Contents

[1 Introduction 3](#_Toc91158675)

[1.1 Overview 3](#_Toc91158676)

[1.2 Brief History of Computers and software 4](#_Toc91158677)

[1.3 Problem statement 4](#_Toc91158678)

[1.4 Objectives of the research 4](#_Toc91158679)

[1.5 Summary 5](#_Toc91158680)

[2 The role of Software in Scientific research 6](#_Toc91158681)

[2.1 Introduction 6](#_Toc91158682)

[2.2 General roles of software in a research 7](#_Toc91158683)

[2.3 Domain specific examples 8](#_Toc91158684)

[2.4 The role of software in research breakthroughs 9](#_Toc91158685)

[2.4.1 Breakthrough - Visual Representation of a black hole 9](#_Toc91158686)

[2.4.2 Breakthrough - Visualization of gravitational waves 9](#_Toc91158687)

[2.4.3 Breakthrough - accelerated drug discovery 10](#_Toc91158688)

[3 Software usage purpose 11](#_Toc91158689)

[3.1 Introduction 11](#_Toc91158690)

[3.2 Software purposes in the literatures 11](#_Toc91158691)

[3.3 Software purposes in ontologies 12](#_Toc91158692)

[3.3.1 WikiData 12](#_Toc91158693)

[3.3.2 The software ontology (SWO) 14](#_Toc91158694)

[3.3.3 OntoSoft 14](#_Toc91158695)

[3.3.4 Other sources 14](#_Toc91158696)

[3.4 Types of software purposes 14](#_Toc91158697)

[3.4.1 Data collection 15](#_Toc91158698)

[3.4.2 Data pre-processing 15](#_Toc91158699)

[3.4.3 Data Analysis 15](#_Toc91158700)

[3.4.4 Data visualization 17](#_Toc91158701)

[3.4.5 Simulation 17](#_Toc91158702)

[3.4.6 Stimulation 17](#_Toc91158703)

[3.4.7 Modelling 17](#_Toc91158704)

[3.4.8 Programing 17](#_Toc91158705)

[4 Annotation of Data 18](#_Toc91158706)

[4.1 Data set 18](#_Toc91158707)

[4.2 BRAT Annotation tool 18](#_Toc91158708)

[5 Feature selection 19](#_Toc91158709)

[5.1 Phrase 19](#_Toc91158710)

[5.2 Sentence 19](#_Toc91158711)

[5.3 Paragraph 19](#_Toc91158712)

[5.4 Dependency tree 19](#_Toc91158713)

[6 Selection of classifiers 20](#_Toc91158714)

[7 Classification and Evaluation 21](#_Toc91158715)

[8 Summary of Results and Conclusion 22](#_Toc91158716)

[8.1 Summary of Results 22](#_Toc91158717)

[8.2 Conclusion 22](#_Toc91158718)

[8.3 Limitations and future work 22](#_Toc91158719)

# Introduction

## Overview

**What have been done and why it is important?**

**With what resources?**

**What is the structure of the work?**

* Analysis of research papers can give a lot of insights about software resources and their dependency.
* In a scientific research different kinds of input resources are used. One of such input is a software.
* Used resources in a research are typically mentioned in a citation. Citation practices of formal articles in a research are matured and various citation styles exist. Even if principles for formal citation of a software has already been put out, most scientists are not properly citing resources.
* Surprisingly, sometimes researchers do not mention the type of software they used entirely or mention it with a rather vague abbreviation and just talk about the results they have obtained.
* As long as software is mentioned using formal methods, like RRID, it is possible to perform citation analysis using regular expressions which can be constructed to capture the pattern of citation.
* Though regular expression based analysis can give basic insights about the software citation it has limitations because:
  + Not so many authors use formal citation of software, like RRIDs
  + Even if scientists use formal citations, they may fail to properly follow the guidelines. For example, some authors tend to ignore the RRID-part and that creates an ambiguity by it self that it is not possible to know weather the author is actually making a software citation or it is completely something else.
  + Rule based method fails to capture context information and ignores dependencies. It is not possible to be sure about the authors intention whether or not using a software citation.
* At the same time pattern based analysis, like using regX, is not suitable to extract information about software citation, for instance the particular use of a software, especially when a software mention statement lacks any form of formality where the information is concealed in a natural language description.
* Therefore it is required to automatically extract the purpose of software use in scientific literatures. This might help to answer questions like:
  + What type of software is being frequently used for what purpose in a specific area of research? This also allows to find an answer further question like what is the most common technique researchers follow when trying to solve a given research problem in a given domain )
* Previous attempts to automatically extract information using machine learning techniques, specifically supervised machine learning technique, about the software use purpose was constrained mainly because of lack of ground truth data. But this time, with the advent of SoMeSci, it is possible to do so.

## Brief History of Computers and software

If we say that computers are for computation, then an idea of using artifacts for computation goes back to 2400 B.C. to the ancient Babylon where abacus was used for arithmetic tasks. From the 11th century and onwards, such computation artifacts evolved to became mechanical devices and were used for solving problems and to perform studies, for instance, astronomy. Though so many artifacts have been developed, it was on early 19th century that the first analytical general purpose *“mechanical computer”* was made by Charles Babbage whom is considered to be the “*father of computer*”. Parts of the machine include ALU, control flow with conditional branching and loops, and even integrated memory. The machine was programmable via data provided in the form of punched cards. Outputs of the machine, analogous to the modern computers printer and speaker, were curve plotter and a bell respectively. After that computers transitioned from mechanical to being electromechanical and soon later to purely electronic devices.

In 1936, the first modern computer known as a universal Turning Machine was proposed and brought new concepts like a computer program ( execution of instructions) and a concept of memory where the instructions are stored. Then sooner transistors and then integrated circuits have emerged giving rise to a modern computer that we use today.

From the history of computers, it is clear that computers have been used for reckoning and computation from day one. However, memory and data storage capability of computers emerged on the later part of the computer history. Thus, any modern computer can be considered as a data processing and storage device.

Based on a context that “computers are data processing and storage” devices, the functionality of computers as data processing machines can be referred to software where as storage of data implies to hardware functionality.

## Problem statement

**Why do software purpose analysis ? what is the problem or for what purpose can we use it?**

## Objectives of the research

**Why do you intend to accomplish in this project? Solve the problem ?**

This work has the following objectives:

* List down the purpose of software usage in a research in a hierarchical manner.
* To extend SoMeSci with a manual annotation of purpose of software usage.
* To select feature for the training model
* To select a classifier and train a model.
* To evaluate and optimize results

## Summary

# 

# The role of Software in Scientific research

## Introduction

Nowadays scientific research is unthinkable without a use of software and investigations in various areas of science are becoming increasingly reliant on software tools {goble2014better, storer2017bridging, hannay2009scientists, jimenez2017four}.

A software is very important asset for building a scientific knowledge and more discoveries in a research are made possible than ever by a use of software tools that automate processing of huge amount of data {jimenez2017four}. Typically a software is used in a research for data processing tasks such as data analysis, modeling, simulation, control processes, knowledge dissemination, etc. {hannay2009scientists, pan2016disciplinary}.

In modern research, a scientific software is as important as any lab-equipment {wilson2014best}. However, the development of scientific software is much more complicated and fundamentally different from an ordinary commercial software like accounting software. Scientific software requires specialized domain knowledge for its development and require direct involvement of domain expert or scientist {wilson2014best, segal2008developing}. Due to this, an increasing number of scientists are developing a software as part of their research work or directly taking part in the development process of a research software {jimenez2017four, kanewala2014testing}.

According to surveys conducted in the UK and USA, 2008 and 2017 respectively, most scientists agree that software plays an important role in their research work {hettrick2014uk, nangia2017track}. Participants of the survey, in UK, were 2000 researchers working in various areas of science in roles ranging from student to senior academic staff whereas participants of the survey, in USA, were members of the US National Postdoctoral Association.

The results from of UK survey {hettrick2014uk} indicate that :

* 38% of researchers spend at least 20% of their time developing a software.
* Almost half of scientists spend more time creating software as part of their research work than five years ago .
* Over 50% of survey respondents reported that they develop their own software.
* Nearly 70% claim that their research directly depends on use of a software &
* Over 90% of scientists say software is important for their research.

The results from of USA survey {nangia2017track} indicate that:

* Over 90% of scientists use software.
* 63% of respondents state that their research is impossible with out using software.
* 31% of scientists say that they could do their work without using a software but more effort would require.
* Only 6% of survey respondents say that there would be no significant difference in their task if they do not use software.

Overall, results from the two surveys clearly indicate that software is pervasive in scientific investigations and many researchers use as well as develop a software for their research.

Even though software plays an important role in a modern research, usually the contributions of software is understated. This can be seen from the poor citation practice of software in research papers across several fields of research {yang2018important, pan2016disciplinary}. In attempt to promote the recognition of the roles of scientific software in a research, the ReSA has collected literatures that evident the roles software play in a research, at Zetoro group library. The main aim of ReSA is to influences decision makers to properly attribute contributions of a research software and give credits to its developers.

The next section presents more details about the role of software: in general, in specific domains, and in research breakthroughs.

## General roles of software in a research

Software is playing crucial roles in a research and making a shift in a research culture in terms of enabling automation of analysis pipelines, creation of new ways of analysis via computational models, supporting sophisticated analysis of large volume of data, documentation of a research, etc. {jay2020software}.

Some of the most general roles of a software in a research are:

* Software helps to explore und understand a research problem {hannay2009scientists}.
* Results from a scientific software is presented as an evidence to support a research conclusion {kanewala2014testing}.
* *A software dictates the quality of a research outcome* {hannay2009scientists}. Outcome of a research becomes unreliable or even useless if there is an error in the software {soergel2014rampant}. For example, several scientists retracted their scientific publications up on a retrospective discovery of a bug in their software {wilson2014best,merali2010computational,miller2006scientist}. A more palpable failure of a research ambition due to an error in the software, for instance, is the failure of *Ariane rocket* in 1996 {enwiki:1054482061}.
* A software also helps to document a research process and to *validate results of a given research* {jay2020software}. Executable cells in a Jupyter notebook is one real world example where a software can be used to validate a research result.
* Software allows experiments to be made beyond constrains of the physical world. This is because experiments that run on a computer are not limited by processes that occur in nature but only by the laws imbedded in the computer code {wolfram1984computer}.

## Domain specific examples

A software is being extensively used for a research in various areas of science such as physics, chemistry, space science, life science and so on.

The physics research facility, the Large Hydron Collider at CERN, for instance uses a software with more than 5 million lines of code which is used for processing of terabytes of data generated from experiments {storer2017bridging}.

In a nuclear research, a software is being developed increasingly to be used for experiments {yan2017case}. For example, testing a modification in a nuclear weapon can not be done on a field, but instead a software that simulate the impact of modification is usually used {kanewala2014testing}. This is because testing of a nuclear weapon on a field is deprecated by regulations like nuclear test ban treaties (NTBT) in addition to the potential disaster that testing a nuclear weapon poses to the environment and life {enwiki:1053274189}.

In chemistry research, a software can be used to model and simulate chemical processes that are challenging, too complex or expensive to conduct in reality. Karplus and Levitt used computer simulations for their joint-research “the development of multi-scale models for complex chemical systems” and won a Nobel prize in 2013 for their work {storer2017bridging, andre2014nobel}.

In a climate and environmental studies, software is used to make predictions about climate changes. For example historical temperature data can be integrated to make predictions about future temperature variations {storer2017bridging}.

In a space science, space probes heavily rely on software. In this case a software navigates space crafts to other planets, processes and transmits scientific data back to Earth for further processing, helps researchers interpret results, etc.{lutz2011software}.

In medical research and diagnosis, imaging software plays a critical role to assist medical researchers, for instance, for early isolation of cancer cells. The main reason for low chance of survival from cancer is mainly due to late detection of cancer cells in the body. This makes a diagnosis of cancer to be a time critical task and early identification of cancer implies curability of a disease and a higher chance of survival {wagner2004challenges}. Especially on the early stages, it is not straight forward to determine which cells are likely to develop a cancer. For this reason, medical scientists use different types of software to identify cancer cell or to decide weather a tumor is malignant or not. Using a software, they could perform various kinds of analysis and processing on imageries obtained from scans such as MRI or CT Scan {al2012lung}. An example of software that is used for cancer imaging research is DMRI. Such software is extensively used by many researchers, more than 75,000 downloads every year {norton2017slicerdmri}. Therefore, it is clear that software plays a critical role in medicine, to diagnose diseases and ultimately to save life.

Software plays an important role in power system planning and operation as well. One of the major activities in power system operation is contingency analysis. During contingency analysis, engineers determine violations of power grid operation conditions, such as overloading, which might occur when outage of a transmission line or a power generation unit occurs. Contingency analysis helps to understand power system behavior after outages and gives an opportunity to take preventative actions {mishra2012contingency}. Power grids are extremely complex and such kind of analysis tasks are unimaginable with out a use of software. An example of software that is used to perform contingency analysis in the power system operation is Power World software {powerworld.com}.

## The role of software in research breakthroughs

The use of software enabled scientists to produce better scientific discoveries and achieve research breakthroughs {goble2014better}. Although it is not possible to list down all the research breakthroughs, few examples have been presented as follows.

### Breakthrough - Visual Representation of a black hole

One of the research breakthroughs is creation of the very first visual representation of a black hole using an open source software NumFOCUS. To observe a black hole that is 55 million light years away, it would have required to build a huge telescope of size of planet earth. But instead of building one giant telescope, which is not possible any way, hundreds of scientists spent decades of years creating a global network of telescopes, known as Event Horizon Telescope (EHT) {enwiki:1052167868}, synchronized precisely using atomic clocks. The EHT gathered a huge amount of data for years. However there was a lot of noise in the collected data because :

* The EHT was a network of non-similar telescopes.
* The radio signals were coming through attenuated due to atmospheric effect like water vapor, clouds, turbulence … etc.

Therefore the scientists had to use various algorithms and data analysis pipelines. The resulting image from various data processing was compared to ensure the integrity of the result. This huge scientific breakthrough in a space research, can be attributed to mainly the use of powerful data processing software.

### Breakthrough - Visualization of gravitational waves

The other scientific breakthrough that can be attributed to role of software in a research is the detection and visualization of gravitational waves for the first time, using a LIGO software {enwiki:1047100294, mukherji2017report}.

### Breakthrough - accelerated drug discovery

Software accelerates drug discovery {bhati2021pandemic}.

# Software usage purpose

## Introduction

In scientific investigations broad range of software is being employed for various purposes [12]. In terms of size, software ranges from simple scripts to extremely complex software with millions of lines of code. In terms task, a software can be used for execution of rudimentary tasks to computation of extremely complex ones. Typical examples of purpose of software use for scientific investigation are simulation, modelling, data analysis, etc. {goble2014better}.

To be able to automatically identify, from context, for what purpose a software is used in a scientific text, a classifier algorithm has to be trained on a manually annotated dataset that indicate software usage purpose. However, the list of potential software usage purposes has to be identified before hand so that it can be used for the annotation purpose.

To enumerate possible software purposes, mainly two things have been done in this thesis. First manual analysis of scientific publications in SoMeSci dataset and second analysis of software ontologies such as WikiData, DBPedia, Ontosoft, …etc. Once a list of potential software usage purposes has been identified, the list has been consolidated further to create a more general list software usage purposes for the sake of convenience during annotation.

This section presents how various software usage purposes has been identified, enumerated and consolidated into a more general groups for annotation of the dataset which will be used for fitting a model that will automatically identify purpose of a software use.

## Software purposes in the literatures

In a research, scientists follow scientific method to discover knowledge. Typically, scientists begin with a question and attempt to answer questions through a research and propose hypothetical answers for their questions. Then, they test the proposed hypothesis by conducting various experiments. Although all scientists do not follow the exact same step, the over all idea remains the same {enwiki:1061107378- Scientific method}. This is where a software use comes into play, aid scientists during their experimentation. Therefore, the analysis of literatures when looking for software usage purpose is aimed at answering “*for what purpose scientists are using a software ?”* in their experiments. Accordingly, some key words that reflect potential software usage purposes have been identified from the literature and listed on the following table:

|  |  |
| --- | --- |
| * Comparison of experimental groups * Quantification * Measurements * Analysis * Mapping * Correction of mapping * Generate scaffolds * Generate trees * Search sequences * Map * Predict gene structure * Align gene * Filter * Evaluate * Select * Optimise * Classify | * Statistical analysis * Data analysis * Densitometric analysis * Voxel-based Analysis * Cross-sectional ROI analysis * Gene analysis * Gene assembling * Construct contigs * Fill gaps * Generate assembly * Calculate or determine a value * Draw heat map * Validate * Annotation * Fit or train a model * Sketch * Identify |

The list of key words in the above table is used to delineate possible software usage purposes, however, it is cumbersome to enumerate all possible software purposes manually. To supplement results obtained from the analysis of literature, software ontologies have been analyzed.

## Software purposes in ontologies

Ontologies are controlled vocabularies that provide formal naming and definition of properties and relation between concepts, entities, data etc. Ontologies are specialized to a specific subject matter and every academic discipline creates ontologies to organize data into useful knowledge {enwiki:1060388948}.

Effective knowledge representation begins with analysis of ontologies with in the domain of interest {chandrasekaran1999ontologies}. Accordingly, analysis of software ontologies have been done to find out possible software usage purposes. The software ontologies, that has been analyzed on this project are WikiData, DBpedia, SWO (the software ontology), OntoSoft and codemeta.

### WikiData

Wikidata is a multilingual knowledge graph that is curated collaboratively by a Wikimedia community and serves as a freely available common source of structured data for everyone {enwiki:1060114687, enwiki:1060408581}.

Wikidata was created by Wikimedia foundation mainly to store meta data that can be used for other Wikimedia projects such as Wikipedia. Interestingly, wikidata is allowed to contain inconsistent and contradicting facts in order to embrace the diversity of knowledge about a given entity {vrandevcic2012wikidata}.

Although wikidata has a tremendous amount of data in it, there was no information that would indicate software usage purposes, rather information about software categories was found. Therefore, an indirect approach has been taken to list down possible software purposes from software categories by assuming each software category has essentially a software purpose associated to it.

Wikidata comes with a bunch of tools like, SPARQL end point, query builder, data visualization tools, etc. Thus a SPARQL end point has bee utilized to query a list software and their potential categories {Take a look at the SPARQL query used on the Appendix}.

Over 400 software categories have been found from the Query. To find out potential relation between these categories and to select more general software categories, a network analysis has been done using Gephi software. Using Gephi software, clustering of related software categories and filtering has been made to have a more generalized software categories. The procedure for network analysis has been described as follows:

1. First query result from the SPARQL terminal of wikidata has been downloaded in a csv file format.
2. Then, the csv file has been opened with Gephi software (version 0.9.2) as “undirected graph”. This renders a network graph with overlapping nodes and edges.
3. To unravel the overlapping nodes for visibility, the lay-out of the graph is then changed to “Fruchterman Reingold”.
4. To find out possible clusters from the network, from the list of statistical tools, “Modularity” has been run. Then partition of nodes and edges has been done using “Modularity class”.
5. Then to adjust size of nodes based on importance, node size ranking has been done with a “Degree” parameter with {minimum, maximum} size of {20, 80} respectively.
6. Then to select the most prominent nodes, among filter tool “Degree range” filter has been used. The Degree range filter estimated prominence of the nodes between values of {1, 60} where the maximum value indicates the most prominent node.

Text

Description automatically generated

Text

Description automatically generated

Degree {3, 60}

Degree {1, 60}

Timeline, map

Description automatically generated

Diagram, map

Description automatically generated

Degree {7, 60}

Degree {5, 60}

According to the network analysis, major types of software categories are:

* Application software
* Utility software
* Computer security software
* System software
* Client
* Programming tool
* Software Library
* Software framework
* Editor
* Science software.
* Graphics software
* Computer aided design software
* Science software
* Mathematical software
* Communication software

It is clear that these software categores are reated to each other and it is possible to group related ones to create more general software categories. Based on manual analysis of wikidata, mathematical software, for instance, is subclass of science software and science software is subclass of application software. Similarly, communication software is subclass of application software, etc. By analyzing the relation between the software categories, overall three main types of software categories have been identified as follows:

1. Application software
2. System software
3. Software Component

A simplified version of relation between software catefories is depicted on the grapgh below.

#### Identifying software purpose

The main aim of software category analysis of wikidata was to find possible software usage purposes from each software category. However, at the top level of hierarchy of software categories, it is difficult to identify software usage purposes. For these reason, software categories at the second level of the tree has been queried from wikidata for each type of software sub-category. From there possible software usage purpose has been derived and summarized on the table below.

An application software, a.k.a app or application program, is a computer program that is designed to carry out a specific task other than operation of a computer and typically made for end-users {enwiki:1060918552}. There are different types of application software that are being used for various purposes. The following table summarizes types of application software and their purpose.

**Horizontal software vs vertical software**

Based on purpose, application software is either horizontal software or vertical software {enwiki:1034388659, enwiki:879502666}. A horizontal (market) software is a kind of application software that is more versatile, generic and useful in wide range of industries. Examples of horizontal type of software are word processors, spreadsheets, web browsers, calendar applications, etc.

On the other hand a Vertical (market) software refers to a software type purposed to address needs of a specific niche in a business, research, or even a specific department within an organization { enwiki:879502666}.

|  |  |  |
| --- | --- | --- |
| Types of Application software  (sample) | Software purpose | Examples |
| Search tool, Browser | * Search * Data mining | * Web browsers, * search engines, * fuzzy finder, etc. |
| Remote sensing software, | * Data collection * information gathering | * Google earth, * OpenEV * ENVI |
| Econometrics software | * Data Analysis | * Stata * R * SATA * SPSS |
| Network simulator | * Simulation | * OPNET * NetSim * GloMoSim |
| IDE , text editors | * Programming * text editing | * Visual Studio, NetBeans, * Atom, Sublime, Vim … etc. |
| Genealogy software | * Record data, * Organize & publish data | * Family Tree builder * Legacy |
| computer-aided design software | * Modelling * Analysis & optimization | * AC3D, SolidWorks * AutoCAD, CATIA, … etc. |
| science software | * Simulation * Modelling * Data Analysis * Visualization * Calculation | * Bioinformatics software, * data analysis software, * plotting software, * chemistry software, * mathematical software, * astronomy software |
| database application | * Retrive, Insert, modify, delate data |  |
| communication software | * communication | * Email, instant messaging, etc. |
| graphics software , animation software | * 3D modelling, * visualization | * 3D computer graphics software, |

From the analysis of software categories, typical software usage purposes are:

* Data recording or collection
* Data Analysis
* Visualization
* Simulation
* Modelling
* Programming.

### The software ontology (SWO)

The software ontology (SWO), particularly describes software used, for preparation and maintenance of data, within fields of computational biology and bioinformatics. The SWO was primarily developed to improve reproducibility by providing detailed description about software used for biomedical investigations {malone2014software}.

SWO was found on ontology search (OLS) website and was examined for possible software purposes. Unlike wikidata, a list of possible software purpose were found directly in “browse terms” section. To navigate to the software purpose list the following steps has been taken: “Browse terms”> “entity“>”occurrent”> “planned” >”planned process”. The software usage purpose in the SWO has been presented in to two main groups as “data transformation” and “data visualization”. Under data transformation, 40 sub -types of data transformation software purposes are listed.

Graphical user interface, website

Description automatically generated

Some examples of software usage purposes retrieved from SWO are shown on the table below:

|  |  |
| --- | --- |
| * Data transformation * Annotation * Text editing * Modelling * Curve fitting * Simulation * Query and retrieval | * Calculation * Analysis * Data visualization * File rendering * Matrix manipulation * Data mining task * Clustering task |

### OntoSoft

Onosoft is a software registry framework that stores important metadata about software to foster reuse and sharing of software among scientific community. The ontology provides descriptions about a software that would help scientists to identify, understand, execute, and do research with a software. Moreover, it helps scientists get information about update and support for the software. These descriptions are visualized in a 6 dimensional pie-chart, with each slice indicating the completeness of the description. Particularly, Ontosoft focuses on the geoscience because software resources are not being shared adequately in that field {gil2015ontosoft}.

Graphical user interface

Description automatically generated

The type of information provided in each dimension of description entries are summarized in the table below:

|  |  |
| --- | --- |
| Dimension | Description |
| Identify | * Name of software, abbreviation of the software, etc. |
| Understand | * Creator of the software, publisher of the software, * *domain specific key words* |
| Execute | * URL for downloading the software, license, system requirements …etc. |
| Do Research | * Input / output file formats, preferred citation information |
| Get support | * Contact details, possible support included, etc. |
| Update | * Version, developer community, software development process , maintenance, etc. |

From the set of information provided in those 6 dimensions, particularly the “understand” dimension has nearly 400 domain specific key words that would potentially indicate software purposes. Therefore, those has been retrieved and condensed into a more general software purposes list of 60.

### Other sources

The other sources of information analyzed for the software purpose are DBpedia and Sci-crunch.

## Types of software purposes

Based on the analysis results form the software ontologies and scientific literatures in SoMeSci, a list of software purposes has been identified and consolidated into a group of 8 main types of software purposes as follows:

1. Data Collection
2. Data pre-processing
3. Data Analysis
4. Data visualization
5. Simulation
6. Stimulation
7. Modelling
8. Programing

### Data collection

Scientists collect data for their research using various means from using paper and to sophisticated software to collect data from various sources.

* https://en.wikipedia.org/wiki/Data\_collection
* Using electronic surveys to collect data: experiences from the field
* Preparing and conducting interviews to collect data
* Recording, Archiving, and Using WSR-88D Data
* Incremental Organization for Data Recording and Warehousing
* [Automatic indexing](https://en.wikipedia.org/wiki/Automatic_indexing)
* [Web Crawler](https://en.wikipedia.org/wiki/Web_crawler)

### Data pre-processing

* <https://en.wikipedia.org/wiki/Data_pre-processing>
* Sequence alignment

Often times after a data is collected, it has to go through a process of processing before it is actually used. This is

### Data Analysis

In a modern research, where a research is increasingly relying on processing of huge amount of data, the most common purpose of software usage purpose is to perform data analysis.

When it comes to the application of data analysis in actual research works, various kinds of data analysis techniques exit. Some of the data analysis techniques can be more general where as others are more domain [specific](https://pubs.acs.org/doi/pdf/10.1021/ac00238a008) .

In a data driven science, one of the most important software use case is for Data analysis. Usually a huge amount of data is analyzed using mathematical or statistical methods. Further more, domain specific data analysis techniques exist.

The data to be analyzed using a scientific software also has a broader range. Data could be something obtained from a sensor, an image retained from a microscope, a data generated from a random walk, data generated from simulation of a model. … etc

Data analysis is a broad term which can refer to inspecting, cleaning, transforming, modelling data, etc. with a particular goal of discovering a meaningful information from the data which can be used to make conclusions or decisions [13] .

Modern research is increasingly data driven and software is being used extensively to analyze large amount of data. Using a data analysis software, scientists usually inspect, clean, transform and model data in search for meaningful information from the data which will in turn will support conclusions for a research [13].

Diverse types of data analysis techniques exist, some of which are more general and applicable to any field of research and science. But there are also some data analysis techniques which are endemic to only some domains. For Example, densitometric analysis. More general data analysis techniques are usually mathematical, like statistical data analysis, numerical analysis and so on.

The use of software for data analysis has several advantages. One is, a data analysis software gives more efficient and effective work. It allows analysis of a large volume of data. The other is, data analysis software gives insights hidden in a data, correlation between variables, etc.

We have diverse types of data analysis software. However, possible two main categories of data analysis software are Qualitative Data analysis software and Quantitative Data analysis software. Qualitative data refers to any from of data obtained by a researcher by direct observation, interviewing, recordings, etc. (<https://en.wikipedia.org/wiki/Qualitative_research> ). Qualitative research software is particularly popular in social science research where most of the data is qualitative by its nature. Qualitative data analysis software are collectively known as Computer Assisted Qualitative Data Analysis software ([CAQDAS](https://en.wikipedia.org/wiki/Qualitative_research#Data_analysis)) .

The other type of data analysis is quantitative which is based on numbers.

For example: to perform Data analysis (might refer to inspecting, cleaning, transformation, and modelling a data). The main purpose of data analysis is to extract meaningful information from a data that will help to make businesses operate more effectively [13].

* <https://en.wikipedia.org/wiki/Data_analysis>
* Data analysis in qualitative research
* Types of Data analysis
* Mining Scientific Data
* Network Analysis
* Statistical Analysis
* Numerical Analysis
* Data Mining – extraction / discovery of patterns in large data sets using ML, Statistics and Data.
* Regression Analysis
* Mathematical analysis
* Numerical Analysis
* Statistical Analysis
* Domain specific analysis
* Densitometric Analysis
* Voxel-based Analysis
* Continency Analysis in power systems reliability
* Image analysis ( computer vision)
* Data mining

### Data visualization

* https://en.wikipedia.org/wiki/Data\_visualization

### Simulation

* <https://en.wikipedia.org/wiki/Simulation>
* Simulation
* Flight simulation
* Event simulation
* Flood dynamics simulation
* Numerical simulation
* Simulations are run to improve understanding of a problem (segal2008developing).
* [Physics engine](https://en.wikipedia.org/wiki/Physics_engine)
* Simulation of vehicle schedule with any logic

### Stimulation

* https://en.wikipedia.org/wiki/Stimulation

### Modelling

* <https://en.wikipedia.org/wiki/Scientific_modelling>
* Thermal modelling and characterization in chips
* Modelling
  + [Graphics software](https://en.wikipedia.org/wiki/Graphics_software)
  + animation software

### Programing

* https://en.wikipedia.org/wiki/Computer\_programming

In a scientific investigation scientists use software for various purposes. The use cases of software in a research ranges from execution of some trivial tasks to execution of more critical tasks that will determine a research result [38].

# Annotation of Data

## Data set

## BRAT Annotation tool

# Feature selection

## Phrase

## Sentence

## Paragraph

## Dependency tree

# Selection of classifiers

# Classification and Evaluation

# Summary of Results and Conclusion

## Summary of Results

## Conclusion

## Limitations and future work