Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és MédiaInformatikai Tanszék

Marouf Dániel

ANALYZING SEARCH ALGORITHMS FOR GRAPH-BASED KNOWLEDGE REPRESENTATIONS

KONZULENS

Tapolcai János Dr.

BUDAPEST, 2018

Contents

Abstract

The graph database is suitable for storing millions of nodes and edges. However its rather difficult to see through even a graph with twenty nodes and connections between them. It makes the search and the managing of the graphs difficult and complicated.

I would like to show the process of the system at the TDK. I am going to choose the key points and related topics. I want to emphasize that if it works like that here, it will work on the research of the professors in a different system. This kind of tool has been called ScienceMesh.In my work,

I am going to use SienceMesh as the name of the process. This new scientific point of view will help to improve in all field of science. As result, remarkable change will happen to the human's way of thinking. The question is why this has never happened yet. We have used the graphs in many fields of science but we have never used in everyday life.

Kivonat

A gráfadatbázisok alkalmasak több milliós méretű gráfok tárolására, ezzel szemben húsz csomópontos gráf átlátása is nehéz egy ember számára. Ez az ellentmondás nehézkessé teszi a gráfadatbázisok ellenőrzését, keresést bennük, és úgy általában a használatát.

A gráf vizualizációval egy adott felületen több találatot tud megjeleníteni és a kapcsolatokat is megtalálja közöttük. Így ha egy cikkre akarunk rákereseni akkor a kapcsolodó cikkek is egyből megjelennének és rálátást nyernénk a kutatási terület többi cikkére is.

A publikáció során a TDK-án fogom bemutatni a gráf vizualizáció módszertanát. Az eddig feltöltött publikációk és adatok felhasználásával egy gráf adatbázist építek és mutatom be rajta a módszer előnyeit. Ennek a folyamatnak a ScienceMesh elnevezést adtam. A gráfokat szinte minden kutatási területen használjuk, de a mindennapi életben még nem történt meg az áttörés. A keresés gráf vizualizációja első sorban most kutatóknak készült, de a jövőben mindenki számára alkalmas lehet a gráfok adta információ többlet megértésére.

1 Introduction

In 1735 Konner Euler[1] published the Seven Bridges of Konigsberg and became the first person who wrote about how graph helps to understand problems. He started to build a new tool case to see the world in a different aspect.

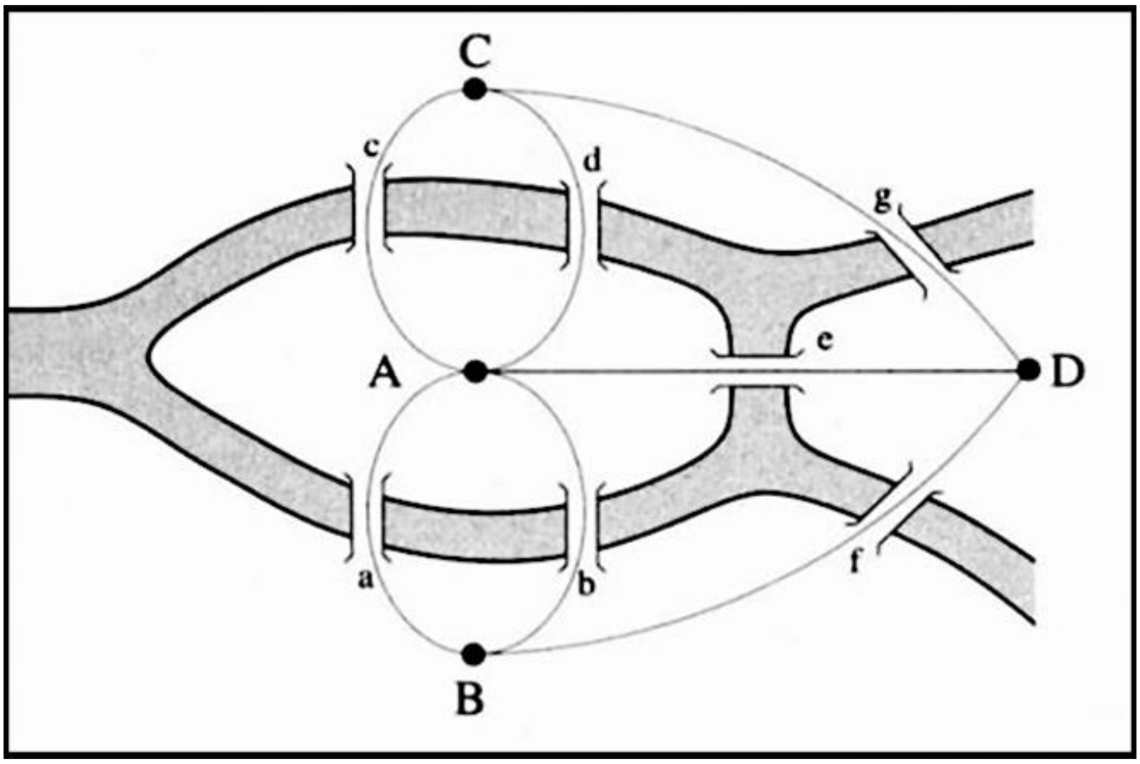


Figure 1.1 The Seven Bridges of Koningsberg

Since 1735, there have been many advances in the field of graph theory and topology. With a rigorous foundation for the field being built shortly thereafter, today's graph theory has grown to be quite broad in scope. Focusing only on the practical applications, we can see that there are many domains where the understanding of graphs and graph algorithms are vital to answering real business questions.

Probably the biggest known application of graph theory is Social Network Analysis or SNA for short. SNA is the study of social networks, their structure and how knowing this structure can lead to better understanding of behavior within social networks. The most well-known social network is, of course, Facebook.

Figure 1.2 Community of Facebook

There are many tools freely available on Facebook that allow you to study your own social network structure and see patterns within your list of friends.

More than simply being a tool to show social networking structure, graphs are used to predict what your interests are. These predictions are based not only on your own behavior within a social network but also on the behavior of your friends

When I was an undergrad taking a graph theory course, I was assigned an interesting project. The client was a hypothetical airline who needed a new system for finding flights for customers. This airline had planes flying between 18 cities. They needed a way to take customers from one city to another while ranking them by one of three conditions: least number of layovers, lowest cost or earliest arrival time. Having cities represented by nodes and flights between cities being represented by edges, we can see that this problem is merely finding the most efficient path based on which of the three conditions we want to be satisfied. This relatively simple problem has a very straightforward solution, provided you know about Dykstra’s algorithm.

Ever wonder where your computer network is most vulnerable or susceptible to failure? Well, computer networks are probably the easiest thing to represent as a graph since networking pretty much uses graphs to represent the layout of any computer network.

It’s easy to see that your graph will have nodes consisting of all the physical connections within a network (computers, routers, hubs, switches, bridges, etc.) and the edges in the network will be the actual connections (wireless or hardwired). Having a simple weighted network graph (weights on edges representing load capacities), we can determine if there are vulnerabilities within a network.

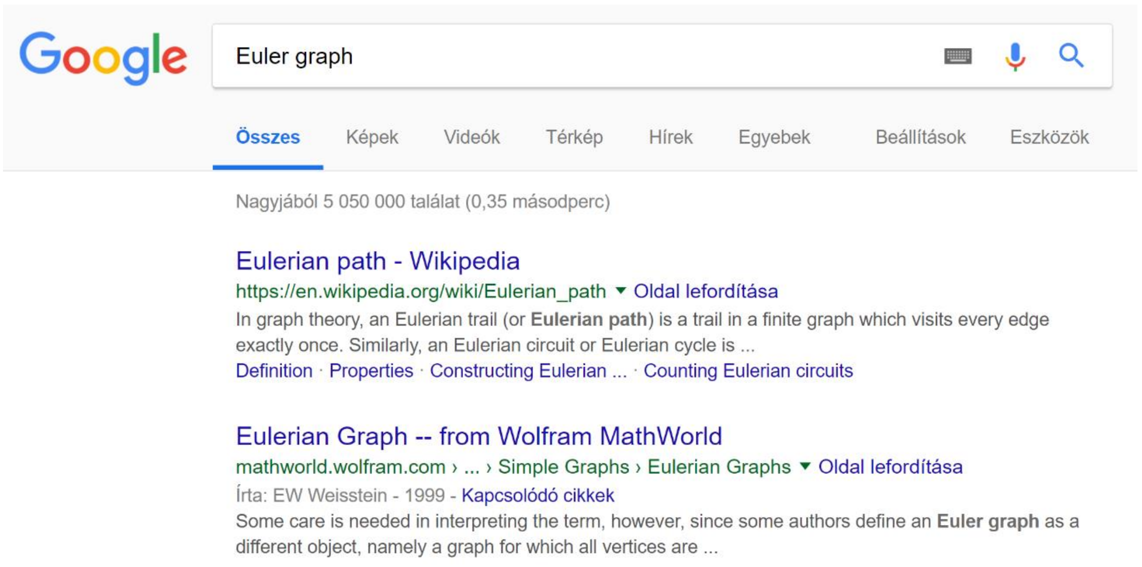
The question is why we stuck using only in science. Why don't we use in all fields of life starting with search?

1.1 Search systems today

We usually use keyword-based search to find expressions on the internet. This is how Google, Bing and our mentality work. We type something into and we want to browse the results to find the most specific answer for our keyword.

1.1.1 An illustrative example

Let's say we want to search the publication of Euler. We will type "Euler graph", which matches. Millions of results that is showed in an order given by an algorithm.



Nagyjából 5 050 000 találat (0,35 másodperc)

Eulerian path - Wikipedia
https://en.wikipedia.org/wiki/Eulerian_path ▾ Oldal lefordítása
In graph theory, an Eulerian trail (or **Eulerian path**) is a trail in a finite graph which visits every edge exactly once. Similarly, an Eulerian circuit or Eulerian cycle is ...
Definition · Properties · Constructing Eulerian ... · Counting Eulerian circuits

Eulerian Graph -- from Wolfram MathWorld
mathworld.wolfram.com › ... › Simple Graphs › Eulerian Graphs ▾ Oldal lefordítása
Írta: EW Weisstein - 1999 - Kapcsolódó cikkek
Some care is needed in interpreting the term, however, since some authors define an **Euler graph** as a different object, namely a graph for which all vertices are ...

Figure 1.3 Keyword based search

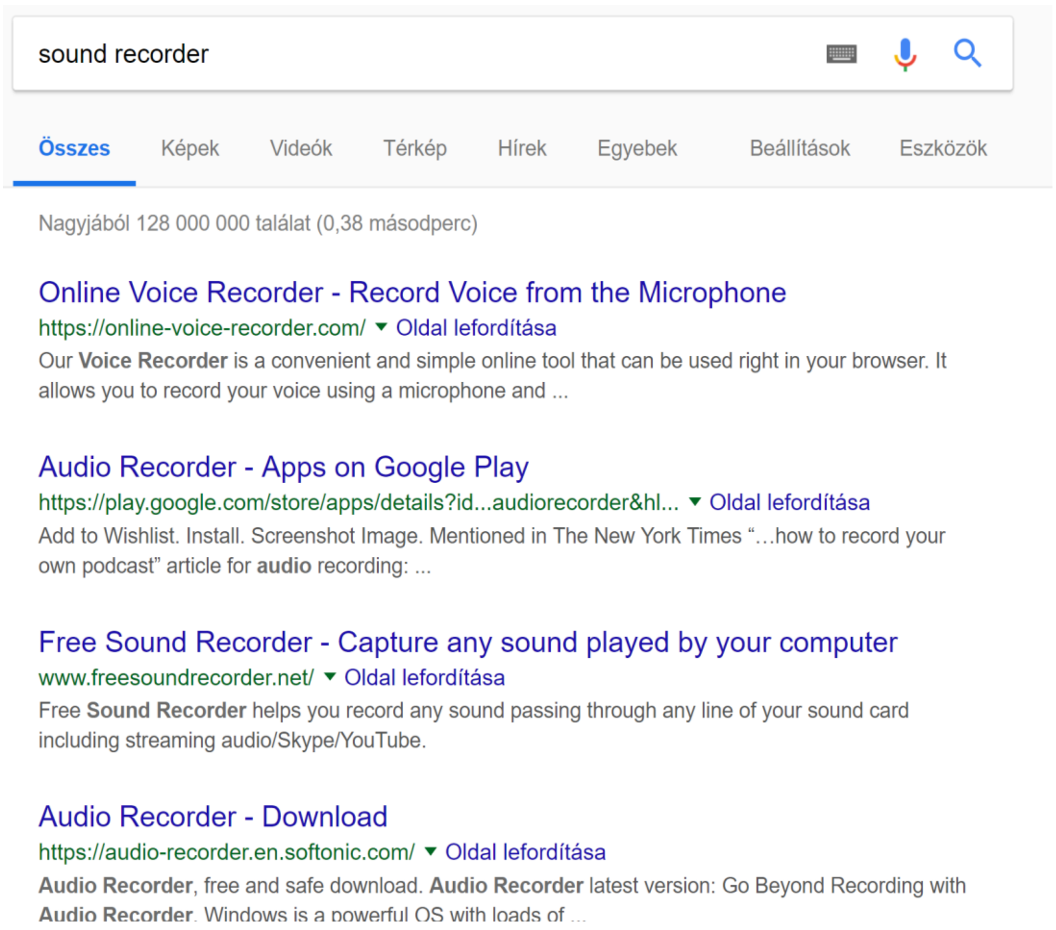
It will contain the publication, maybe it is the first result or the third one. However, the human brain works in a slightly different way, it is not only interested in the specific answer but tries to find a connection to some related knowledge. For example, probably we will get interested in Euler's life as well. Should google list a reference about Euler's life in the search result? Currently, for one search we only get the information which is the closest to the search word.

Let us give a simple intuitive example of how our brain can process information. Assume we want to find a pencil on a table. Here, a Google search result can be interpreted as listing the coordinates of the three pens on the table. However, showing a picture of the table would be way more information to a human than a result list. A human can immediately identify the three pens on the table, and can easily pick up one. Our brain can process a lot of complex graphical information which should be better utilized in presenting the search results.

Whoever, as soon as we find our pencil we will know where the pencil was and what on the range of the pencil was. But Google only tells us the exact location of all pencils but that’s all it can do. Our brain can understand more than the google results.

1.1.2 Several results same information

Most of the search results bear the same meaning but they are from a different source. In most case, the user will choose the first option, which is mostly a good pick but you used only a small fraction of the information you can see on the first page. The high-level idea of the study to use a graph to represent the results which can help us to see several closely related content to the search phrase that gives a more comprehensive answer to the user’s query.



Nagyjából 128 000 000 találat (0,38 másodperc)

Online Voice Recorder - Record Voice from the Microphone
https://online-voice-recorder.com/ ▾ Oldal lefordítása
Our **Voice Recorder** is a convenient and simple online tool that can be used right in your browser. It allows you to record your voice using a microphone and ...

Audio Recorder - Apps on Google Play
https://play.google.com/store/apps/details?id...audiorecorder&hl... ▾ Oldal lefordítása
Add to Wishlist. Install. Screenshot Image. Mentioned in The New York Times “…how to record your own podcast” article for **audio** recording: ...

Free Sound Recorder - Capture any sound played by your computer
www.freesoundrecorder.net/ ▾ Oldal lefordítása
Free **Sound Recorder** helps you record any sound passing through any line of your sound card including streaming audio/Skype/YouTube.

Audio Recorder - Download
https://audio-recorder.en.softonic.com/ ▾ Oldal lefordítása
Audio Recorder, free and safe download. **Audio Recorder** latest version: Go Beyond Recording with **Audio Recorder**. Windows is a powerful OS with loads of ...

Figure 1.4 Google results

1.2 Advantages of graph visualization

In graphs nodes and edges are drawn, where an edge connects two nodes. When the graph is an outcome of a search query nodes represent different contents related to the search result.

There are many ways to present data to your users, but the node-link structure instantly makes sense, even to people who've never worked with graphs before. With graph visualization, it's easy to explore relationships and uncover trends, insights and network structures in graph data. This means your analyst tools can be used by the widest possible audience. The node-link model of graph visualization can be applied to pretty much any dataset. As long as there's an interesting relationship in your data somewhere, you'll find value in graph visualization.

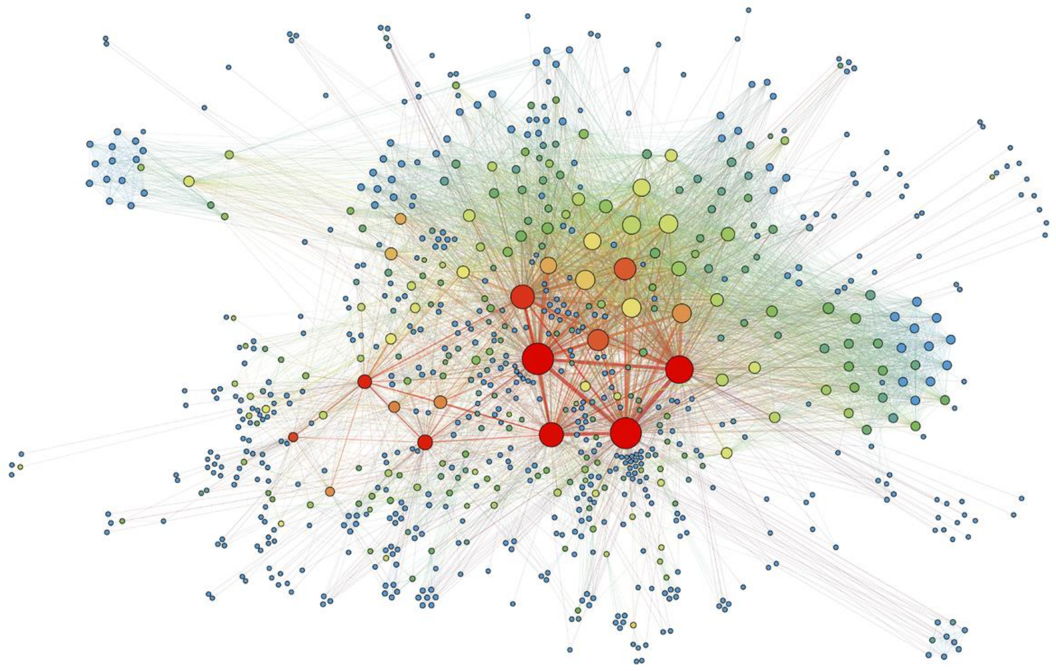


Figure 1.5 An example: Bitcoin transactions

Whether your users need to visualize email communication, credit card fraud, IT networks, or data that isn't inherently 'graph'[2]– graph visualization can help. It brings a new dimension to any dataset, revealing insight that would otherwise go undiscovered. Our brains struggle with processing data but are great at spotting patterns if the data is presented in a tangible format. Armed with visualizations, we can spot trends and outliers very effectively. Integrating graph visualization into your web application will help your users gain insight into their data quickly. Analysts often need to make fast decisions, based on a deep understanding of data. Graph visualization empowers them to do that with confidence. Exploring network data interactively allows users to gain deeper knowledge, understand the context and ask more questions, compared to static visualization.

1.3 ScienceMesh

We have also a university project called ScienceMesh. In the future, we want to use graph visualization in a bigger database where the professors and publications can

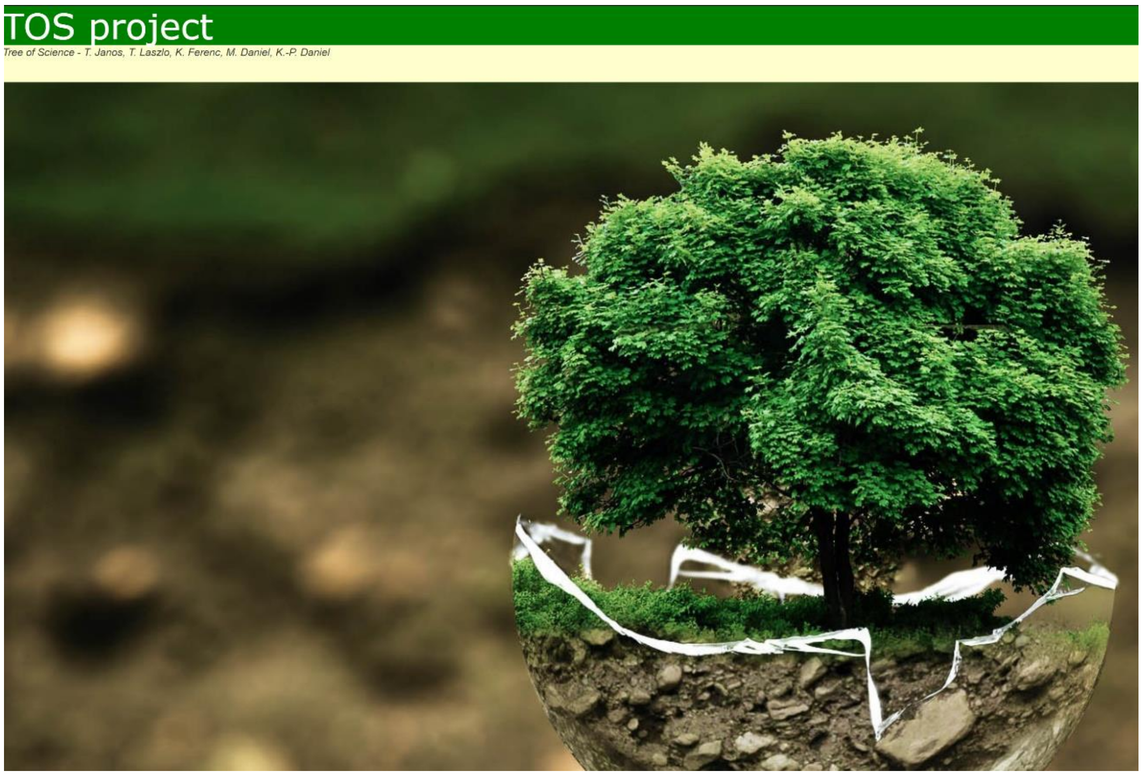

TOS project
Tree of Science - T. Janos, T. Laszlo, K. Ferenc, M. Daniel, K.-P. Daniel

Figure 1.6 Tree of Science base page

upload all the researches they have. Make a community with the researchers from all field of science and build together the Tree Of Science.

1.3.1 Google knowledge graph

Somewhere in 2012, Google announced the unveiling of the Knowledge Graph, a tool which collates all the facts about people, places, and things to create interconnected search results and present them in form of useful answers and not just links.

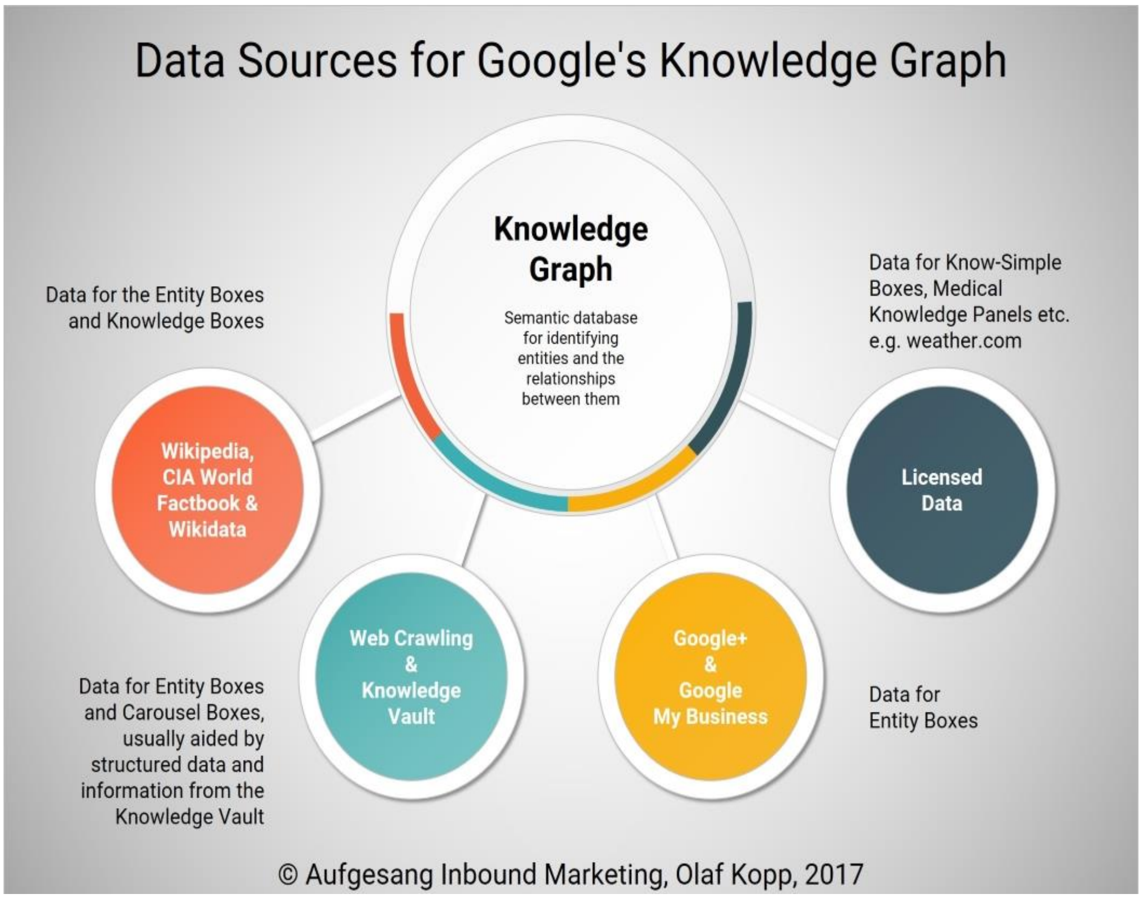

Data Sources for Google's Knowledge Graph
Knowledge Graph
Semantic database for identifying entities and the relationships between them
Data for the Entity Boxes and Knowledge Boxes
Data for Know-Simple Boxes, Medical Knowledge Panels etc. e.g. weather.com
Wikipedia, CIA World Factbook & Wikidata
Licensed Data
Web Crawling & Knowledge Vault
Google+ & Google My Business
Data for Entity Boxes and Carousel Boxes, usually aided by structured data and information from the Knowledge Vault
Data for Entity Boxes
© Aufgesang Inbound Marketing, Olaf Kopp, 2017

Figure 1.7 Data for Knowledge

In science, the numbers of publications are growing like a tree. We make publications from publication. And humanity wants to know about himself more and more. The Knowledge Graph is a knowledge base used by Google and its services to enhance its search engine's results with information gathered from a variety of sources.

The information covered by the Knowledge Graph grew significantly after launch, tripling its original size within seven months (covering 570 million entities and 18 billion facts), and being able to answer "roughly one-third" of the 100 billion monthly searches Google processed in May 2016. The Knowledge Graph has been criticized for providing answers without source attribution or citation.

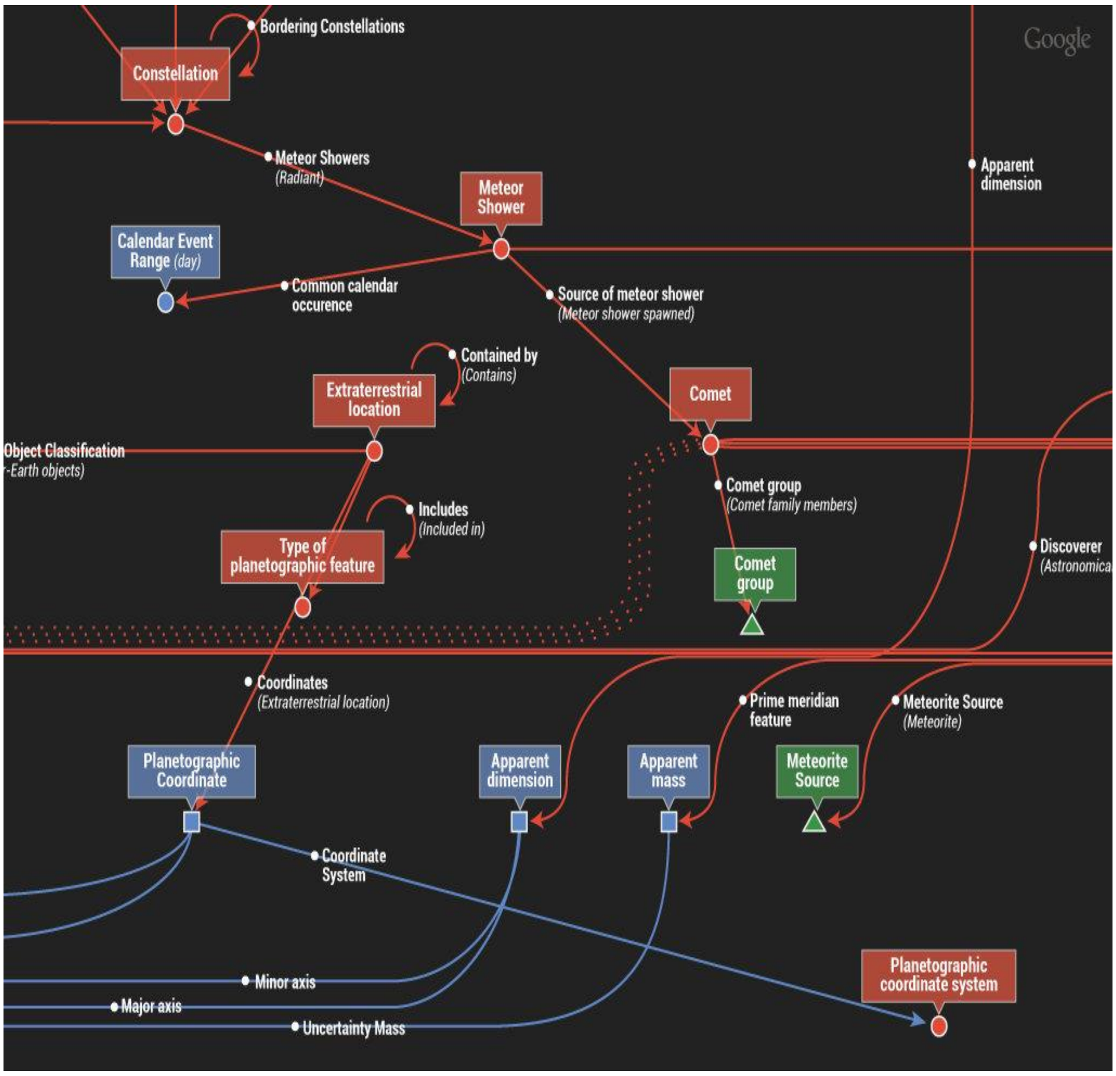


Figure 1.8 The Flow of Knowledge

1.3.2 Aims

We would rather build a system which could be helpful to scientists all over the world to browse the knowledge tree in a different way.

1.4 How it works on TDK[3]

TDK is a good example to show how ScienceMesh can help to improve search and browse experience of scientific knowledge. Independent research opportunity is offered to university students in the framework of the so-called Scientific Students' Associations ("TDK", which stands for 'Tudományos Diák Konferencia' in Hungarian).

Perhaps the TDK report is not as informative as a scientific paper but it is a closer example.

Kezdőlap
Tudományos Diákköri Konferencia
Válasszon a karok rendezvényei közül »

Figure 1.9 TDK main page

1.4.1 Nodes and edges

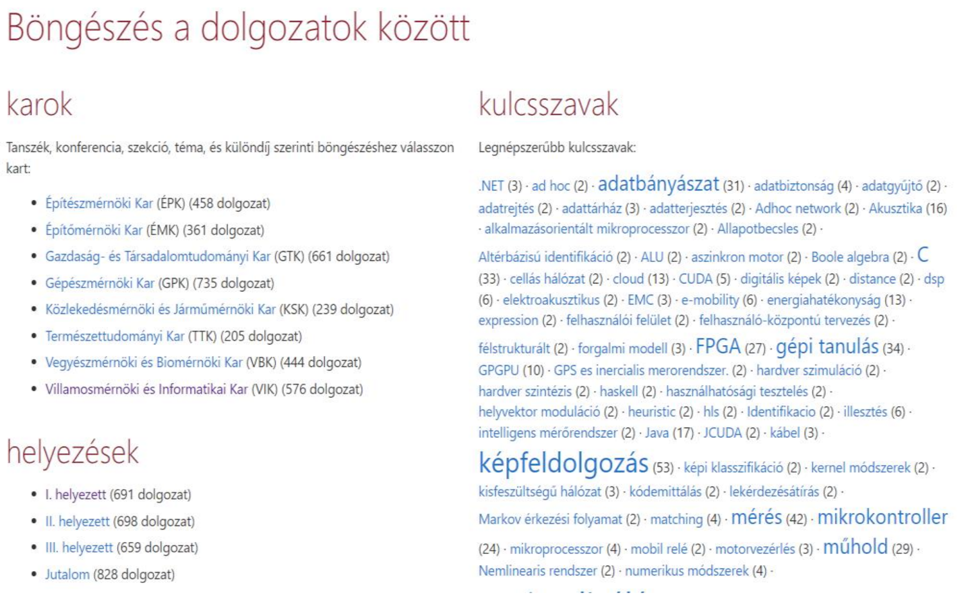
To build the SicenceMesh on TDK we have to choose what will be our nodes and find connections between them to localize all the edges. We also have to make nodes types to show the difference between the publications and the keywords.

2 Preparation for Database

To build this new system the first thing is to get the data for the optimized graph-based database. If we had the tools to connect directly to the database, the preparation would be simple. We get the data and transfer into nodes and edges but in this option, we don't have this tool, so we have to build an automatic downloader to get all the information from the webpage. Using this technic, we can build ScienceMesh from any structured web pages.

2.1 Get data from TDK webpage

The TDK topology is simple, there are faculties, rankings, years and keywords on the main browsing page.



Böngészés a dolgozatok között

karok

Tanszék, konferencia, szekció, téma, és különdíj szerinti böngészéshez válasszon kart:

- Építészmérnöki Kar (ÉPK) (458 dolgozat)
- Építőmérnöki Kar (ÉMK) (361 dolgozat)
- Gazdaság- és Társadalomtudományi Kar (GTK) (661 dolgozat)
- Gépészmérnöki Kar (GPK) (735 dolgozat)
- Közlekedésmérnöki és Járműmérnöki Kar (KSK) (239 dolgozat)
- Természettudományi Kar (TTK) (205 dolgozat)
- Vegyészmérnöki és Biomérnöki Kar (VBK) (444 dolgozat)
- Villamosmérnöki és Informatikai Kar (VIK) (576 dolgozat)

helyezések

- I. helyezett (691 dolgozat)
- II. helyezett (698 dolgozat)
- III. helyezett (659 dolgozat)
- Jutalom (828 dolgozat)

kulcsszavak

Legnépszerűbb kulcsszavak:

.NET (3) · ad hoc (2) · adatbányászat (31) · adatbiztonság (4) · adatgyűjtő (2) · adatrejtés (2) · adattárház (3) · adatterjesztés (2) · Adhoc network (2) · Akusztika (16) · alkalmazásorientált mikroprocesszor (2) · Allapotbecsles (2) · Altérbázisú identifikáció (2) · ALU (2) · aszinkron motor (2) · Boole algebra (2) · C (33) · cellás hálózat (2) · cloud (13) · CUDA (5) · digitális képek (2) · distance (2) · dsp (6) · elektroakusztikus (2) · EMC (3) · e-mobility (6) · energiahatékonyság (13) · expression (2) · felhasználói felület (2) · felhasználó-központú tervezés (2) · félstrukturált (2) · forgalmi modell (3) · FPGA (27) · gépi tanulás (34) · GPGPU (10) · GPS es inercialis merorendszer. (2) · hardver szimuláció (2) · hardver szintézis (2) · haskell (2) · használhatósági tesztelés (2) · helyvektor moduláció (2) · heuristic (2) · hls (2) · Identifikacio (2) · illesztés (6) · intelligens mérőrendszer (2) · Java (17) · JCUDA (2) · kábel (3) · képfeldolgozás (53) · képi klasszifikáció (2) · kernel módszerek (2) · kisfeszültségű hálózat (3) · kódemittálás (2) · lekérdezésátírás (2) · Markov érkezési folyamat (2) · matching (4) · mérés (42) · mikrokontroller (24) · mikroprocesszor (4) · mobil relé (2) · motorvezérlés (3) · műhold (29) · Nemlinearis rendszer (2) · numerikus módszerek (4) ·

Figure 2.1 Browsing page

2.1.1 Make topology

The webpage also has a searching page where we can choose where and what we want to search. From this, we can start the hierarchy choosing the faculties and start the

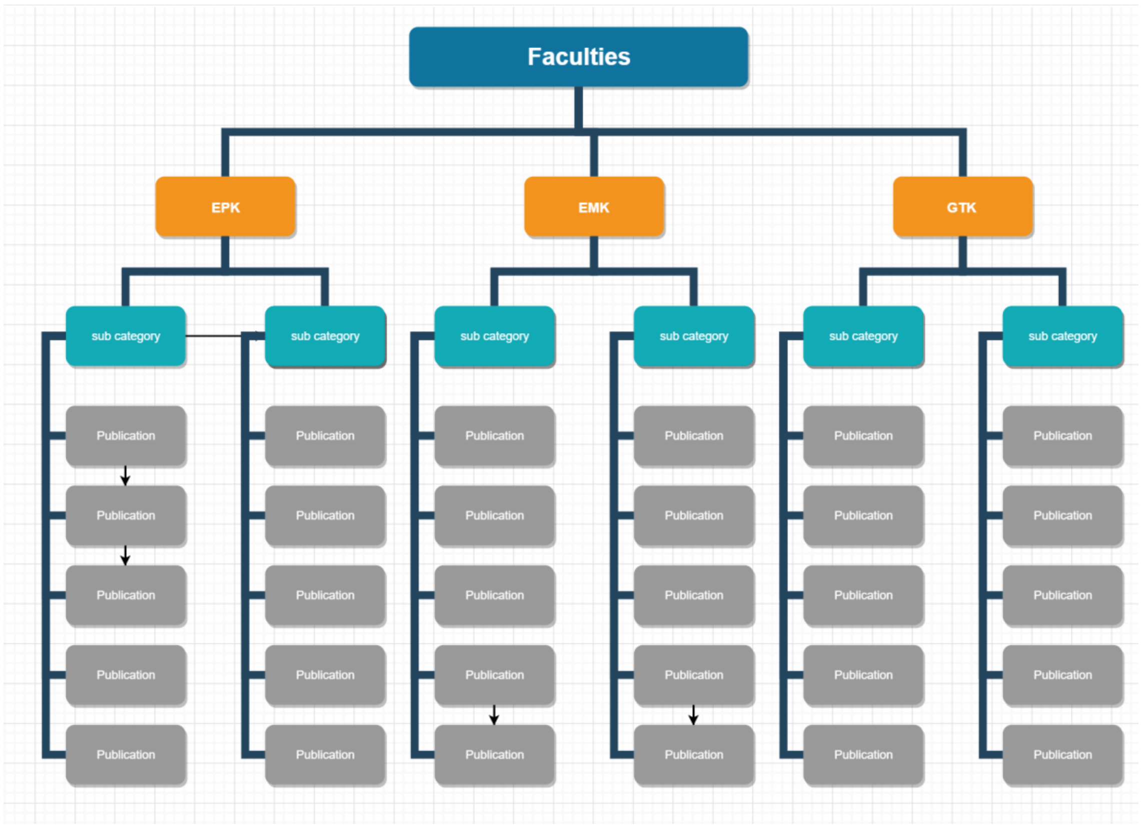
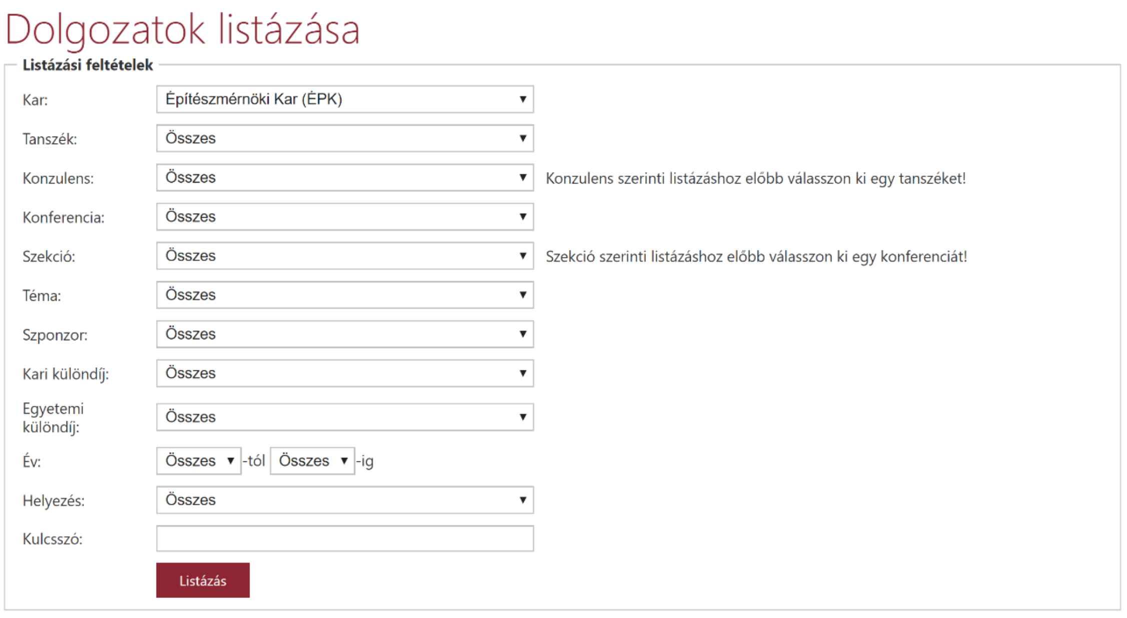


Figure 2.3 Example of structure

search engine. It will give us all the results on that faculties and we can start to get the details of the publications, going into the results and download the details and the pdf.



Dolgozatok listázása

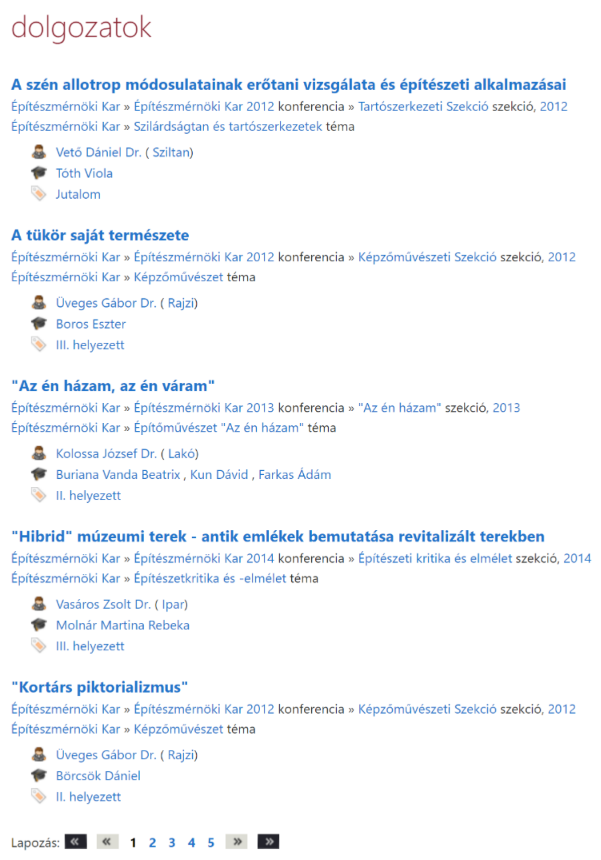
Listázási feltételek

Kar:	Építészmérnöki Kar (ÉPK)	
Tanszék:	Összes	
Konzulens:	Összes	Konzulens szerinti listázáshoz előbb válasszon ki egy tanszéket!
Konferencia:	Összes	
Szekció:	Összes	Szekció szerinti listázáshoz előbb válasszon ki egy konferenciát!
Téma:	Összes	
Szponzor:	Összes	
Kari különdíj:	Összes	
Egyetemi különdíj:	Összes	
Év:	Összes -tól Összes -ig	
Helyezés:	Összes	
Kulcsszó:		

Listázás

Figure 2.2 Search page

The system will go step by step into all the research from all the pages and then do the same in the next faculties.



dolgozatok

A szén allotrop módosulatainak erőtani vizsgálata és építészeti alkalmazásai
Építészmérnöki Kar » Építészmérnöki Kar 2012 konferencia » Tartószerkezeti Szekció szekció, 2012
Építészmérnöki Kar » Szilárdságtan és tartószerkezetek téma

- Vető Dániel Dr. (Sziltan)
- Tóth Viola
- Jutalom

A tükör saját természete
Építészmérnöki Kar » Építészmérnöki Kar 2012 konferencia » Képzőművészeti Szekció szekció, 2012
Építészmérnöki Kar » Képzőművészet téma

- Üveges Gábor Dr. (Rajzi)
- Boros Eszter
- III. helyezett

"Az én házam, az én váram"
Építészmérnöki Kar » Építészmérnöki Kar 2013 konferencia » "Az én házam" szekció, 2013
Építészmérnöki Kar » Építőművészet "Az én házam" téma

- Kolossa József Dr. (Lakó)
- Buriana Vanda Beatrix , Kun Dávid , Farkas Ádám
- II. helyezett

"Hibrid" múzeumi terek - antik emlékek bemutatása revitalizált terekben
Építészmérnöki Kar » Építészmérnöki Kar 2014 konferencia » Építészeti kritika és elmélet szekció, 2014
Építészmérnöki Kar » Építészetkritika és -elmélet téma

- Vasáros Zsolt Dr. (Ipar)
- Molnár Martina Rebeka
- III. helyezett

"Kortárs piktorializmus"
Építészmérnöki Kar » Építészmérnöki Kar 2012 konferencia » Képzőművészeti Szekció szekció, 2012
Építészmérnöki Kar » Képzőművészet téma

- Üveges Gábor Dr. (Rajzi)
- Börcsök Dániel
- II. helyezett

Lapozás: « « 1 2 3 4 5 » »

Figure 2.4 Publications

2.1.2 Request and Beautiful python libraries

The request package in python gives us the opportunity to automate the webpage calling.

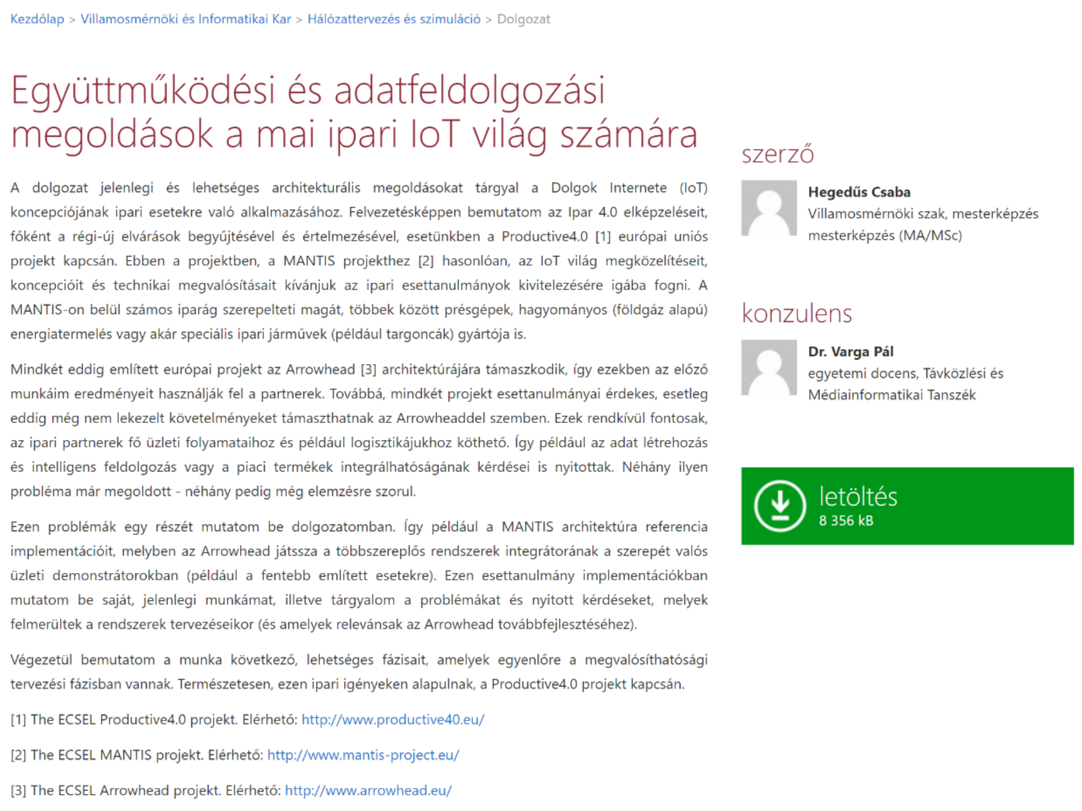
Requests[4] allow you to send organic, grass-fed HTTP/1.1 requests, without the need for manual labor. There's no need to manually add query strings to your URLs or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, thanks to urllib3.

With BeatifulSoup[5] we can download the page and parse into a string to find the data we want to save to our database. It also helps us to get the next reference to open by request.

BeautifulSoup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching and modifying the parse tree. It commonly saves programmers hours or days of work.

2.1.3 Publication details selection

Some of the details are also in the publications and in the pages, we download from, like our teacher's name, my name etc. In order to avoid duplication of the data. First, we get all the information we have from the web page and after that, we download the pdf and get the text from it by a parser.



Kezdőlap > Villamosmérnöki és Informatikai Kar > Hálózattervezés és szimuláció > Dolgozat

Együttműködési és adatfeldolgozási megoldások a mai ipari IoT világ számára

A dolgozat jelenlegi és lehetséges architekturális megoldásokat tárgyal a Dolgok Internete (IoT) koncepciójának ipari esetekre való alkalmazásához. Felvezetésképpen bemutatom az Ipar 4.0 elképzeléseit, főként a régi-új elvárások begyűjtésével és értelmezésével, esetünkben a Productive4.0 [1] európai uniós projekt kapcsán. Ebben a projektben, a MANTIS projekthez [2] hasonlóan, az IoT világ megközelítéseit, koncepcióit és technikai megvalósításait kívánjuk az ipari esettanulmányok kivitelezésére igába fogni. A MANTIS-on belül számos iparág szerepelteti magát, többek között présgépek, hagyományos (földgáz alapú) energiatermelés vagy akár speciális ipari járművek (például targoncák) gyártója is.

Mindkét eddig említett európai projekt az Arrowhead [3] architektúrájára támaszkodik, így ezekben az előző munkáim eredményeit használják fel a partnerek. Továbbá, mindkét projekt esettanulmányai érdekes, esetleg eddig még nem lekezelt követelményeket támaszthatnak az Arrowheaddel szemben. Ezek rendkívül fontosak, az ipari partnerek fő üzleti folyamataihoz és például logisztikájukhoz köthető. Így például az adat létrehozás és intelligens feldolgozás vagy a piaci termékek integrálhatóságának kérdései is nyitottak. Néhány ilyen probléma már megoldott - néhány pedig még elemzésre szorul.

Ezen problémák egy részét mutatom be dolgozatomban. Így például a MANTIS architektúra referencia implementációit, melyben az Arrowhead játssza a többszereplős rendszerek integrátorának a szerepét valós üzleti demonstrátorokban (például a fentebb említett esetekre). Ezen esettanulmány implementációkban mutatom be saját, jelenlegi munkámat, illetve tárgyalom a problémákat és nyitott kérdéseket, melyek felmerültek a rendszerek tervezéseikor (és amelyek relevánsak az Arrowhead továbbfejlesztéséhez).

Végezetül bemutatom a munka következő, lehetséges fázisait, amelyek egyenlőre a megvalósíthatósági tervezési fázisban vannak. Természetesen, ezen ipari igényeken alapulnak, a Productive4.0 projekt kapcsán.

[1] The ECSEL Productive4.0 projekt. Elérhető: http://www.productive40.eu/

[2] The ECSEL MANTIS projekt. Elérhető: http://www.mantis-project.eu/

[3] The ECSEL Arrowhead projekt. Elérhető: http://www.arrowhead.eu/

szerző

Hegedűs Csaba
Villamosmérnöki szak, mesterképzés
mesterképzés (MA/MSc)

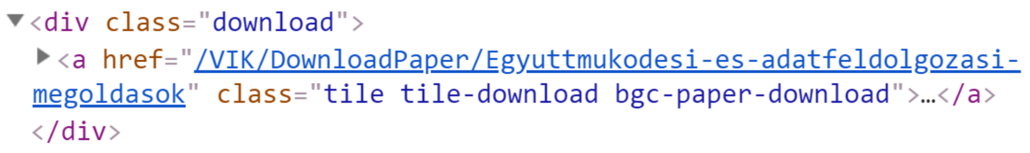
konzulens

Dr. Varga Pál
egyetemi docens, Távközlési és Médiainformatikai Tanszék

Figure 2.5 An example publication

2.1.4 PDF downloader

With BeautifulSoup, we can find the tag which contains the reference for the article. After we found it we have to send a request and it will download it.



```
▼<div class="download">
  ▶<a href="/VIK/DownloadPaper/Egyuttmukodesi-es-adatfeldolgozasi-
megoldasok" class="tile tile-download bgc-paper-download">…</a>
</div>
```

Figure 2.6 The link for pdf

2.2 Parse pdf into JSON

2.2.1 ScienceBeam

A set of tools to allow PDF to XML conversion, utilizing Apache Beam and other tools. The aim of this project is to bring multiple tools together to generate a full XML document.

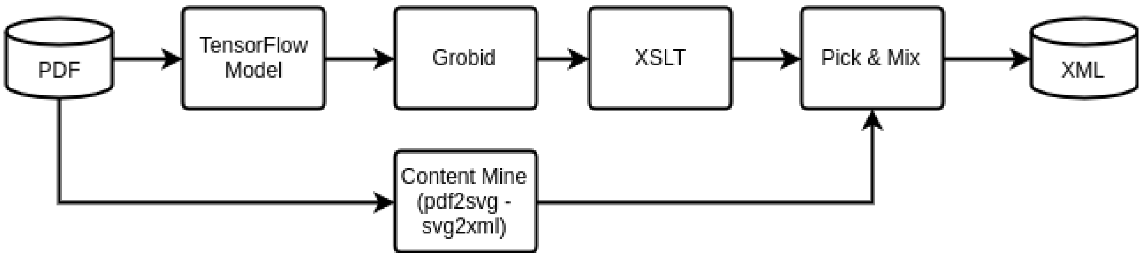


Figure 2.7 ScienceBeam process

2.2.2 Parse information by the scheme

Almost all the research was written by the same schema. We will parse all the texts we have in the publication into XML by ScienceBeam[6]. In XML we have tags for the string types. From this tags we make json. From json we can get all attribute we will save in our database.

2.2.3 Get references from publications

The most exciting things are the references. We want to know how many people have used our research and from that, we can make more nodes for the referenced article.

Therefore, we have to find also the footnotes inside the text and select the sentence which was referred.

2.2.4 Make the structure

Above the all we have to make an automatic data miner which would be also good if there is a new research and can connect for the keywords and to be stable for the system

3 Database

Now we have all the data from the old system. The next step is to build a database management system to save the information we have in a structured way.

3.1 Choosing a database management system

The database management system is the heart of modern applications, and choosing the best DBMS software for your organization is imperative to the success of your IT projects and systems.

3.1.1 Mongo

MongoDB[7] stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

MONGO is a NoSQL type system. Which would be good for the project because we can build our system based on nodes and edges and NoSQL language can handle easily if the connections number its not the same. But the main problem is that if we want to search we will have problems in the future with the time and for the selection what we want or what we don’t.

Any relational database has a typical schema design that shows a number of tables and the relationship between these tables. While in MongoDB, there is no concept of relationship.

3.1.2 Neo4j

The graph platform takes a connections-first approach to data. It broadens a company’s ability recognize the importance of persisting relationships and connections through every transition of existence: from the idea, to design in a logical model, to implementation in a physical model, to operation using a query language and to persistence within a scalable, reliable database system.

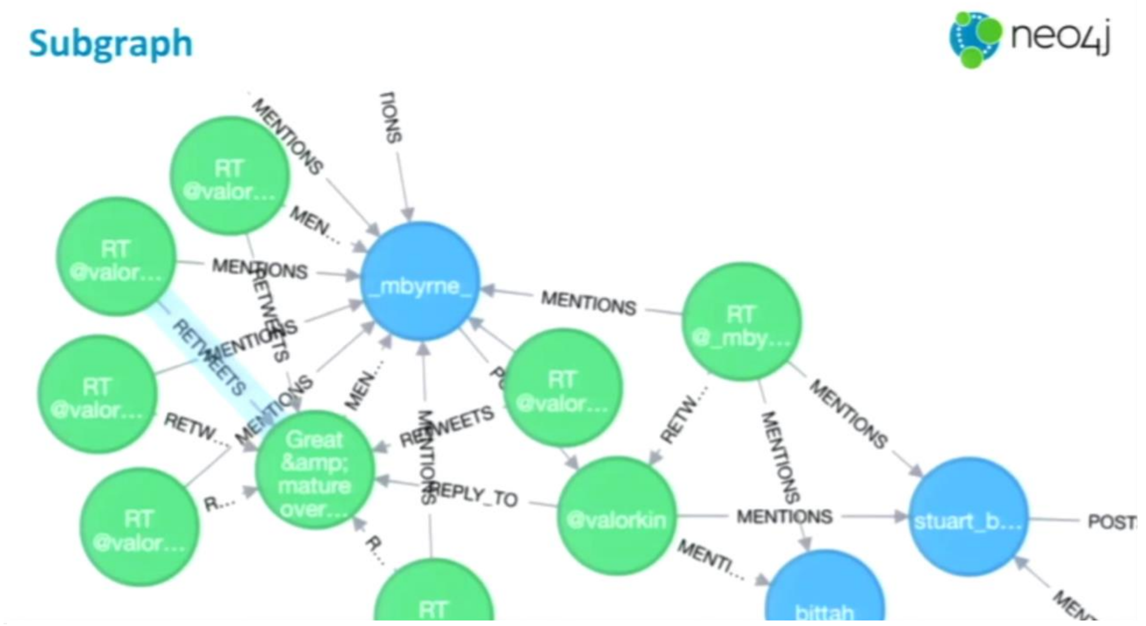


Figure 3.1 Subgraph from Neo4j

Neo4j[8] is a graph based database tool. It's really good for making the edges and nodes but not enough to make a structure which also contains the information of the nodes if there is a lot of information.

3.1.3 MySQL

MySQL[9] is the world's most popular open source database. Whether you are a fast growing web property, technology ISV or large enterprise, MySQL can cost-effectively help you deliver high performance, scalable database applications.

MySQL is an SQL language. For this project, I will use the only MySQL because whit the tables and the keys and whit the capability searchable thinks it's the best. If we want to search something we have to write the query which will give us the best answer and also the data can storage in the most optimum way.

3.2 MYSQL hierarchy

In this hierarchy, we have three layers. One for visualization which can help to show the nodes and edges. Thes second layer shows the inside of the nodes and the third one give us details of the connections.

3.2.1 Visual label

This layer contains the nodes the edges and the textboxes. We make a node for any object we can show for the user. We make the edges for all the connections we have with the nodes. And the textboxes help us to get all the knowledge we have inside of the node.

3.2.2 Knowledge label

In the knowledge label, we have a table for all types of nodes to have more specific results if the user wants to know more about the node.

3.2.3 Details label

In this label, we store the connection tables to be able to count the types of the subcategorize.

3.3 Tables

In MySQL If we have an attribute which contains more then one object, we have to normalize to have a faster search and more compact database. Maybe we have to make extra tables to save the information structure but it helps us to get more information about the objects.

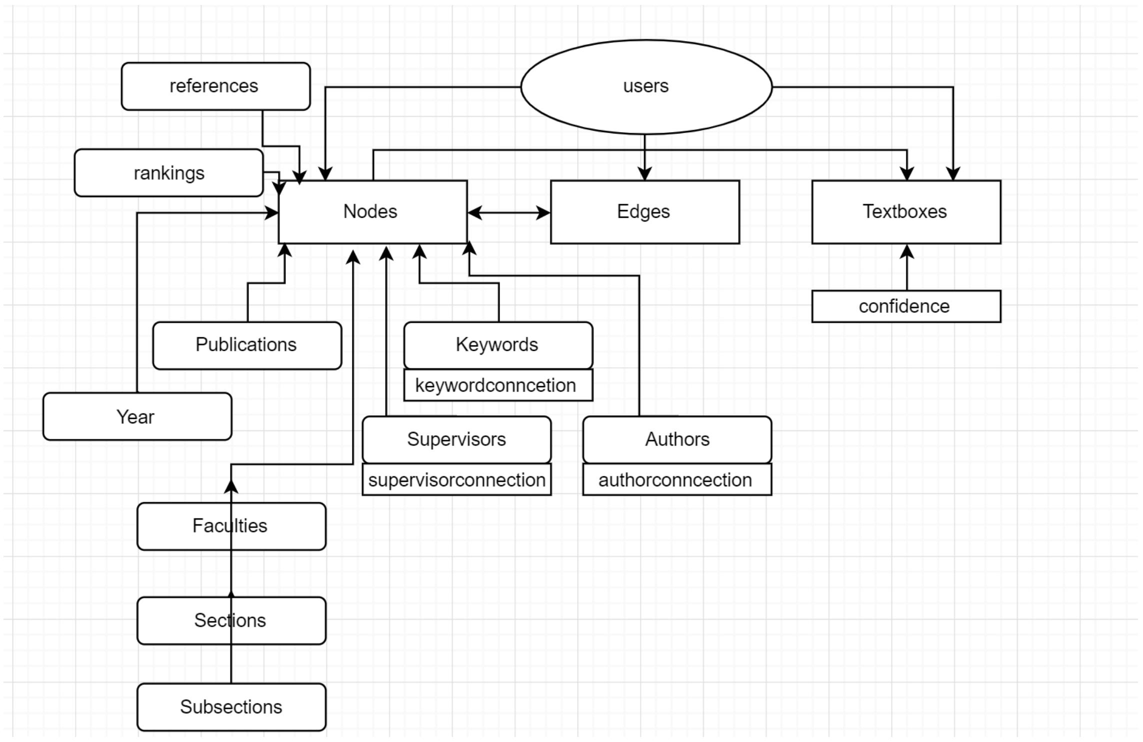


Figure 3.2 Table connections

3.3.1 Nodes

This table is from the Vision label. The labelid helps us to connect to the specific type from knowledge label. The added field give us if it is from the database or some user added to it.

Field	Type
id	int(11)
labelid	int(11)
type	enum('publ','keyword',’references’,’author’,’supervisor’,’year’,’section’,’subsection’ ’faculties’)
label	varchar(31)
title	varchar(255)
colour	varchar(31)
size	int(3)
added	bool

3.3.2 Edges

The edges table is also from vison label. We make edges between the source nodeid and the target nodeid.

Field	Type
id	int(11)
source	int(11)
target	int(11)
type	enum('subproblem','desc',’ref’)
size	int(3)
added	bool

3.3.3 Textboxes

The last element of vision label. We can add content to nodes by the nodeid. The type of the content its given by the user identified by the userid.

Field	Type
id	int(11)
nodeid	int(11)
content	text
type	varchar(31)
url	varchar(255)
confidence	int(3)
userid	int(10)

3.3.4 Publications

Elements of knowledge label. We save here the rawtext, abstract, title and the year of the publication.

Field	Type
id	int(11)
title	varchar(255)
rawtext	text
year	int(4)
abstarct	text

3.3.5 Authors

Elements of knowledge label. We save the name the faculties and the counted numbers of researches.

Field	Type
id	int(11)
name	varchar(255)
faculties	varchar(255)
counted	int(3)

3.3.6 Authorconncetor

Elements of details label. Make the connections in knowledge label.

Field	Type
pubid	int(11)
authorid	int(11)

3.3.7 Keywords

Elements of knowledge label. Keywords name and counted in publications.

Field	Type
id	int(11)
keyword	varchar(255)
keywordcount	int(3)

3.3.8 Keyword connector

Elements of details label make the connection between keywords and publications.

Field	Type
pubid	int(11)
keywordid	int(11)

3.3.9 Supervisors

Elements of knowledge label. We save the name the faculties and the counted numbers of researches.

Field	Type
id	int(11)
name	varchar(255)
faculties	varchar(255)
counted	int(11)

3.3.10 Supervisorconnector

Elements of knowledge label. Keywords name and counted in publications.

Field	Type
pubid	int(11)
supervisorid	int(11)

3.3.11 References

Elements of knowledge label. We save here the references of the publications and the sentence for the referenced text.

Field	Type
pubid	int(11)
title	varchar(255)
author	varchar(255)
URL	text
sentence	text

3.3.12 Faculties, section, subsection

Elements of knowledge label. The names of the categories and the counted publications in that categories. Both of the three tables have the same columns.

Field	Type
id	int(11)
name	varchar(255)
count	int(4)

3.3.13 Users

This is the table to save users information. The confidencenum is to show the user recommendation number.

Field	Type
id	int(11)
username	varchar(255)
email	varchar(255)
passwd	varchar(255)
coinfidencenum	int(3)

3.3.14 Confidence

Elements of details label to save the points given to the contents by the users.

Field	Type
userid	int(11)
textboxid	int(11)

3.3.15 Rankings

Elements of knowledge label. We save the name of the rankings and the counted numbers of researches.

Field	Type
id	int(11)
rankings	varchar(255)
counted	int(4)

3.3.16 Year

Elements of knowledge label. We save the years and the counted numbers of researches.

Field	Type
id	int(11)
year	int(4)
count	int(4)

3.4 Graph representation

We also want to browse in the graph not only search. We have to make an opening view to start somewhere the browsing if we don’t know what we want to search we can find it in this way. We have to find the optimum number of nodes and edges to show. If we have too many we can't see through the graph and we will lose information.

4 Search algorithm

Nowadays search algorithms give the results which contain the keyword. If our system would work like this, we will give dozens of nodes and if we are lucky they connect to each other but most of the time they are not.

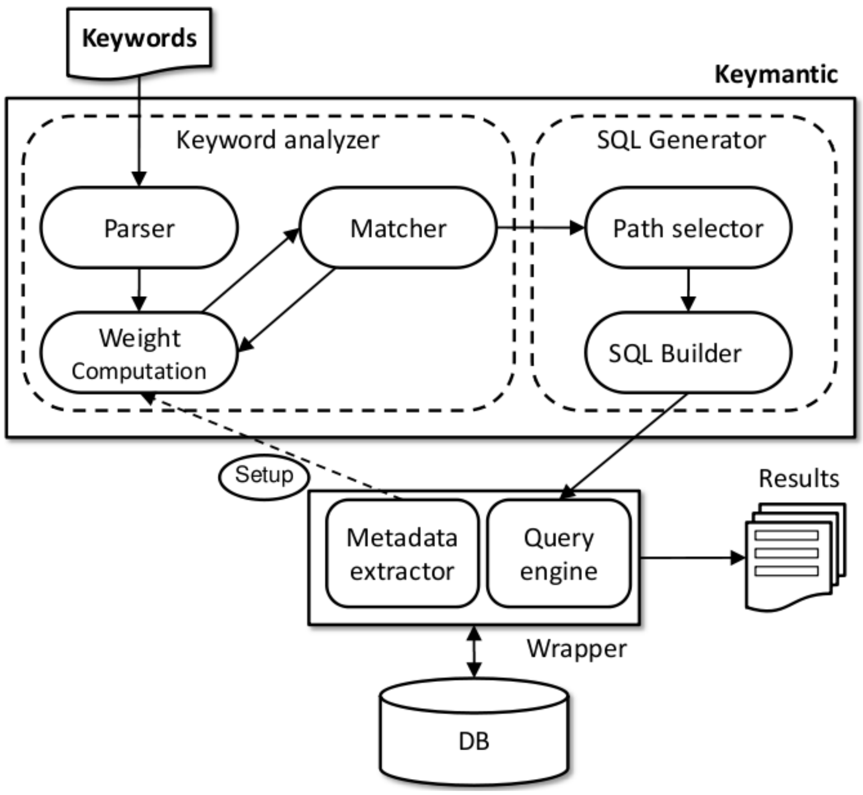


Figure 4.1 SQL keyword search engine

4.1 Algorithms

In Google, if we want to search for one common word or a more specific expression the algorithm will be the same. We will have fewer results in the second search.

4.1.1 Simple search

For a basic search, we use simple SQL query to find nodes. The aim is to have nodes result and for the nodes we have to find edges which connected to them. The users give us the expression to search in the nodes table.

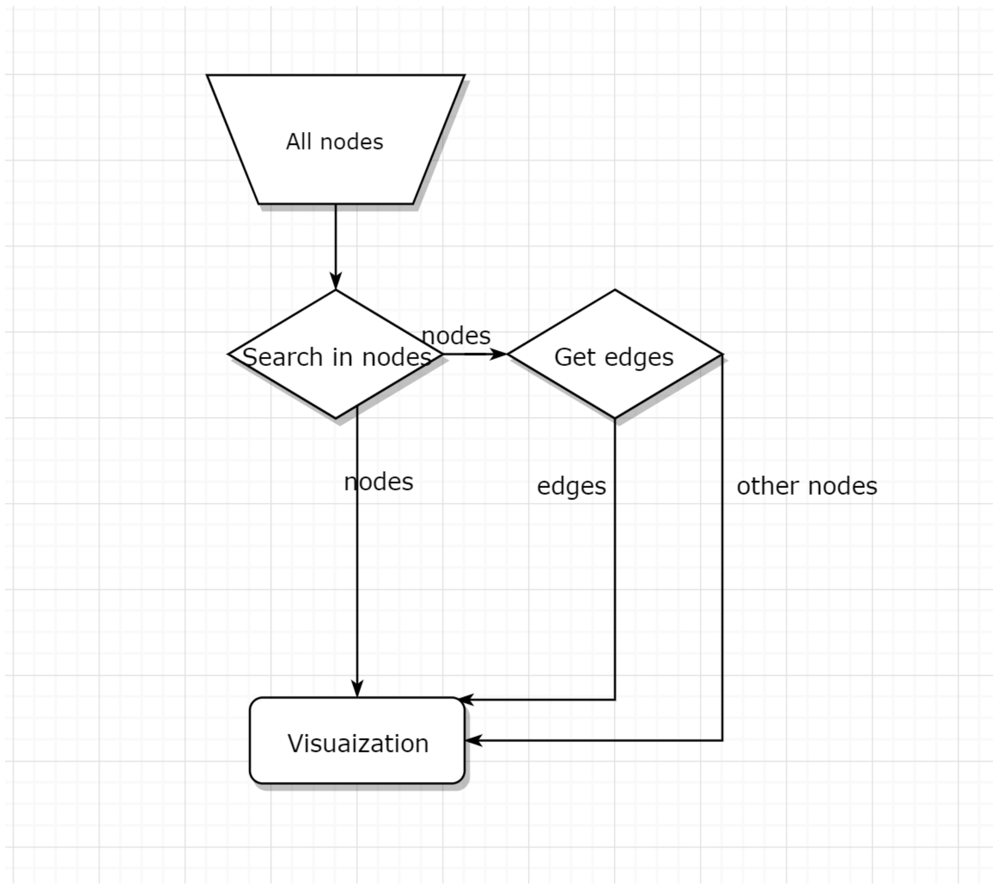


Figure 4.2 Get nodes and neighbours

4.1.2 Search in database

We also have the opportunity to search inside of the publications or the given contents in the textbox. We have the rawtexts of the researches in publications tables. The contents are inside of textboxes table in the column of texts. This type of search is more

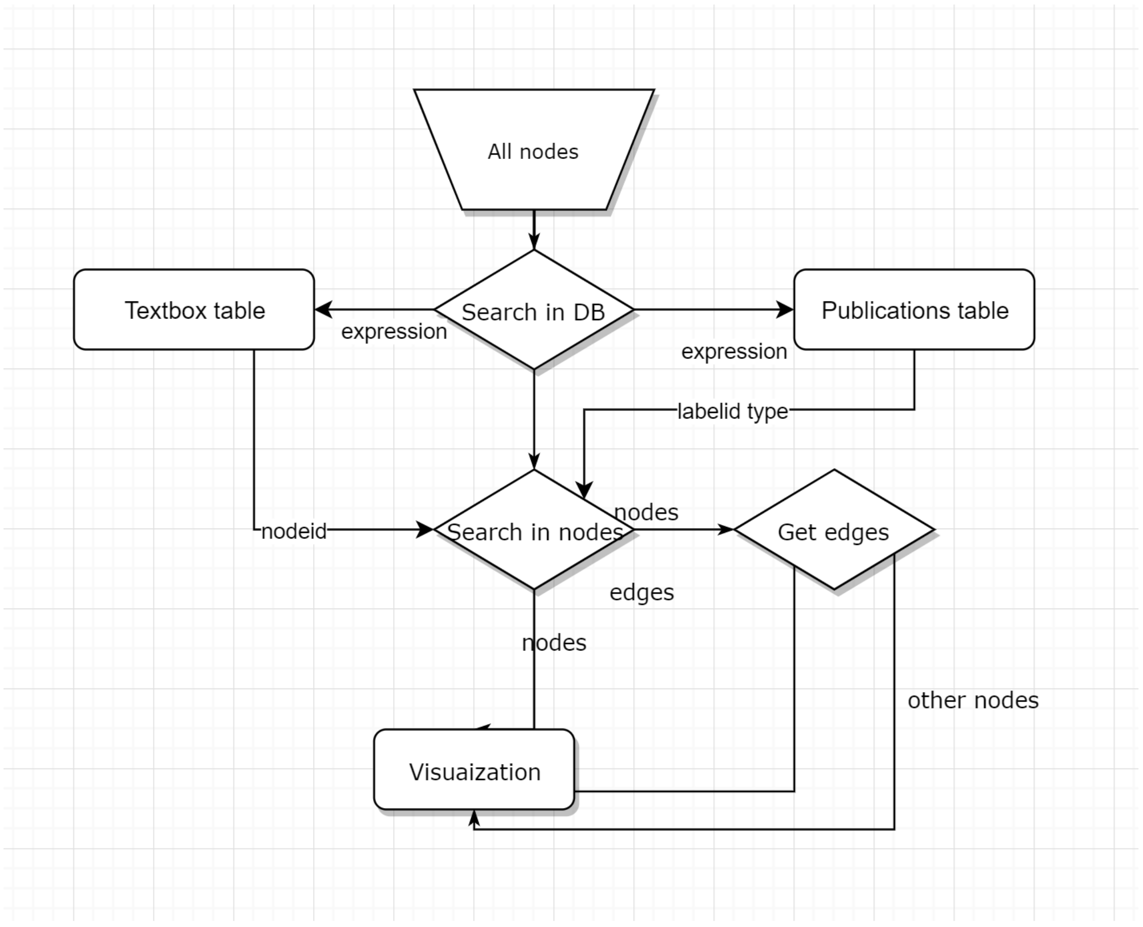


Figure 4.3 Search in texts

complicated, we have to get texts from the tables to have all the results which contain the expression.

4.2 A new way to get results

With the power of graph visualization, we have the opportunity to show the range of the expression. If we have all kind of nodes it will be an unsettled graph. To solve this problem we have view types which help the user to have well-tended architecture.

4.2.1 Get results of neighbours

The first thing is to found the results nodes. After that we search in edges table to have all the connections, then we search for the end of the edges to find the neighbours of the nodes. It will help us to localize the results in the system and get more knowledge about the expression we are looking for.

4.2.2 Choose the specific nodes based on view types

The chosen view gives us which type of nodes will be shown. The pre-settled hierarchy improves the browsing experience.

4.3 The number of nodes

If the nodes rank we can not see the connections and if we have too many edges we don't understand the meanings of them. We have to optimize the number of nodes.

For this, we give the opportunity to change the limit of the nodes. The default number is twenty, it is the optimum number of nodes to show in any device.

4.4 Order by the recommendation system

Now all the nodes we have are on the same level but if we want search we have to make an order on the results. In TDK all the research have a rank given by the professors of the judge. On the other way, we can count the publication number of supervisors or by the keywords. Based on this information we can order the results.

The users will also have the opportunity to give feedback on the contents in the textbox.

4.4.1 Size of nodes

We have to create rules for all type of nodes. With publications, we know the ranking. If it's first prize winner the size of the node is bigger then if its only have placed in third. We can count the research based on the keywords, supervisors and in all objects have count column. The size of the nodes is proportional to the counted number.

4.4.2 Confidence

We can add nodes and textbox contents in this system. About this new knowledge given by the users, we don't have any information. We have to add confidence points for the textboxes. If a user reads a content about the nodes he can recommend a number from one to ten to give feedback.

4.5 View types

In Google, we can choose the type of results. We can search in pictures and in the map. In our graph visualization, we have view types which help us to look after a problem with other prospects.

4.5.1 Keyword based

The root nodes of this type of view are the keywords. Then we have publications and the author of the publication and the supervisor.

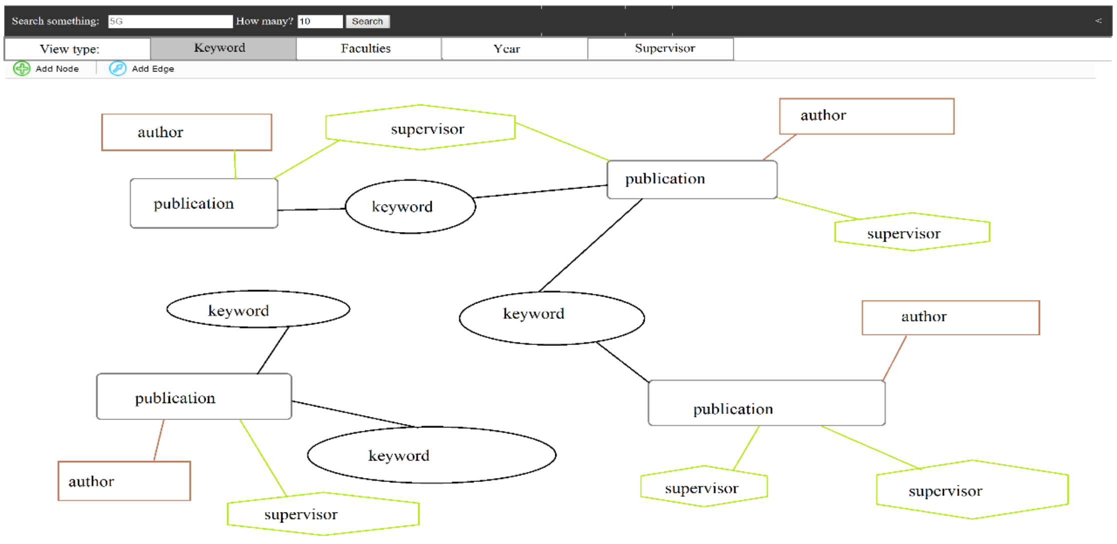


Figure 4.4 Illustration for keyword based view

4.5.2 Supervisor based

Here the root nodes are the supervisors. The supervisors have students who wrote the researches. The research also has a reference to show.

4.5.3 Faculties based

We start from the faculties to the section for subsection to get the publications

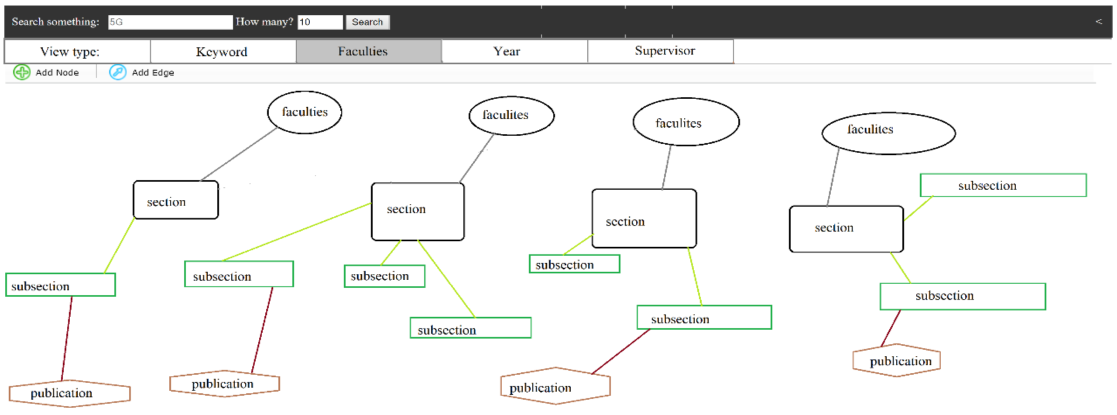


Figure 4.5Illustration for faculties based view

4.5.4 Year based

The base nodes of this kind of view are years. From years we have the faculties and the publications. It is a good view if we want to compare the years.

5 Visualization

Graph visualizations are a powerful tool to convey the content of a graph. They can highlight patterns and show clusters and connections. There are many excellent options for graph visualization, such as D3.js, three.js, sigma.js, Alchemy.js and vis.js[10].

5.1 Vis JavaScript

A dynamic, browser-based visualization library. The library is designed to be easy to use, to handle large amounts of dynamic data, and to enable manipulation of and interaction with the data.

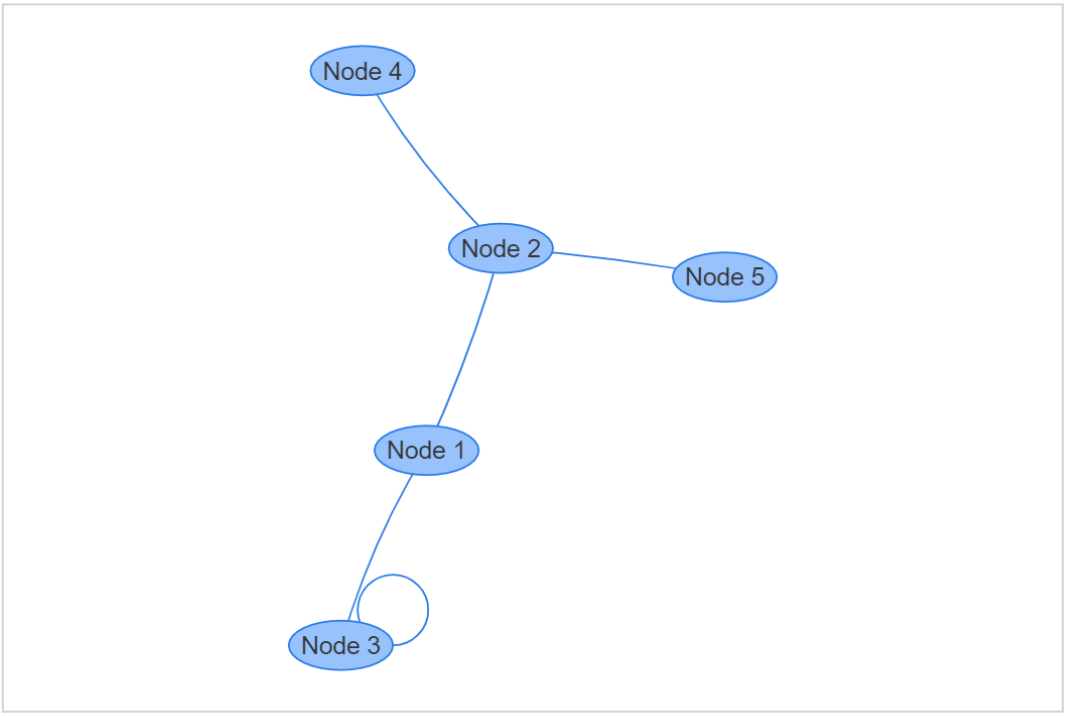


Figure 5.1Example view of nodes

5.1.1 Connect to database

Flask[11] is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

5.2 Objects

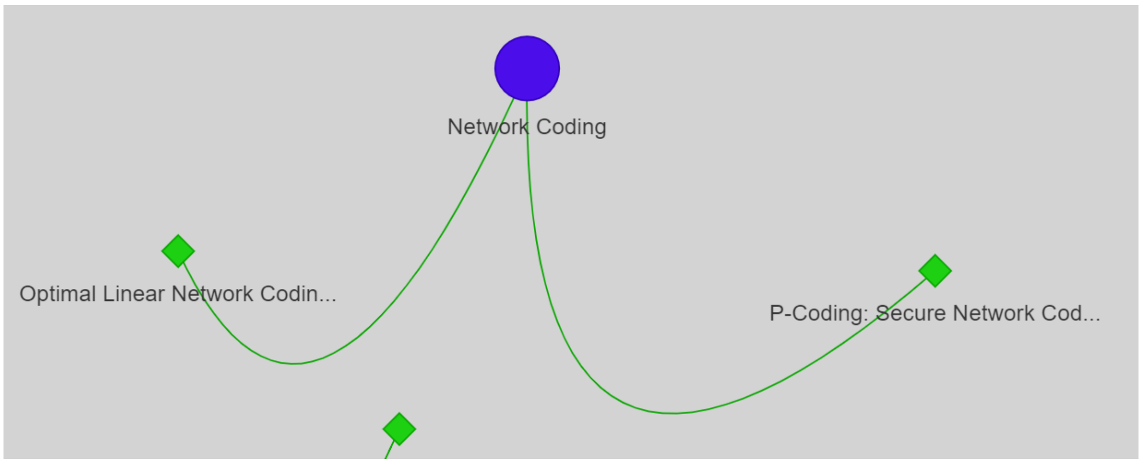

Network Coding
Optimal Linear Network Codin...
P-Coding: Secure Network Cod...

Figure 5.2 Nodes and edges in vis.jss

5.2.1 Nodes

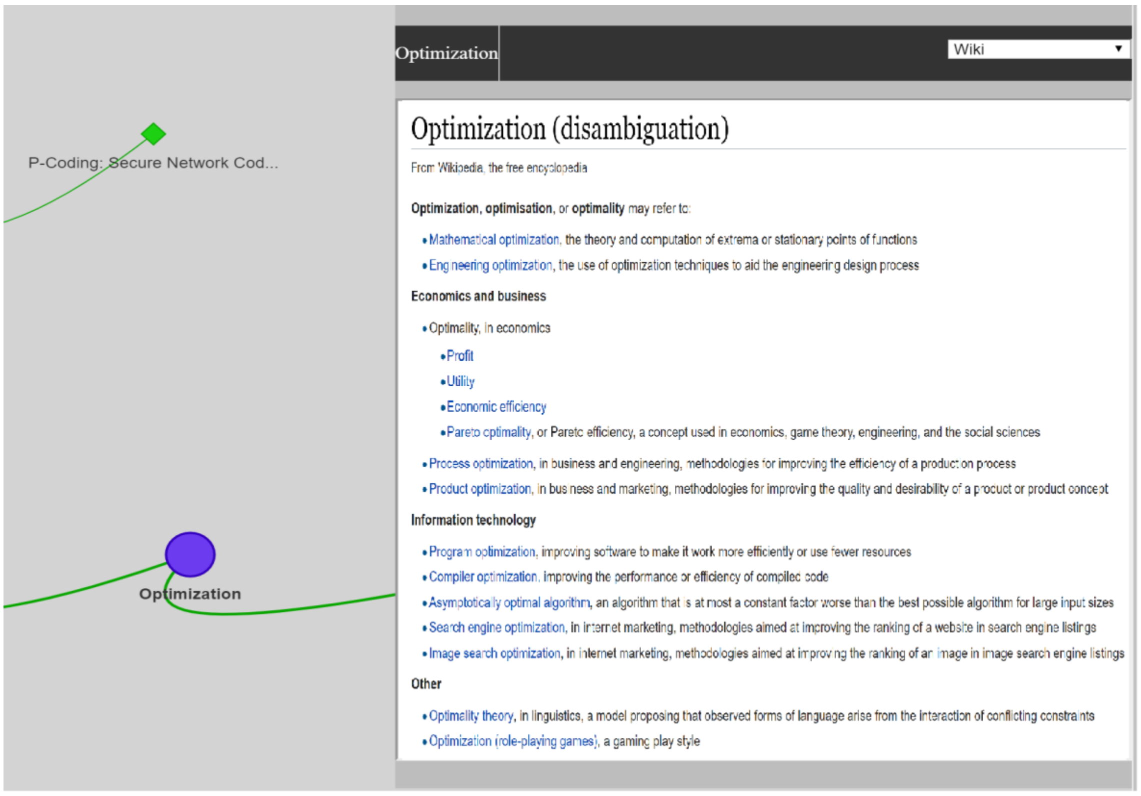
By definition, a Graph is a collection of nodes (vertices) along with identified pairs of nodes. A node v is a terminal point or an intersection point of a graph. It is the abstraction of a location such as a city, an administrative division, a road intersection or a transport terminal.

5.2.2 Edges

Edges can be either directed or undirected. Directed edge points from one vertex to another and an undirected has no direction. An example of this distinction would be on the Facebook and Twitter social graphs.

5.2.3 Textboxes

To browse inside the contents of the node we make pop-ups inside of the webpage. With this solution, we don't have to leave the main page we can see the main graph and the content od a node at the same time. This textbox can be full sized of the blank and we can choose how many percent of the page wanted to be shown by editing the window size.



Optimization (disambiguation)

From Wikipedia, the free encyclopedia

Optimization, optimisation, or **optimality** may refer to:

- Mathematical optimization, the theory and computation of extrema or stationary points of functions
- Engineering optimization, the use of optimization techniques to aid the engineering design process

Economics and business

- Optimality, in economics
  - Profit
  - Utility
  - Economic efficiency
  - Pareto optimality, or Pareto efficiency, a concept used in economics, game theory, engineering, and the social sciences
- Process optimization, in business and engineering, methodologies for improving the efficiency of a production process
- Product optimization, in business and marketing, methodologies for improving the quality and desirability of a product or product concept

Information technology

- Program optimization, improving software to make it work more efficiently or use fewer resources
- Compiler optimization, improving the performance or efficiency of compiled code
- Asymptotically optimal algorithm, an algorithm that is at most a constant factor worse than the best possible algorithm for large input sizes
- Search engine optimization, in internet marketing, methodologies aimed at improving the ranking of a website in search engine listings
- Image search optimization, in internet marketing, methodologies aimed at improving the ranking of an image in image search engine listings

Other

- Optimality theory, in linguistics, a model proposing that observed forms of language arise from the interaction of conflicting constraints
- Optimization (role-playing games), a gaming play style

Figure 5.3 Example of a textbox

5.3 Editable surface

If we have a recommendation, for example, to add a keyword for a publication, vis js have a function to add, edit or delete edges and nodes. These functions can improve social recommendation system.

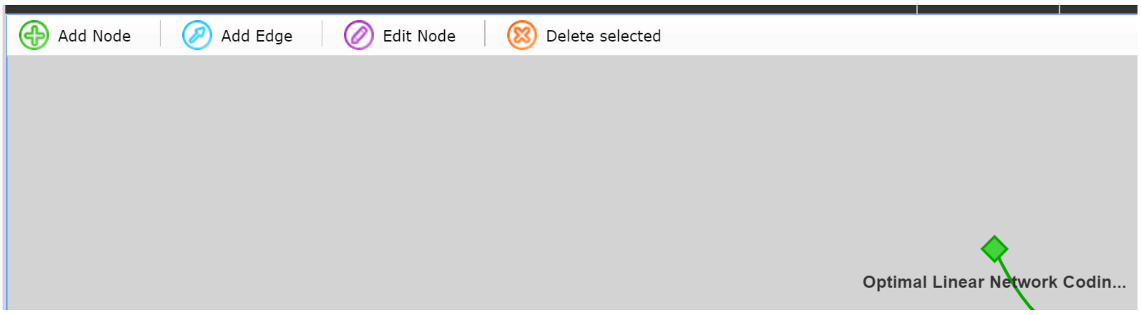


Figure 5.4 The functions in the toolbar

5.3.1 Add node or edges

You can use these options to switch certain functionalities on or off of attach a handler function to them. These functions are called before the action is performed. If a node is going to be added through the manipulation system, the addNode function will be called first. With this, you can provide a guide for your users, abort the process or anything else you want to do. For all except the editNode functionality, these handler functions are optional.

When you supply a boolean, you only toggle the 'add node' button on the GUI of the manipulation system. The lack of handling function could effect the API when using the methods. When a function is supplied, it will be called when the user clicks the canvas in 'addNode' mode. This function will receive two variables: the properties of the node that can be created and a callback function. If you call the callback function with the properties of the new node, the node will be added.

This function will receive two variables: the properties of the edge that can be created and a callback function. If you call the callback function with the properties of the new edge, the edge will be added.

5.3.2 Edit node or edges

Editing of nodes is only possible when a handling function is supplied. If this is not the case, editing of nodes will be disabled. The function will be called when a node is selected and the 'Edit Node' button on the toolbar is pressed. This function will be called like the addNode function with the node's data and a callback function.

If boolean, toggle the editing of edges in the GUI. If a function is supplied, it will be called when an edge is selected and the 'Edit Edge' button on the toolbar is pressed. Initially, the endpoints of the edge may be dragged to connect to a different node, then the function will be called in the same way the addEdge function is called. If an object, if a function is given for the 'editWithoutDrag' property, the function will be called immediately (without dragging any endpoints) in the same way the addEdge function is called. If the callback is not performed, the edge will remain hanging where it was released. To cancel, call the callback function with null as an argument or without arguments.

5.3.3 Delete

If boolean, toggle the deletion of nodes in the GUI. If function, it will be called when a node is selected and the 'Delete selected' button is pressed. When using a function, it will receive a callback and an object with an array of selected nodeids and an array of selected edges Ids. These are the items that will be deleted if the callback is performed.

Boolean or Function true If boolean, toggle the deletion of edges in the GUI. If function, it will be called when an edge is selected and the 'Delete selected' button is pressed. When using a function, it will receive a callback and an object with an array of selected nodeids (empty) and an array of selected edges Ids. These are the items that will be deleted if the callback is performed.

5.4 Information of nodes

The nodes are the results of the search query. We can see the range of the expression we are looking for but the only thing we can see of nodes is the title. To have specific information about the nodes we have to use a textbox.

5.4.1 Multiple contents

If the type of the nodes is a publication in textboxes we can show the user the research in pdf or the abstract in another type of content. We can add our point of view of the topic to the nodes or an URL connected to the topic. In keywords, the base content in the textbox is the URL of the Wikipedia page if its exist. In all type of nodes, the users can add contents and the community will be the judge about the truth value with the confidence points.

6 Future Work

Between 2008 and 2014, the number of scientific articles cataloged in the Science Citation Index of Thomson Reuters' Web of Science grew by 23%, from 1 029 471 to 1 270 425. Growth was strongest among the upper-middle-income economies (94%), primarily driven by growth in Chinese publications (151%)[12]

The knowledge of humanity growing exponentially. With ScienceMesh we can improve the experience of research even if the database will be enormous in the future.

6.1 Make a new trend

The graph visualization is simple, we can explain that for the people in two minutes. This type of visualization is more effective also in cooking in financial in almost all part of life.

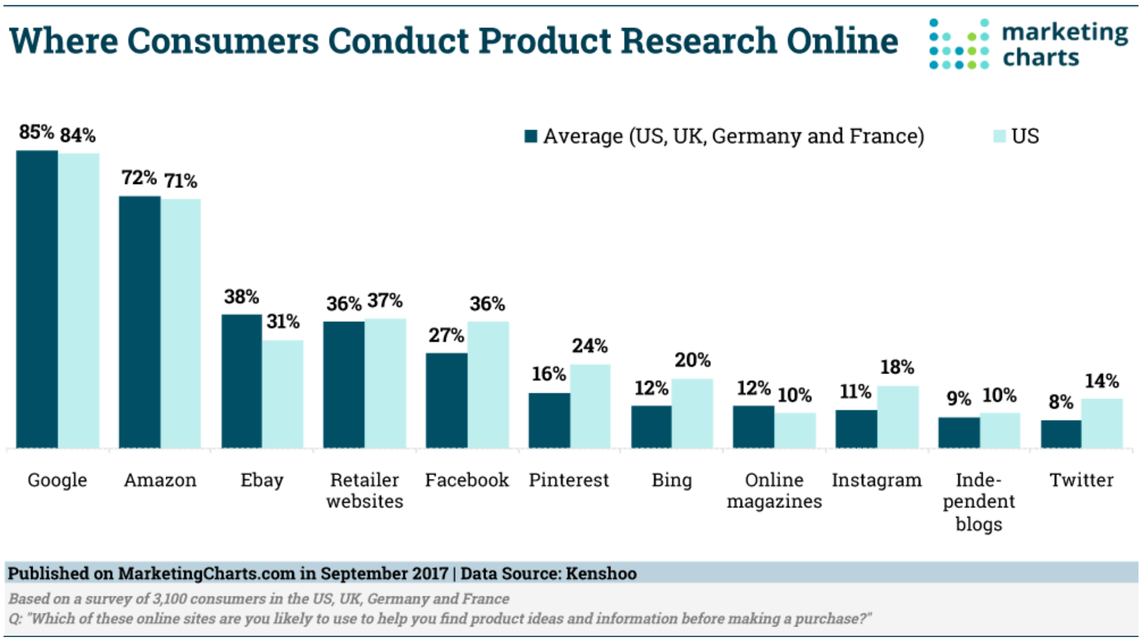


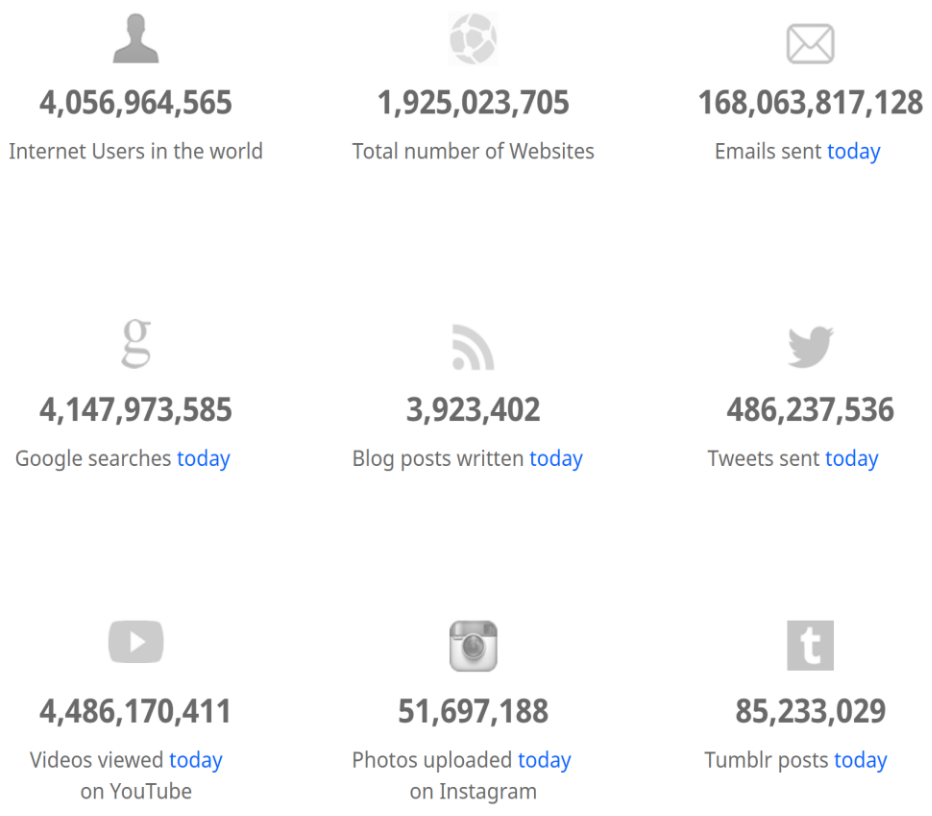
Figure 6.1Which of these sites are you likely to use to help you find product ideas and information before making purchase?

6.1.1 Statistics

There are over 1.5 billion websites on the world wide web today. Of these, less than 200 million are active. The milestone of 1 billion websites was first reached in

September of 2014, as confirmed by NetCraft in its October 2014 Web Server Survey and first estimated and announced by Internet Live Stats.[13]

Internet Live Stats claims there's around 5,5 billion Google searches per day or more than 63,000 search queries per second as you're reading this.[14]



4,056,964,565
Internet Users in the world

1,925,023,705
Total number of Websites

168,063,817,128
Emails sent today

4,147,973,585
Google searches today

3,923,402
Blog posts written today

486,237,536
Tweets sent today

4,486,170,411
Videos viewed today on YouTube

51,697,188
Photos uploaded today on Instagram

85,233,029
Tumblr posts today

Figure 6.2 Growing fast

6.2 Financial

Search platforms are the best place to advertise. If this project will grow more and we can use also in everyday life we can sell places on the webpage for advertising. With this, we can finance the project.

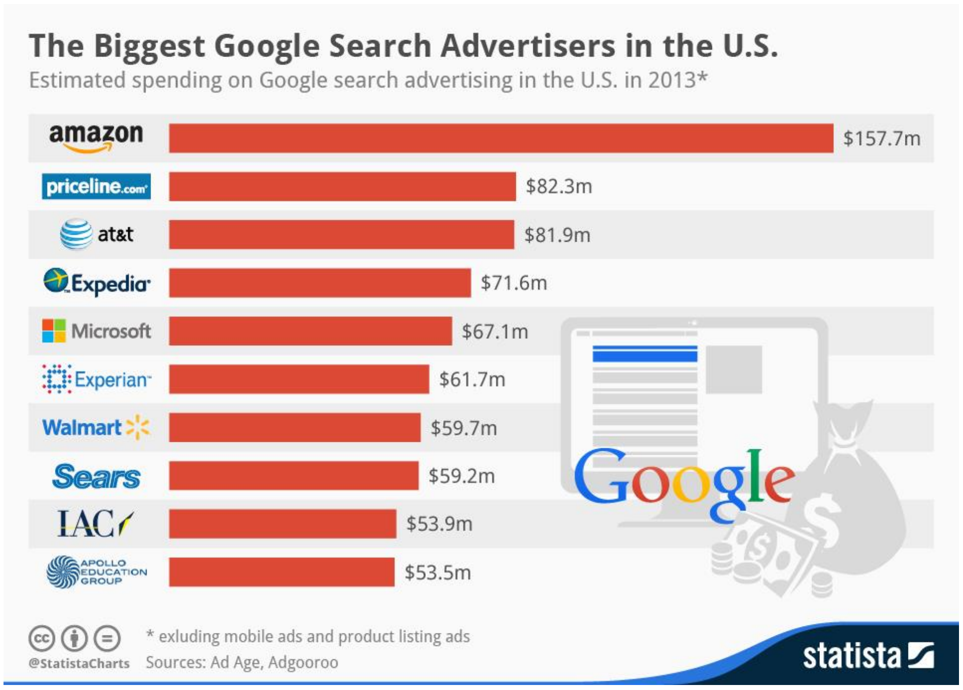


Figure 6.3 Search platform advertising fundations

6.3 Acknowledgment

Special thanks for Janos Tapolcai, Ferenc Kukucska, and Daniel Kis-Pal.

Bibliography

[1] Leonhard Euler Biography, https://www.biography.com/people/leonhard-euler-21342391 (Last updated May 12, 2014)

[2] Why you should power your application with graph visualization,https://cambridge-intelligence.com/why-you-should-power-your-application-with-graph-visualization/,(2018 May 18)

[3] TDK main page, https://tdk.bme.hu/Browse

[4] Request library,Python , http://docs.python-requests.org/en/master/

[5] Documentation of BeautifulSoup, https://www.crummy.com/software/BeautifulSoup/bs4/doc

[6] ScienceBeam Github repo, https://github.com/elifesciences/sciencebeam

[7] MongoDB webpage, https://www.mongodb.com/what-is-mongodb

[8] Neo4j graph database,https://neo4j.com/product/

[9] MySQL relational database, https://www.mysql.com/products/

[10] Vis javascript libary documentation, http://visjs.org/docs/network/index.html

[11] Flask documentation, http://flask.pocoo.org/

[12] Research on search,https://en.unesco.org/node/252282

[13] Statistics on websites, http://www.internetlivestats.com/total-number-of-websites

[14] Statistics on search, http://www.internetlivestats.com/total-number-of-websites