of querying a graph remains mostly the same no matter how complicated the relationship between the data is [18]. Table 2 will represent the key differences between a classic relational database and a graph database.

Relational Database	Graph Database
Rigid schema which is not meant to be changed	Flexible schema which is meant to be adaptive and change as new data arrives
Relationships between the data are secondary to the data itself	Relationships between the data are just as important as the data itself
Queried using Joins and other SQL functions	Queried using natural human language
Cost increases exponentially the more complicated a query is	Cost remains constant no matter how complicated a query is

Table 2. Key differences between a classic relational database and a graph database.

Although the classic relational database model has been the backbone of computer applications for almost 40 years, it is far from becoming obsolete and there are still many advantages towards using the relational database model over the graph model, if the relationships between the data are not crucial to what you are trying to achieve. It cannot be denied though that the graph model has multiple advantages when it comes to creating very complex relations between unrelated sets of

data and being able to present that data in a structured form which can be accessed and changed by natural human language. The advancements in graph databases gave way to new combinations of such technologies with technologies like machine learning, and natural language processing, and that led to the conception of knowledge graphs, which will be analyzed in more depth in the next chapters.

2.5. Knowledge Graphs

It is crucial to understand how the topics mentioned above correlate with the notion of Knowledge Graphs, thus, this will be accordingly explained on the following chapters. According to Amit Singhal – former vice president of Google Inc., a knowledge graph can be described as a collection or a network of real-world entities and the relationships between them, organised and arranged as a graph [19]. The way it is different from a classic database of entities or a knowledge base is that the relationship between the data is just as important as the data itself, which is why it is arranged in a graph pattern as shown in Figure 5.

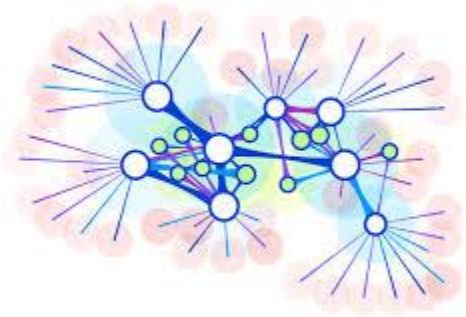

Figure 5. The way data resides in a knowledge graph. Different entities and concepts are represented by circles, and the relationships between them are represented by lines [20].

Every knowledge graph is essentially a knowledge base, but, being able to see how all of the entities are or are not connected to each other, adds a whole new level of value to the data. The graph pattern also allows it to be very flexible when introduced to new data or to new types of entities entirely, since there is an established framework already present. The fact that a knowledge graph is essentially a mathematical graph, opens the way for a multitude of techniques and algorithms related to graphs to be applied to knowledge graphs. This allows for information like

predictions or effectiveness to be extracted or derived from the graph which would not be apparent when analyzing with human eyes [20].

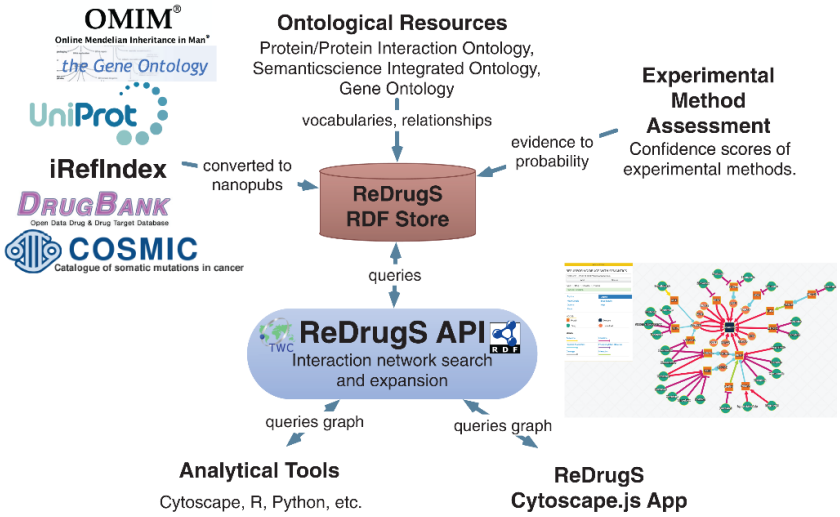

Figure 6. A case of the data inside a knowledge graph being used for experimental knowledge assessment and the development of analytical tools used in the Pharmaceutical Industry [21].

The data inside the knowledge graph is also self descriptive and can be considered semantic in nature, because in addition to the data, the meaning of the data is also saved with it (see Figure 7).

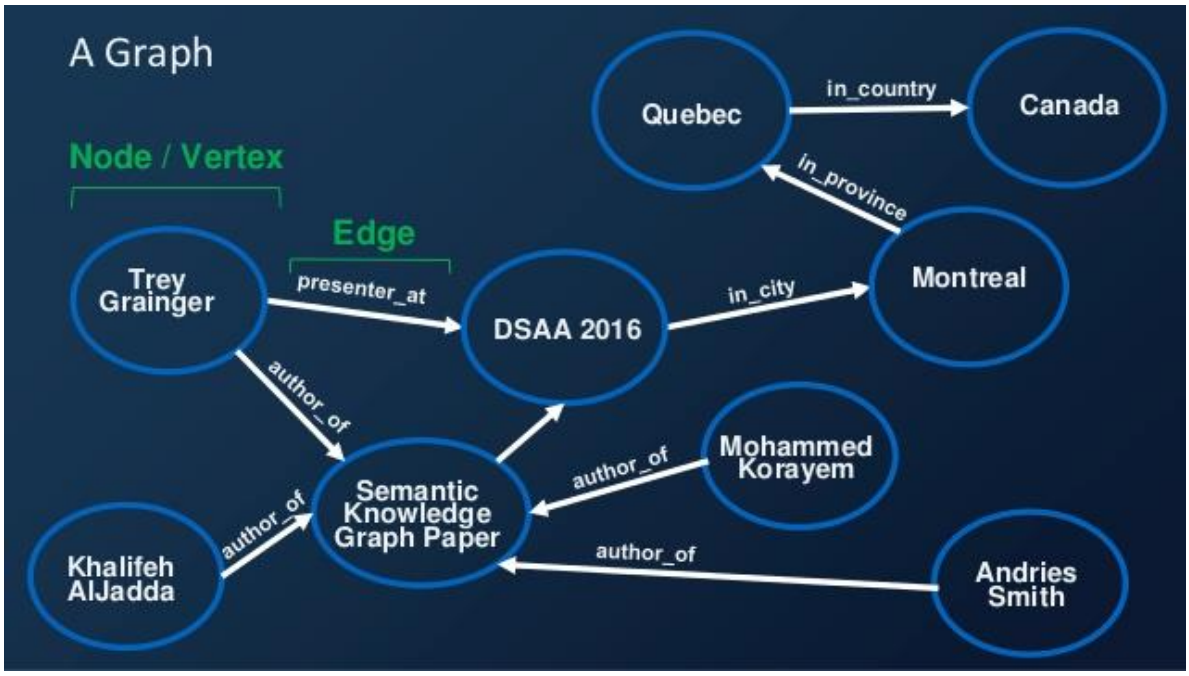

Figure 7. The semantic nature of the data inside a knowledge graph.

According to the Figures 6 and 7, we can gather that every entity is identifiable by what it is and also, by what it means. This opens the way for expanding the relationships between the data even further. The semantic nature of the data means that it can be queried or searched in methods that resemble natural human language, which in turn leads to smarter and faster search and discovery [22].

Figure 8. An example of a natural human language query providing a (A) semantic best effort answer and (B) an exact answer [23].

According to the study shown as in the example above, the system has used the open-domain DBpedia and YAGO to gather the required information, and then collect and show it as an exact answer, keeping in mind that the system is trying to understand what the human is requesting [23]. By this, we can assume that flexible structure of the knowledge graph allows it to grow even more powerful as new data arrives. The underlying ontology that is used to categorise and arrange the data can be updated and reorganised as new entities are introduced. This means that a knowledge

graph can be subjected to a continuous stream of data, and it will reshape and refine itself the more data it is introduced to.

2.6 Machine Learning

It is safe to say that advancements in AI gave way to the conception of many new fields of research. One of the fields that is birthed from advancements in AI which takes part as one of the important themes to the examination of the thesis, is Machine Learning. According to Daniel Faggella, "Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions" [24]. According to Alex Smola, much of the art of machine learning is to reduce a range of disparate problems to a set of fairly narrow prototypes. Much of the science of machine learning is then to solve those problems and provide good guarantees for the solutions [25]. As a simpler definition produced for this thesis, is it safe to say that machine learning expresses the ability of a machine or a computer to actually learn and improve from external stimuli without being explicitly programmed to do so.

Until now, we have been introduced to several notions and topics that help to have a deeper understanding of how Knowledge Graphs are incorporated in the usage of creating learning machines. In practice, it is also crucial to comprehend how in reality a learning machine works, meaning that, how does the public perceive its funcionality. Machine Learning is everywhere these days. It is applied in various software starting from the simple ones; the act of classifying and organizing our photos in our smartphones, to more complex softwares, which help us predict the next action with a high accuracy, i.e self-driving cars. This approach is being used in other various cases, such as in our email account which filters out spams and other emails we don't want to read, to recommending products and personalizing online shopping experience at Amazon. These are all factors that influence the creation of a learning machine with the help of Knowledge Graphs. Recently, the biggest players battling to dominate the very fast-growing machine learning services are Amazon, Google, IBM, and Microsoft. If we take the case of Google, who are the pioneers of the unveiling of knowledge graphs; the technology was used to further enhance the capabilities of its search engine, to provide more accurate results which are more sensitive to what is actually

being searched, and to add a natural language element to its searching capabilities. On the other hand, IBM has also taken a different approach to the technology with its main IBM knowledge graph called Watson; a more general-purpose tool in the sense that it can be adapted to many different fields and occupations, and its main qualities are based upon strong language processing and data analysis. Amazon uses technologies like machine learning and knowledge graphs in order to refine the process of suggesting their buyers what to buy based on previous purchases. Lastly, the Bing knowledge graph contains information about the world and powers question answering on Bing. This is the largest knowledge graph at Microsoft, as its aim is to contain general knowledge about the entire world [26].

3. Problem Declaration

From a realistic point of view, the current age we live in is characterized by the immense amount of ways that we as humans can extract information about any need. There is a virtually infinite amount of data that lives online and the only thing that is required to gain access to it is an internet connection. The fact that the data that is available online is massive, and it even grows exponentially year by year introduces a few problems of its own when it comes to organizing all that data and making it presentable in a useful form, and that is further complicated when that data originates from multiple sources, which means that verifying which data is true and which is false becomes a problem of its own. The majority of information that is available online, be it scientific information, or non-scientific information is stored in a classic textual form, which is raw and unorganized, and the first search engines which were constructed to sort through this data were based on searching through long strings of texts in order to determine what was relevant and what not [25]. Different companies have already started tackling this task of structuring and organizing all of this data online into useful information. If we take Google as an example, who were one of the first companies to start experimenting with the technology, they already had access to the biggest data pool in the world with its search engine. The Google Knowledge Graph already has billions of entities which populate its data, and the majority of that data was constructed using data mining algorithms and Web Crawlers in order to first of all gather the data, and then using data extraction and natural language processing in order to construct the information in structured form.

3.1 Aim

For the purpose of this thesis, we are analyzing an already-conducted research [27] about the essence of finding the most suitable and effective Knowledge Graphs (from this chapter on referred to as KG), for any given setting. The methodology used for our research will contain similar tactics like [27]. In order to reach the expected result, certain KGs will be analyzed each and compared with each other. The selected KGs for this thesis are Diffbot, Wikidata and IBM Watson Discovery.

3.2 Objective

In order to achieve the aim of the thesis, the following objectives have been considered:

1. Using the literature review to identify the contribution in KGs.

2. Conducting an in-depth analyze of Diffbot, Wikidata and Watson Discovery, in terms of how they are created, stored and queried;
3. Comparison between the above-mentioned KGs;
4. Finding the most suitable KG for individual needs, according to the products features.

4. Methodology

This paper is structured and composed in a way that it tries to represent an extended elaboration of the notions that make up KGs. All together, until now we have been introduced to the concepts such as Artificial Intelligence, Machine Learning, Web Scraping and Crawling etc. The introduction of these concepts helps us to get a fundamental perspective on the so-called spectrum of how KGs work. The research has used a number of methods for each of the objectives specified in section 3.3. For Objective 1, the literature review is used to identify the contribution in KGs, to identify the fields of use and further to narrow the aim of this thesis (see Chapter 2). Objective 2 is conducted by giving a systematic overview of these KGs in their current versions and discuss how the knowledge in these KGs is modeled, stored, and can be queried (see Chapter 4). Further, Objective 3 is conducted by using comparison method for these selected KGs according to the gathered information and achieving results (Chapter 5). These methods fascilitate the process of differentiating all three providers based on: Main Features, Pricing Model, Language Support, Devices Support, Integration, Prominent Clients Etc. Lastly, Objective 4 is to reach a conclusion which is to see which of the selected KGs is proven to be the most effective and suitable for the audience.

5. Knowledge Graph technologies

5.1 Diffbot

Diffbot was founded by Michael Tung in 2008 in the campus of Stanford University as a startup with the goal of revolutionising Web Data Extraction and making it accessible to everyone. Michael Tung began the development of Diffbot after dropping out of Stanford's grad school and in order to find funding, he turned to filing out patent laws, so he could keep working on the development of Diffbot. Later, Diffbot became the first company to be backed and funded by Stanfords venture capital fund called StartX. The arrival of much needed funding allowed Tung to continue focusing on developing Diffbot and allowed him to launch the first products based on Diffbot which generated revenue, by making it essentially an on-demand service, where companies would pay a small fee for each URL which was processed [28]. Many of the initial customers, such as Ebay, Microsoft, and Cisco continue to use the services of Diffbot to this day and allowed it to expand its initial array of knowledge exponentially (Bernatte Tansey, 2018). The AI industry giants such as Google all have access to enormous amounts of data, which are organised and categorised by their data entry employees in order to present them in a language which can be understood and manipulated by AI software. By contrast, smaller companies which are developing AI solutions do not have access to this magnitude of data, which is why Diffbot is such an appealing solution. In 2016 they began on the development of a "true knowledge graph" called Diffbot Knowledge Graph with the aim of designing and building the "world's largest database of structured knowledge." Utilising a combination of AI, machine learning, natural language processing and computer vision, the Diffbot Knowledge graph is tasked with transforming all of the knowledge on the internet into a singular source of information which is organised and structured as data, answers and insights. The database already contains upwards of 1 trillion facts and 10 billion entities, which already contains 500 times more information than Google's Knowledge Graph [29].

5.1.1 Diffbot as a technology

During Literature Review on Chapter 2, we have been introduced to notion of web data extraction and its techniques. On the case of Diffbot, the data extraction solution is equipped with a comprehensive KG that contains accurate and detailed information about different entities found

on the web such as people, places, companies, organizations, businesses, products, articles, and discussions. According to [https://reviews.financesonline.com], users can easily and precisely query this KG to surface whatever data they need. In addition, Diffbot identifies how information and entities are connected with each other, making it easy for users to make sense of the data delivered by the software and utilize it to reach their specific goals. The Diffbot product is a set of APIs (Application Programming Interface) that allow you to retrieve specific types of structured data from the web. The APIs analyze the page type, extract all of the elements of an article, product, or discussion thread and return data about images and videos. Any of these APIs work with their *Crawlbot*, which can create a "structured index of practically any site's data." As well, Diffbot's custom API toolkit allows users to override, correct or add fields to any of the automatic APIs. (Mike Tung, 2008). As a conclusion, the definiton on how the data should be constructed is a very important part of a KG, but in order to construct the KG itself, a method for storing it at the data and concept level should be defined.

The method can be either fully automated using technologies described earlier in this paper such as data scraping or data mining, or it can be done semi-manually with the help of a human operator. In the case of Diffbot, the extraction is done automatically [30]. In the tables below we have summarized a list of all the relevant technical details of Diffbot, including its features, price, costumer-type support, device support and language support.

No.	Features	Sub-features
1	Data Extraction	✓ Disparate Data Collection ✓ Document Extraction ✓ Email Address Extraction ✓ Image Extraction ✓ IP Address Extraction ✓ Phone Number Extraction ✓ Pricing Extraction ✓ Web Data Extraction
2	Data Mining	✓ Data Extraction and Data Visualization × Fraud Detection × Linked Data Management ✓ Machine Learning × Predictive Modeling ✓ Semantic Search × Statistical Analysis
3	Load Generation	✓ Contact Discovery ✓ Contact Import/Export

		✓ Lead Capture × Lead Database Integration ✓ Lead Nurturing and Scoring ✓ Lead Segmentation ✓ Pipeline Management ✓ Prospecting Tools
4	Sourcing	✓ Auction and Budget Management ✓ Collaboration ✓ Global Sourcing Management ✓ Rfx Management × Spend Management ✓ Supplier Management and Qualification ✓ Supplier Risk Management ✓ Supplier Web Portal × Template Management

Table 3. An overview of features and sub-features of Diffbot, gathered from an online B2B Provider, Capterra [30].

Further technical details are described on the tables below.

No.	Pricing Plan	Price
1	Free Trial	✓
2	Startups	$299/month
3	Plus	$899/month
4	Professional	$3,999/month
5	Enterprise	No Data/ Custom Quote

Table 4. Offered Pricing Plans as monthly-basis subscription plans according to the business needs [30].

In addition to the information above, it is important to add that the devices supported by Diffbot are: Windows, Linux and Mac, in several languages such as: English, Chinese, German, Japanese, Spanish, French and Russian. According to [30], attention to Diffbot is given by Small Businesses, Large Enterprises and Medium Businesses.

5.1.2 Generated queries from Diffbot

We already know that Diffbot has the ability to query the web as a database. We also know that this feature is based on a KG. According to sub-section 4.1.1, the difference between Diffbot and other knowledge bases is that Diffbot's KG is only partially curated by humans and is automatically populated by crawling the web.After crawling the web with its search engines, the results are stored in the web as documents-thus the importance of learning the concept of Machine Learning. This machine then will try to mimic the way that humans would break down the document (Michael Tung, 2018). In this section, we are going to elaborate further another concept regarding the querying of the information from the web. There is already a bunch of graph query language, such as SPARQL, Gremlin, Opencypher etc. Diffbot has its own query language called Diffbot Query Language (DQL) [31]. In this section, we are using an example of a query to determine how Diffbot generates the query and what results it gets. In this case, we have used our *Crawlbot* to start a new request and have synced it with an URL Seed. Steps to run the crawl are mentioned below:

1. Enter a new crawl name;
2. Enter a seed URL, https://api.diffbot.com/v3/analyze?token=...&url=... ;
3. Select the Analyze API from the "Diffbot API" menu;
4. Click "Start".

In the case of “Analyze API”, each page found on the site will be analyzed and all supported page-types (article, discussion, image, product, etc.) will be automatically extracted and made available in the resulting collection [32].

With the developer tools, it is instructed to provide the main and optional arguments such as;

<token>

<url>

<mode>

<fallback>

<fields>

<discussion>

<timeout>

<callback>

The Analyze API returns data as a JSON (Java Script Object Notation) format file. Each response includes a <request> object (which returns request-specific metadata), and an <objects> array, which will include the extracted information for all objects on a submitted page. If the Analyze API identifies the submitted page as an article, discussion thread, product or image, the associated object(s) from the page will be returned automatically in the <objects> array.

5.1.3 Example response as JSON format

Because the below classified page is an article, the extracted example response is shown on Figure 9 below.


```
{
  "request": {
    "pageUrl": "http://tcrn.ch/Jw7ZKw",
    "resolvedPageUrl":  "http://techcrunch.com/2012/05/31/diffbot-raises-2-million-seed-round-f
or-web-content-extraction-technology/",
    "api": "analyze",
    "options": [],
    "fields": "",
    "version": 3
  },
  "objects": [
    {
      "type": "article",
      "resolvedPageUrl":  "http://techcrunch.com/2012/05/31/diffbot-raises-2-million-seed-round
-for-web-content-extraction-technology/",
      "pageUrl": "http://tcrn.ch/Jw7ZKw",
      "human_language": "en",
      "text": "Diffbot , the super-geeky/awesome visual learning robot technology which aims t
o \"see\" the web the way that people do, is today announcing a new infusion of capital. The c
ompany has closed $2 million in funding from a number of technology veterans, including EarthL
ink founder Sky Dayton ; Andy Bechtolsheim , co-founder of Sun Microsystems; Joi Ito , Directo
r of MIT Media Lab; Brad Garlinghouse , CEO of YouSendIt ( and formerly of TechCrunch parent c
ompany AOL ), Maynard Webb , Chairman of the Board at LiveOps, formerly eBay COO; Elad Gil , V
P of Corporate Strategy at Twitter; Jonathan Heiliger , former VP of Technical Operations at F
acebook; Redbeacon co-founder Aaron Lee ; and founder of VitalSigns Montgomery Kersten .\nMatr
ix Partners also participated in the round. Of the new investors, Sky Dayton will be the first
to join Diffbot's board and will be taking an active role in the company, including plans to g
o hands-on with various Diffbot projects.\nLast August, the company publicly debuted its first
APIs , which allow developers to build apps that can automatically extract meaning from web pa
ges. For example, the FrontPage API is able to analyze site homepages, and understands the dif
ference between article text, headlines, bylines, ads, etc. The Article API can then extract c
lean article text, images and videos. Another example of Diffbot in action is the \"follow API
,\" which can track the changes made to a website.\nToday, Diffbot has categorized the web int
o about 20 different page types, including homepages and article pages, which are the first tw
o types it can now identity. Going forward, Diffbot plans train its bots to recognize all the o
ther types of pages, including product pages, social networking profiles, recipe pages, review
pages, and more.\nIts APIs have been put to use by AOL (again: disclosure, TC parent) in its n
ews magazine AOL Editions , as well as by companies like Nuance , SocMetrics , and others. Dif
fbot says it's now processing 100 million API calls per month on behalf of its customers. Thou
sands of developers are using the APIs, the company notes, but paying customers are only in th
e \"tens.\" Correction: we're now told they have \"a lot more!\"\nDiffbot founder and CEO Mich
ael Tung (aka \"Diffbot Mike\") says the new funding will be put towards new hires and expandi
ng its resources. "More than that, we're receiving a huge vote of confidence from veterans who
have built massive companies and understand the fine points of building for scale, maintaining
uptime and delivering the absolute highest standards of service."\nTung is a patent attorney a
nd Stanford PhD student who left the doctoral program to pursue Diffbot, thanks to seed fundin
g from Stanford's incubator, StartX . Diffbot was StartX's first investment. With today's fund
ing, Diffbot total raise is $2 million and change.",
      "title": "Diffbot Raises $2 Million Angel Round For Web Content Extraction Technology",
      "images": [
        {
        "primary": "true",
        "url": "http://tctechcrunch2011.files.wordpress.com/2012/05/diffbot_9.png?w=300"
        }
      ],
      "date": "Thu, 31 May 2012 07:00:00 GMT"
    }
}
```


Figure 9. The extracted API Article as requested with a Crawlbot, using instructions provided by Diffbot [31].

5.2 Wikidata

Since its inception in late 2012, Wikidata [33] has become one of the largest and most prominent collections of open data on the web. Its success was facilitated by the proximity to its big sister Wikipedia, which has supported Wikidata both socially and technically, e.g., with reliable server infrastructure and global user management. Wikidata thereby has grown into one of the largest public collections of general knowledge, consisting of more than 400 million statements about more than 45 million entities. These figures still exclude over 60 million links from Wikidata entities to Wikipedia articles (familiar from Wikipedia's Languages toolbar), over 200 million labels and aliases, and over 1.2 billion short descriptions in several hundred languages. Wikidata thus has become the central integration point for data from all Wikipedia editions and many external sources, an authoritative reference for numerous data curation activities, and a widely used information provider. Applications range from user-facing tools such as Apple's Siri or EuroWings' in-flight app to research activities, e.g., in the life sciences [34] and in social science [35]. First and foremost, just like its predecessor Wikipedia, the actual data of wikidata must be editable or extendable by any user, even for users without accounts. Aside from the data itself, the schema of the data also must be editable and controllable by all users. The second design characteristic is the notion of plurality. Some of the data present conflicts with other data when it comes to certain subjects, since some of the subjects can be not so certain or disputable.

On section 4.2 we have already been introduced to a brief history of Wikidata, and we already are aware that Wikidata is a document-oriented database, focused on items, which represent topics, concepts, or objects [36]. In order to review Wikidata a product, this section will further examine the technical part of it. According to Denny Vrandecic [33], for every Wikipedia article, a page has been created on Wikidata where links to related Wikipedia articles in all languages are managed. Such pages on Wikidata are called *items.* Another key design concept is the concept of multilinguality. Not all data is connected to a certain language. Each item is identified by a unique number, prefixed with the letter Q, known as a "QID". Wikidata is meant to be multilingual by design. Allowing the data to be easily accessible is also one of the most important design decisions of wikidata. The data and information are meant to be used both by Wikipedia and by third party applications, in the web or everywhere. The data is meant to be easily exportable in standard formats such as JSON so that it can easily be used by external applications [33]. In the tables below

we have summarized a list of Wikidata's main features, the pricing plan and other relevant technical details.

No.	Features	Description
1	Open Editing	Wikidata allows every user of the site to extend and edit the stored information, even without creating an account. A form-based interface makes editing very easy.
2	Community Control	Not only the actual data but also the schema of the data is controlled by the contributor community.
3	Plurality	Wikidata allows conflicting data to coexist and provides mechanisms to organize this plurality.
4	Secondary Data	Wikidata gathers facts published in primary sources, together with references to these sources.
5	Multilingual Data	Most data is not tied to one language: numbers, dates, and coordinates have universal meaning.
6	Easy Access	Wikidata's goal is to allow data to be used both in Wikipedia and in external applications. Data is exported through Web services in several formats, including JSON and RDF. Data is published under legal terms that allow the widest possible reuse.

Table 5. An overview of features of Wikidata, gathered from [37].

No.	Pricing Plan	Price
1	No name	✓ Free for all

Table 6. An overview of pricing plan of Wikidata, gathered from [37].

In terms of pricing, wikidata is free of use for any type of costumer, be it in developer mode or as a random human scrolling the internet. In addition to the information above, it is important to add that the devices supported by Wikidata are: Windows, Linux and Mac, and one of its most important features is its ability to embrace languages, meaning that it supports 358 languages. Currently wikidata data model has 16 million entities, 34 million statements and 80 million labels [38].

5.2.2 Generated query from Wikidata

Since mid 2015, Wikimedia provides an official public Wikidata SPARQL query service (WDQS) at [http://query.wikidata.org/], built on top of the BlazeGraph RDF store and graph database [39].In this section, we are using an example of a query to determine how wikidata SPARQL generates the query and what results it gets. In this case, we have used [http://query.wikidata.org/], to start a new request. Steps to run the request are seen on the figure below.

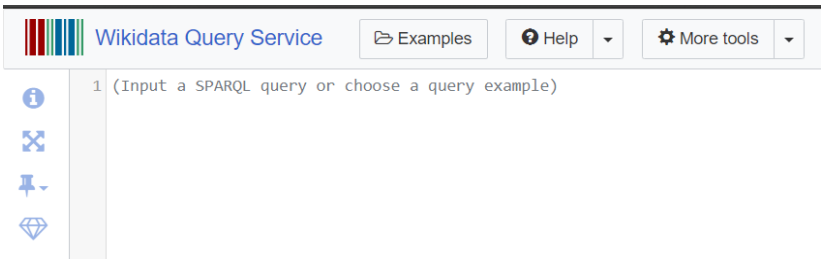

Figure 10. Wikidata SPARQL Query Service with an empty query platform, taken from [http://query.wikidata.org/] [40].

To elaborate further our query, we have used a query requesting information regarding the largest cities in the world. The query is as follows:

Figure 11. Wikidata SPARQL Query Service with an example query before rendering the results, taken from [http://query.wikidata.org/] [40]

5.2.3 Example response from Wikidata

The Wikidata KG is internally stored in JSON format and edited by users through custom interfaces. The response can be retrieved as JSON, HTML, PHP and CVS file. In this case, the response has been retrieved in a visual form, including: Bubble Chart (Figure 12), Bar Chart (Figure 13), and Table Content (not included due to repetitive information). The complete query response in PHP format is included in section 8.2 as Appendix B.

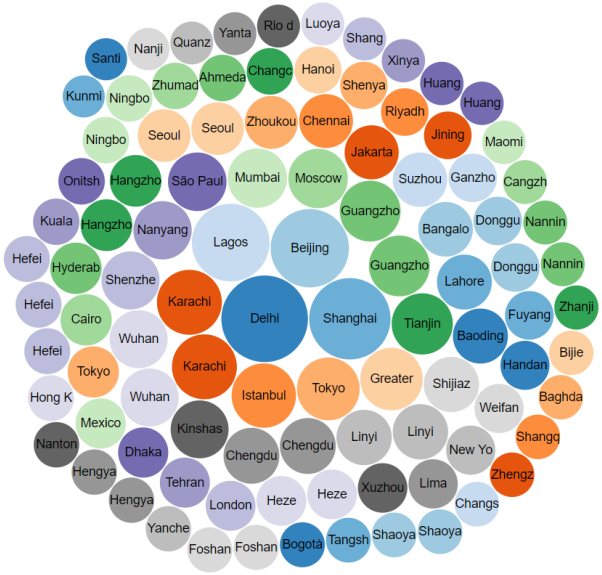

Figure 12. Bubble Chart response for the query of The Largest Cities in the world, retrieved by [http://query.wikidata.org/] [40].

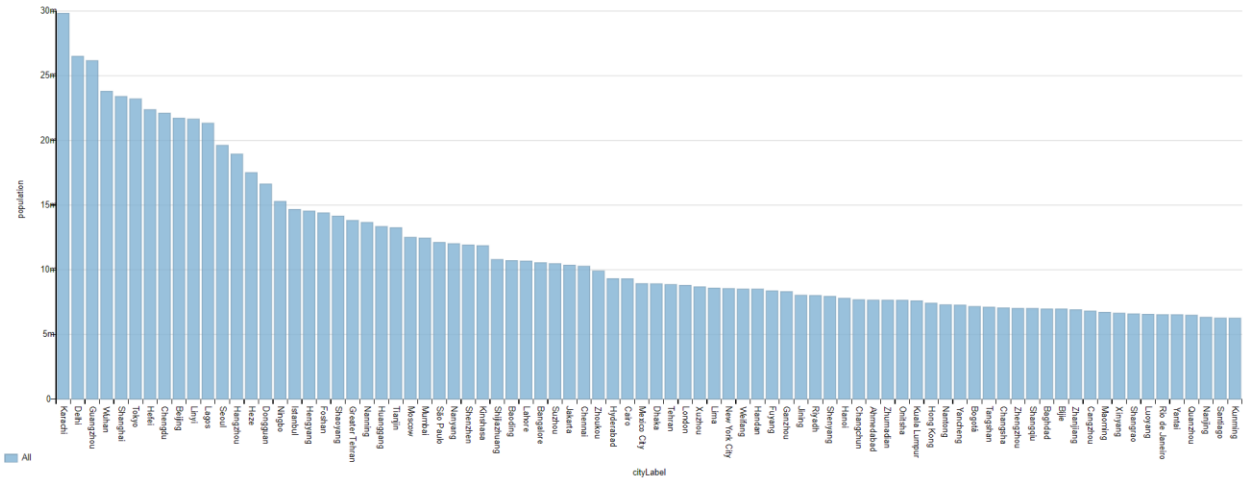

Figure 13. Line Chart response generated automatically by [http://query.wikidata.org/] [40].

5.3 IBM Watson

Named after the founder of IBM Thomas J. Watson, the IBM Watson initially began in 2008 as a project to build a super-computer which would utilise a combination of technologies like artificial intelligence, machine learning, web data extraction, enhanced analytical algorithms, and knowledge graphs to achieve a brand of "cognitive computing" in order to function as essentially a question answering bot. When an AI like Watson is compared to a classic personal assistant AIs like Microsoft's Cortana or Apple's Siri, the key difference to pinpoint is that the latter are essentially search engines with speech recognition capabilities, that can query specific back-end providers like Wikipedia or Google Maps. By contrast, Watson utilises its "cognitive computing" capabilities combined with text mining and deep analytical algorithms to provide insight from immense amounts of unstructured data and translate that data into speech and other forms of communication [41]. Watson is already in use in many different real-world fields like business insights, statistics, healthcare, education, banking, online marketplaces, geolocation and navigation, automated costumer care, even criminology and forensics. Some of the major companies that use IBM Watson are *Autodesk;* the 3D modelling and design software giant has incorporated the IBM Watson in order to basically replace customer service operators. They began utilising Watson as an artificial intelligence "chatbot" called Ava, which is able to resolve around 15.000,00 customer service calls per month and has swifted the customer service by 100% [42].*Wimbledon*; started utlising the visual capabilities of Watson in order to create an AI that can automatically generate sporting events highlights, essentially replacing a human video editor. The intelligent system will generally gather the clips based on factors like crowd reaction, facial recognition of the players, and other events happening right after a point has been won, in order to complete a highlight reel of the action in record time [43]. *General Motors*; In 2016, GM decided to explore the capabilities of AI using the help of Watson. Together with IBM, they developed a system to analyse user preferences and patterns when travelling by car, in order to provide special and personalised marketing and location-based services to drivers [44]. *American Cancer Society;* ACS has had a collaboration with Watson in order to create a "virtual adviser" which utilises machine learning with the aim of providing patients with personalised advice. The virtual advisor will extract information such as the cancer type, the stage, and previous therapies and treatments using natural language, and will provide spoken answers and advice in return [45]. *Under Armour;* The giant sportswear brand has tapped into the capabilities of Watson in order to create

personalised fitness apps. The app uses Watson to extract physiological, psychological, and nutritional data in order to provide suggestions for the optimal fitness plan for its users [46].

On section 4.3 we have been introduced to e brief history of IBM Watson and its journey through years in creating and expanding its concept into many other fields of businesses. Further we are going to analyze the technical part of it. According to [https://reviews.financesonline.com], the IBM Cloud Platform is able to consume and comprehend a variety of data types, allowing businesses to extract insights that are meaningful and actionable. The platform offers industry-leading AI workload optimization and is able to comprehend a variety of data types. As done on section 4.1 and 4.2 for Diffbot and Wikidata, we are going to present a table with the technical details of IBM. In the tables below, we will examine its main features, pricing plans, device support and other information.

No.	Features	Description
1	Natural Language Dialogue	This system has a wide range of language processing techniques to have efficient conversional partners.
2	Simplified Analysis	IBM Watson helps to quickly understand what your data wants to convey to you by using automation. The automation does all the hard work so that you can spend less time in analyzing your data using new and unexpected insights to your business.
3	Automated Predictive Analytics	It is a service in which a data owner upload and build predictive or descriptive models with minimum data. IBM Watson provides you with automated predictive analysis service that automatically surfaces the driving outcomes.
4	Speech-to-text-to-speech	It is also known as a speech recognition system. This feature can easily convert the audio or voice into written text for quick understanding. It is the opposite to speech to text. It converts written text into audio in a variety of languages and voice. It enables your system to speak like a human being.
5	Visual Recognition	This feature of IBM Watson allows you to analyze the visual content of images and videos using machine learning.
6	Trade-Off Analytics	This feature of IBM Watson helps businesses to make decisions by balancing multiple objectives. With the help of trade-off analytics, you can avoid unnecessary options and determine the right options from multiple objectives.

Table 7. An overview of features of IBM Watson, gathered from [https://www.newgenapps.com/] [47].

No.	Pricing Plan	Price
1	Plus	$30/user/month
• Single user • 2GB storage • On premises and cloud relational databases • 18 data connectors • IBM Analytics Exchange data and more		
2	Professional	$80/user/month
• Plus features and • 1+ users • 100GB storage • 19 data connectors (including IBM Cognos reports)		

Table 8. Offered Pricing Plans as monthly-basis subscription plans according to the business needs [47].

In addition to the information above, it is important to add that the devices supported by IBM are: Windows, Linux and Mac, in only the English language. According to [47] attention to IBM is given by Small Businesses, Large Enterprises and Medium Businesses.

4.3.2 Generated query from IBM Watson

One of the main features of the IBM is that it offers powerful content search capabilities through queries. According to [https://cloud.ibm.com], after our content is uploaded and enriched by Discovery, we can build queries, integrate Discovery into our own projects, or create a custom application by using the Watson Explorer Application Builder. There are a lot of methods of conducting a query, depending on the field you want to enquire (Basic Query, Combined Query, Aggregation, etc). These functionalities depend on the type of services that you are subscribed to. Since the service is not free for us, we are using an online-provided example to showcase how a basic query would look like through IBM. According to their official website, the IBM Knowledge Graph allows entities and relationships to form while querying. The Knowledge Graph relations query JSON object is seen below. A copy of the query response is enclosed on Appendix C on Section 8.3. If no match is found, the JSON object is returned to as per below [48].

```
{
  "relations": []
}
```
The <relations> enrichments must be specified as follows:
```
"relations": {
  "model": "en-news"
}
```
The <entities> enrichment must be specified as follows and must also have the <mentions>, <mentions_types>, and <sentence_locations> parameters specified:
```
"entities": {
  "mentions": true,
  "mention_types": true,
  "sentence_locations": true,
  "model": "en-news"
}
```
An entity query is performed by <post>ing a JSON object to the <v1/environments/{environment_id}/collections/{collection_id}/query_relations> endpoint.
```
{
  "entities": [
    {
      "text": "Steve Jobs",
      "type": "PERSON",
      "exact": true
    }
  ],
  "context": {
    "text": "iphone"
  },
  "sort": "score",
  "filter": {
    "relation_types": {
      "exclude": ["colocation"],
      "include": ["locatedAt", "employedBy", "managerOf", "founderOf"]
    },
    "entity_types": {
      "exclude": ["EVENT"],
      "include": ["PERSON", "GPE", "ORGANIZATION"]
    },
    "document_ids":  ["b95df4c1-d00f-4771-abb2-a52baea0444a",  "ad340635-bf3e-47a5-bea5-5e778f600c32"]
  },
  "count": 10,
  "evidence_count": 0
}
```

6. COMPARISION OF SELECTED KNOWLEDGE GRAPH TECHNOLOGIES

On Chapter 5, we have conducted a systematic overview of the selected KGs in their current versions and discussed how the knowledge in these KGs is modeled, stored, and can be queried. Further, after these KGs were analyzed in-depth, a comparison method is used in order to determine the quality of each KG, in terms of suitability in product use (part of third objective). We are going to elaborate this further through our comparison methodology. On the table below, we will compare several key elements of each KG, starting from main features each of them portrays, their prices, what devices to they support etc.

No.	Components	Diffbot	Wikidata	IBM Watson
1	Features	✓ AI-Powered Web Data extraction ✓ Extracts clean, normalized and structured data ✓ Requires no rules and training ✓ Article API ✓ Analyze API ✓ Image-Video API ✓ Comprehensive Knowledge Graph ✓ Web Crawling Tool	✓ Open Editing ✓ Community Control ✓ Plurality ✓ Secondary Data ✓ Multilingual Data ✓ Easy Access	✓ Domain Specific Research ✓ Interraction enrichment ✓ Adaptive costumer experiences ✓ Personalized communication ✓ System conditioning monitoring ✓ Targeted recommendation ✓ Risk Mitigation ✓ Chatbots ✓ Knowledge Management ✓ Visual Recognition
2	Pricing Model	✓ From $299 to $3,999 per month ✓ According to enterprise needs	✓ Free for all users	✓ From $30 to $80 per month ✓ According to solution needs
3	Language Supported	✓ English ✓ Chinese ✓ German ✓ French ✓ Spanish ✓ Russian	✓ Available in 358 languages	✓ English
4	Prominent Clients	✓ Yandex ✓ Ebay ✓ Salesforce	✓ Public domain for the people	✓ Staples ✓ Korean Air ✓ Autodesk ✓ Wimbledon

5	Integrations	✓ Wordpress ✓ Youtube	✓ Wikipedia ✓ Wikimedia ✓ Wikiweb ✓ WikiBOT	✓ Facebook ✓ Slack ✓ Intercom
6	Available Devices	✓ Windows ✓ Linux × Android × Iphone/Ipad ✓ Mac ✓ Web-based × Windows mobile	✓ Windows ✓ Linux ✓ Android ✓ Iphone/Ipad ✓ Mac ✓ Web-based ✓ Windows mobile	✓ Windows ✓ Linux × Android × Iphone/Ipad ✓ Mac ✓ Web-based × Windows mobile
7	Company Size	✓ Small Business ✓ Large Enterprise ✓ Medium Business × Freelancers	✓ Small Business ✓ Large Enterprise ✓ Medium Business ✓ Freelancers	× Small Business ✓ Large Enterprise ✓ Medium Business × Freelancers
8	Available Support	✓ E-mail ✓ Phone ✓ Live support ✓ Training ✓ Tickets	✓ Website	✓ E-mail ✓ Phone × Live support ✓ Training ✓ Tickets

Table 9. Comparison table for all selected KGs; Diffbot, Wikidata and IBM Watson.

According to the comparison table above, it is yet difficult to determine which KG might be the most suitable and effective to use. At the end of the day, that depends on individual preferences. However, keeping in mind that online sources are available to review these back-to-back with each other, we have also used that as a factor influencing on our result. The information that lead up to this thesis has proven that the efficiency of each depends also on costumer feedback. Pricing plans are almost as crucial as features and user support qualities, and while cost should not be a sole aspect, its without a doubt a key thing to think about. When trying to shop for the most suitable KG, its without a doubt that we expect a flexible pricing package that can be matched with our teams' size and easily scaled up when the team grows. Therefore, it would be a useful experience to try and experiment with a free trial of each solution before engaging in bigger decisions. And in such a digital era, it has become even easier to optimize our business performances in the way that we structure and view our data. As a result, this thesis has shown that all three of them are individualistic in a decision point of view, however we would leverage with whichever product we choose to go with, in order to improve your practical operation.

7. Conclusion

In this study, we have have elaborated the notion of Knowledge Graphs and its first existence, with the main aim of trying to understand how this big spectrum comes into life, in terms of how it is incorporated into a product. In order to try and understand better how the KGs are fitted for product use, we have analyzed three selected KGs; Diffbot, Wikidata and IBM Watson. The reason why these KGs were selected is because of the broad and unique range of features all of them offer, and because the three of them are not entirely the same with one another. Through this study we have come to realise that the enormous amount of data which is found in the Web, can be structured and treated differently by each person, depending on what the intentions are. We have also come to the conclusion that the discovery and analysis of the data can be done in several ways and all of them can be efficient without the help of a professional data analyst, meaning that, we as humans, can interact with data by using the cognitive tools of these KGs, which have capability to work in a human language environment. In terms of business, users can determine a trend and visualize data reports for business outcomes. One of the main lessons learned during this study, was that while examining different alternatives for KGs and the way they are outlined as products, we ought to pay attention not just to functionalities but also to a wide range of aspects like price, quality of costumer support, supported mobile devices and available integrations. With enough knowledge we should be able to find a solution that is going to have all the elements we need at an inexpensive pricing. With this study, we have tried to showcase each product differently, in order to see how they are very different from each other, but in the same time they match their fitting on what we expect from them. It might in some cases be a true challenge to locate a trustworthy Artificial Intelligence Software app that will not only fit our needs but will also be in accord with our budget limits and with sufficient research we should be able to locate a service that is going to have all the variables we need at a reasonable cost.

8. References

[1] NILS J. NILSSON, The Quest for Artificial Intelligence, Stanford University, October 2009

[2] A. M. TURING, Computing Machinery and Intelligence, Vol. 59, No. 236, October 1950

[3] JOHN MCCARTHY, Computer Science Department, Stanford University, November 2007

[4] SHUBHAM PANCHAL, Types of Artificial Intelligence and examples, August 2018

[5] RACHEL SCHUTT, CATHY O'NEIL, Doing Data Science, O'Reilly Media, October 2013

[6] ALEXANDER KONOVALOV, Predicting Knowledge Base Revisions from Realtime Text Streams, The Ohio State University, 2018

[7] VOJTECH DRAXL, Web Scraping - Data Extraction from websites, February 2018

[8] C. OLSTON AND M. NAJORK, Web Crawling Information Retrieval, Vol. 4, No. 3, Pg 175–246, 2010

[9] AYOUB MOHAMED H. ELYASIR, KALAIARASISONAIMUTHU ANBANANTHEN, Web Crawling Methodology, 2012

[10] TRUPTI V. UDAPURE, RAVINDRA D. KALE, RAJESH C. DHARMIK, Study of Web Crawler and its Different Types, Volume 16, Issue 1, Ver. VI, Feb. 2014

[11] AKSHADA K. DHAKADE, Web Crawler: Essential Component of Search Engine, 2013

[12] MICHELANGELO DILIGENTI, FRANS COETZEE, STEVE LAWRENCE, C. LEE GILES, MARCO GORI, Focused Crawling using Context Graphs, 2000

[13] DHIRAJ KHURANA, SATISH KUMAR, Web Crawler: A Review, Vol. 12, Issue 01, January 2012

[14] ANKUR M. PATWA, Design and Implementation Of A Parallel Crawler, 2006

[15] NO AUTHOR (Public to the Web), Practical Web Scraping for Data Science: Best Practices and Examples with Python - Apress; 1st ed. Edition, April 2018

[16] ARPAN JHA, Web Crawling: Data Scraping vs. Data Crawling, May 2012

[17] FAVIO VÁZQUEZ, Understanding Graph Databases, n.d

[18] MICHAEL HUNGER, The Definitive Guide to Graph Databases, April 2015

[19] AMIT SINGHAL, Introducing the knowledge graph: things, not strings, May 2012

[20] JIM WEBBER, Why Knowledge Graphs Are Foundational to Artificial Intelligence, 2018

[21] JAMES P. MCCUSKER, MICHEL DUMONTIER, RUI YAN, SYLVIA HE, JONATHAN

S. DORDICK, DEBORAH L. MCGUINNESS, Finding melanoma drugs through a probabilistic knowledge graph, February 2017

[22] TREY GRAINGER, KHALIFEH ALJADDA, MOHAMMED KORAYEM, AND ANDRIES SMITH, The Semantic Knowledge Graph: A Compact, Auto-Generated Model for Real-Time Traversal and Ranking of any Relationship within a Domain, September 2016

[23] ANDRÉ FREITAS, FABRÍCIO F. DE FARIA, SEÁN O'RIAIN, EDWARD CURRY, Answering Natural Language Queries over Linked Data Graphs: A Distributional Semantics Approach, 2013

[24] DANIEL FAGGELLA, EMERJ ENTERPRISE, What is Machine Learning, Updated February 2019

[25] ALEX SMOLA AND S.V.N. VISHWANATHAN, Introduction to Machine Learning, 2008

[26] NATASHA NOY, YUQING GAO, ANSHU JAIN, ANANT NARAYANAN, ALAN PATTERSON, AMIE TAYLOR Industry-scale Knowledge Graphs; Lessons and Challenges, April 2019

[27] MICHAEL FÄRBE, BASIL ELL, CARSTEN MENNE, ACHIM RETTINGER, AND FREDERIC BARTSCHERER, Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO Semantic Web, 2016

[28] NO AUTHOR (Public to the Web), Introducing the Diffbot Knowledge Graph

[29] NO AUTHOR (Public to the Web), Rivaling Google, Web-Mining Startup Diffbot

[30] NO AUTHOR (Public to the Web), Diffbot as a data extraction software

[31] NO AUTHOR (Public to the Web), Introduction to Diffbot

[32] NO AUTHOR (Public to the Web), [https://www.diffbot.com/]

[33] DENNY VRANDEČIĆ, MARKUS KRÖTZSCH, Wikidata: A free collaborative knowledge base, 2016

[35] WAGNER, C.,GRAELLS-GARRIDO, E., GARCIA, D., MENCZER, F.: Women through the glass ceiling: gender asymmetries in Wikipedia, 2015

[36] NO AUTHOR (Public to the Web), [https://en.wikipedia.org/wiki/Wikidata]

[37] PETER BUNEMAN, JAMES CHENEY, WANG-CHIEW TAN, AND STIJN VANSUMMEREN, Curated databases, 2008

[38] PETER HAASE, Querying the Wikidata Knowledge Graph, 2015

[39] STANISLAV MALYSHEV, MARKUS KRÖTZSCH, LARRY GONZÁLEZ, JULIUS GONSIOR AND ADRIAN BIELEFELDT, Getting the Most out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph, n.d
cfaed, TU Dresden, Dresden, Germany, firstname.lastname@tu-dresden.de
[40] NO AUTHOR (Public to the Web), [http://query.wikidata.org]
[41] THOMAS J. WATSON, Introduction to "This is Watson", IBM Journal of Research and Development, Volume: 56 , Issue: 3.4 , May-June 2012
[42] CHRISTIE SCHNEIDER, How Autodesk sped up customer response times by 99% with Watson, [Available at: https://www.ibm.com/blogs/watson/2017/10/how-autodesk-sped-up-customer-service-times-with-watson/]
[43] JEREMY KHAN, Game, Set, Machine Learning: How IBM Is Fine-Tuning Its Wimbledon Tech, [Available at: https://fortune.com/2019/07/11/wimbledon-2019-ibm-tech/]
[44] REUTERS, Here's What IBM Watson Will Be Doing in GM's Cars, [Available at: https://fortune.com/2016/10/26/gm-onstar-ibm-watson/]
[45] NO AUTHOR (Public to the Web), American Cancer Society and IBM Collaborate to Create Virtual Cancer Health Advisor, April 2016 [Available at: http://pressroom.cancer.org/WatsonACSLaunch]
[46] NO AUTHOR (Public to the Web), Under Armour And IBM To Transform Personal Health and Fitness, Powered By IBM Watson, January 2016, [Available at: https://www-03.ibm.com/press/us/en/pressrelease/48764.wss]
[47] NO AUTHOR (Public to the Web), IBM Watson and its Key Features, 2018, [https://www.newgenapps.com/blog/ibm-watson-and-its-key-features]
[48] NO AUTHOR (Public to the Web), [https://cloud.ibm.com/docs/services/discovery?topic=discovery-kg]

9. Appendices

9.1 Appendix A – JSON response from Section 4.1.3


```
{
  "request": {
    "pageUrl": "http://tcrn.ch/Jw7ZKw",
    "resolvedPageUrl": "http://techcrunch.com/2012/05/31/diffbot-raises-2-million-seed-round-for-web-content-extraction-technology/",
    "api": "analyze",
    "options": [],
    "fields": "",
    "version": 3
  },
  "objects": [
    {
      "type": "article",
      "resolvedPageUrl": "http://techcrunch.com/2012/05/31/diffbot-raises-2-million-seed-round-for-web-content-extraction-technology/",
      "pageUrl": "http://tcrn.ch/Jw7ZKw",
      "human_language": "en",
      "text": "Diffbot , the super-geeky/awesome visual learning robot technology which aims to \"see\" the web the way that people do, is today announcing a new infusion of capital. The company has closed $2 million in funding from a number of technology veterans, including EarthLink founder Sky Dayton ; Andy Bechtolsheim , co-founder of Sun Microsystems; Joi Ito , Director of MIT Media Lab; Brad Garlinghouse , CEO of YouSendIt ( and formerly of TechCrunch parent company AOL ), Maynard Webb , Chairman of the Board at LiveOps, formerly eBay COO; Elad Gil , VP of Corporate Strategy at Twitter; Jonathan Heiliger , former VP of Technical Operations at Facebook; Redbeacon co-founder Aaron Lee ; and founder of VitalSigns Montgomery Kersten .\nMatrix Partners also participated in the round. Of the new investors, Sky Dayton will be the first to join Diffbot's board and will be taking an active role in the company, including plans to go hands-on with various Diffbot projects.\nLast August, the company publicly debuted its first APIs , which allow developers to build apps that can automatically extract meaning from web pages. For example, the Front Page API is able to analyze site homepages, and understands the difference between article text, headlines, bylines, ads, etc. The Article API can then extract clean article text, images and videos. Another example of Diffbot in action is the \"follow API,\" which can track the changes made to a website.\nToday, Diffbot has categorized the web into about 20 different page types, including homepages and article pages, which are the first two types it can now identity. Going forward, Diffbot plans train its bots to recognize all the other types of pages,
```

including product pages, social networking profiles, recipe pages, review pages, and more.\nIts APIs have been put to use by AOL (again: disclosure, TC parent) in its news magazine AOL Editions , as well as by companies like Nuance , SocMetrics , and others. Diffbot says it's now processing 100 million API calls per month on behalf of its customers. Thousands of developers are using the APIs, the company notes, but paying customers are only in the \"tens.\" Correction: we're now told they have \"a lot more!\"\nDiffbot founder and CEO Michael Tung (aka \"Diffbot Mike\") says the new funding will be put towards new hires and expanding its resources. "More than that, we're receiving a huge vote of confidence from veterans who have built massive companies and understand the fine points of building for scale, maintaining uptime and delivering the absolute highest standards of service."\nTung is a patent attorney and Stanford PhD student who left the doctoral program to pursue Diffbot, thanks to seed funding from Stanford's incubator, StartX . Diffbot was StartX's first investment. With today's funding, Diffbot total raise is $2 million and change.",

"title": "Diffbot Raises $2 Million Angel Round For Web Content Extraction Technology",
"images": [
{
"primary": "true",
"url": "http://tctechcrunch2011.files.wordpress.com/2012/05/diffbot_9.png?w=300"
}
],
"date": "Thu, 31 May 2012 07:00:00 GMT"
}
}

9.2 Appendix B – PHP response from Section 4.2.3

```
<?php
class SPARQLQueryDispatcher
{
    private $endpointUrl;
    public function __construct(string $endpointUrl)
    {
        $this->endpointUrl = $endpointUrl;
    }
    public function query(string $sparqlQuery): array
    {
        $opts = [
            'http' => [
                'method' => 'GET',
                'header' => [
                    'Accept: application/sparql-results+json'
                ],
            ],
        ];
        $context = stream_context_create($opts);
        $url = $this->endpointUrl . '?query=' . urlencode($sparqlQuery);
        $response = file_get_contents($url, false, $context);
        return json_decode($response, true);
    }
}
$endpointUrl = 'https://query.wikidata.org/sparql';
$sparqlQueryString = <<< 'SPARQL'
#Largest cities of the world
#added before 2016-10
 #defaultView:BubbleChart
SELECT DISTINCT ?cityLabel ?population ?gps
WHERE
{
  ?city wdt:P31/wdt:P279* wd:Q515 .
  ?city wdt:P1082 ?population .
  ?city wdt:P625 ?gps .
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en" .
  }
}
ORDER BY DESC(?population) LIMIT 100
SPARQL;
$queryDispatcher = new SPARQLQueryDispatcher($endpointUrl);
$queryResult = $queryDispatcher->query($sparqlQueryString);
var_export($queryResult);
```

9.3 Appendix C – JSON response from Section 4.3.3


```
{
  "relations": [
    {
      "type": "FOUNDEROF",
      "frequency": 7,
      "arguments": [
        {
          "entities": [
            {
              "type": "PERSON",
              "text": "Steve Jobs"
            }
          ]
        },
        {
          "entities": [
            {
              "type": "ORGANIZATION",
              "text": "Apple"
            }
          ]
        }
      ]
    }
  ]
}
```