DSC 202 Final Project

By : Sagarika Sardesai, Tejo Nandini, Mansi Sharma

Introduction - Project Definition

CONTEXT: Many countries and associations work largely to promote patents, businesses and innovation. This is greatly indicated by SBIR Awards which is given by the US government to small businesses as a support for getting established. The Small Business Innovation Research (SBIR) Program is a highly competitive three-phase award system which provides qualified small business concerns with opportunities to propose innovative ideas that meet the specific research and research and development needs of the federal Government.

RELEVANCE : If your product addresses a government need, there is a pretty good chance that it meets a commercial need in commercial markets, too. Can serve as an attraction point for future investors.

Intention

ACCOMPLISHMENT OF THE APPLICATION: (basically figure out where to invest)

Identify the activity trends in different technology sectors (patentdb)

Identify companies and the number of SBIR projects under their wing

Identify the trend of the investment by government agencies and branches for these SBIR companies.

Answer the question - Should commercial companies keep an eye on SBIR companies, based on the investments in SBIR companies

Source of the dataset

The datasets were obtained from the remote dataset provided by Prof. Amarnath Gupta, by using the provided credentials.

Since the SBIR dataset played the primary role in our application, we derived our use cases keeping that in mind.

Patentdb contains the information regarding patents created by the inventors belonging to a country over the years. It also contains additional information related to technology being used (classname) , a short abstract from the published paper and whether or not the patent assignee has changed.

Sbir_award_data contains information regarding awards given out by several agencies (from several branches) of the US Government and additional data such as award amount, year it was given out to, complete information about that company etc.

SBIR Data

Column name	Data type	Description
RecordID	bigserial	Unique identifier for the record
Company	text	The name of the company that received the grant
Award Title	text	The title of the grant award
Agency	text	The name of the agency that has awarded the grant to the company
Branch	text	The exact branch (department) within the agency that has awarded the grant to the company
Phase	text	The phase of the grant (eg. Phase I would mean initial phase)
Program	text	The program the grant falls under
Agency Tracking Number	text	Self-explanatory
Contract	text	Contract number of the grant
Proposal Award Date	text	Self-explanatory
Contract End Date	text	Self-explanatory
Solicitation Number	text	Self-explanatory
Solicitation Year	integer	Self-explanatory
Topic Code	text	Code for the topic of the abstract
Award Year	integer	Self-explanatory
Award Amount	text	Self-explanatory
Duns	text	Identification for the grant

And some more columns...

PatentDB Data

Column name	Data type	Description
patentid	varchar(100)	Unique identifier for the patent
application_number	varchar(100)	Application number for the parent application
inventor_name	varchar(1000)[]	Author(s) of the patent
assignee_name_origin	varchar(2000)[]	Organization(s) originally owning the patent
assignee_name_current	varchar(2000)[]	Organization(s) currently owning the patent (since original ownership might change over the course)
time_events	jsonb	All application updates the patent has gone through
cite	jsonb	There are two types of fields in the JSON: ForwardReferences or ForwardReferencesOrig: The list of patents that cited this patent BackwardReferences or BackwardReferencesOrig: The list of patents that were cited by this patent. The patents listed here may or may not be present as an entry in the patentsdb table, but basic information like patentID, date, inventor and title can be found in this field
classification	varchar(2000)[]	An array consisting of classification codes corresponding to the categories and sub-categories the patent belongs to
classname	varchar(2000)[]	An array consisting of the names of categories and sub-categories the patent belongs to
countrycode	character varying[]	An array consisting of the country codes where the patent is valid. Ignore duplicate values.
abstract	text	The abstract for the patent
title	text	The title for the patent
id	serial	The id for the patent
issuing_country	varchar(64)	The country where the patent was issued

And some more columns...

Flow of the application - Methodology

- Data cleaning
- Structure of queries
- Visualization

Data Cleaning

Data cleaning was needed to be done since the data obtained from the source was unclean and not suitable for further processing and analysis.

For the **patentdb dataset**, we extracted the information for US based patents by using the patentids that began with US. This was done because - Our use-cases are relevant to SBIR. This was then exported from the remote server, as a CSV that could be accessed and updated as and when needed.

For **sbir_award_data dataset**, we used PostgreSQL to clean up the data. We first updated the all the null values of the dataset to ‘NA’. Converted the type of award_amount to int, and replaced ()- in the phone numbers to ‘NA’ as this would give us a clean format. We also concatenated address1 and address2 as a single address. These were cleaned for ease of graph creation in Neo4j.

Patentdb cleaning

Select * from patentdb where recordid ilike ‘US%’;
-> Since SBIR Award is given by the US government to small businesses in the US itself, so to attain a relationship between these two datasets, we extracted all the US based patents.

SBIR_AWARD_DATA CLEANING

```
--sbir remote server csv cleaning
select "recordID" as recordID,coalesce(company,'NA')as company ,coalesce(award_title,'NA')as
award_title,
      coalesce(agency,'NA')as agency,coalesce(branch,'NA')as branch, coalesce(phase,'NA')as phase,
      coalesce(program,'SBIR')as program,agency_tracking_number,
      coalesce(contract,'NA')as contract,coalesce(proposal_award_date,'NA')as proposal_award_date,
      coalesce(contract_end_date,'NA')as contract_end_date,coalesce(solicitation_number,'NA') as
solicitation_number,
      coalesce(solicitation_year,'NA')as solicitation_year,coalesce(topic_code,'NA')as topic_code,
      award_year,convert_to_int2(award_amount) as award_amount,coalesce(duns,'NA')as
duns,coalesce("HUBZone_owned",'NA') as hubzone_owned,
      coalesce(socially_economically_disadvantaged,'NA')as sed,coalesce(women_owned,'NA')as
women_owned,
      employee_number,coalesce(company_website,'NA')as company_website, concat(address1,address2) as
address,
      coalesce(city,'NA')as city,coalesce(state,'NA')as state,zip,coalesce(abstract,'NA')as abstract,
      coalesce(contact_name,'NA')as contact_name,coalesce(contact_title,'NA')as contact_title,
      coalesce(replace(contact_phone,'() -','NA'),'NA')as contact_phone,coalesce(contact_email,'NA')as
contact_email,coalesce(pi_name,'NA')as pi_name,
      coalesce(pi_title,'NA')as pi_title, coalesce(replace(pi_phone,'() -','NA'))as
pi_phone,coalesce(pi_email,'NA')as pi_email,
      coalesce(ri_name,'NA')as ri_name,coalesce(ri_poc_name,'NA')as
ri_poc_name,coalesce(ri_poc_phone,'NA')as ri_poc_phone
from sbir_award_data;
```

Postgres' Queries Structure

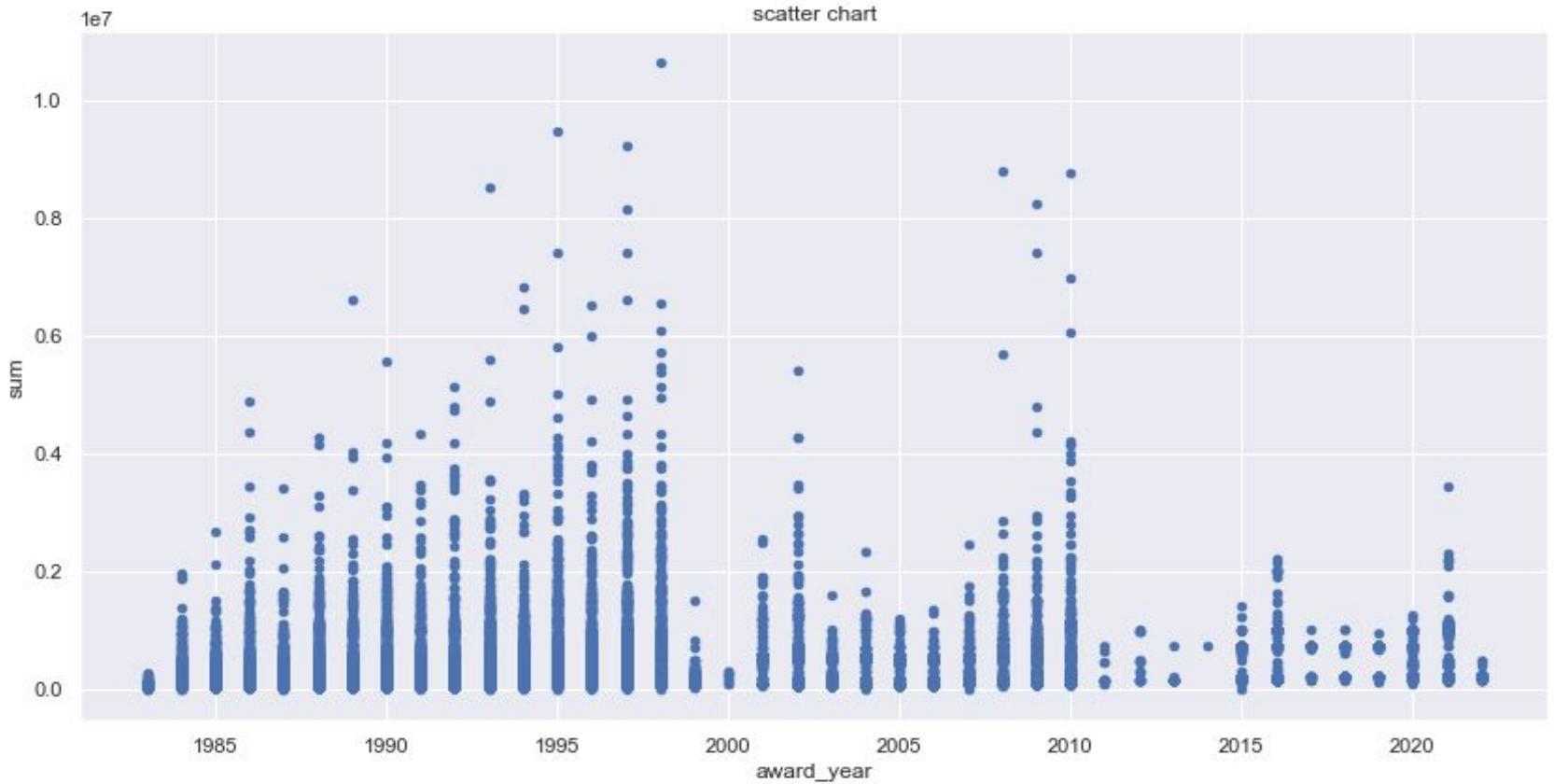
For sbir_award_data , we have utilised window functions to perform statistical analysis and gain certain insights into the data. This has been visually represented on the jupyter notebook. Next we are extracting each company's year-wise total awards

For patentdb, we have utilised common table expressions . We have utilised optimisation technique to increase its runtime on jupyter notebook. On the data frame obtained we have queried to find the number of patents in each technology.

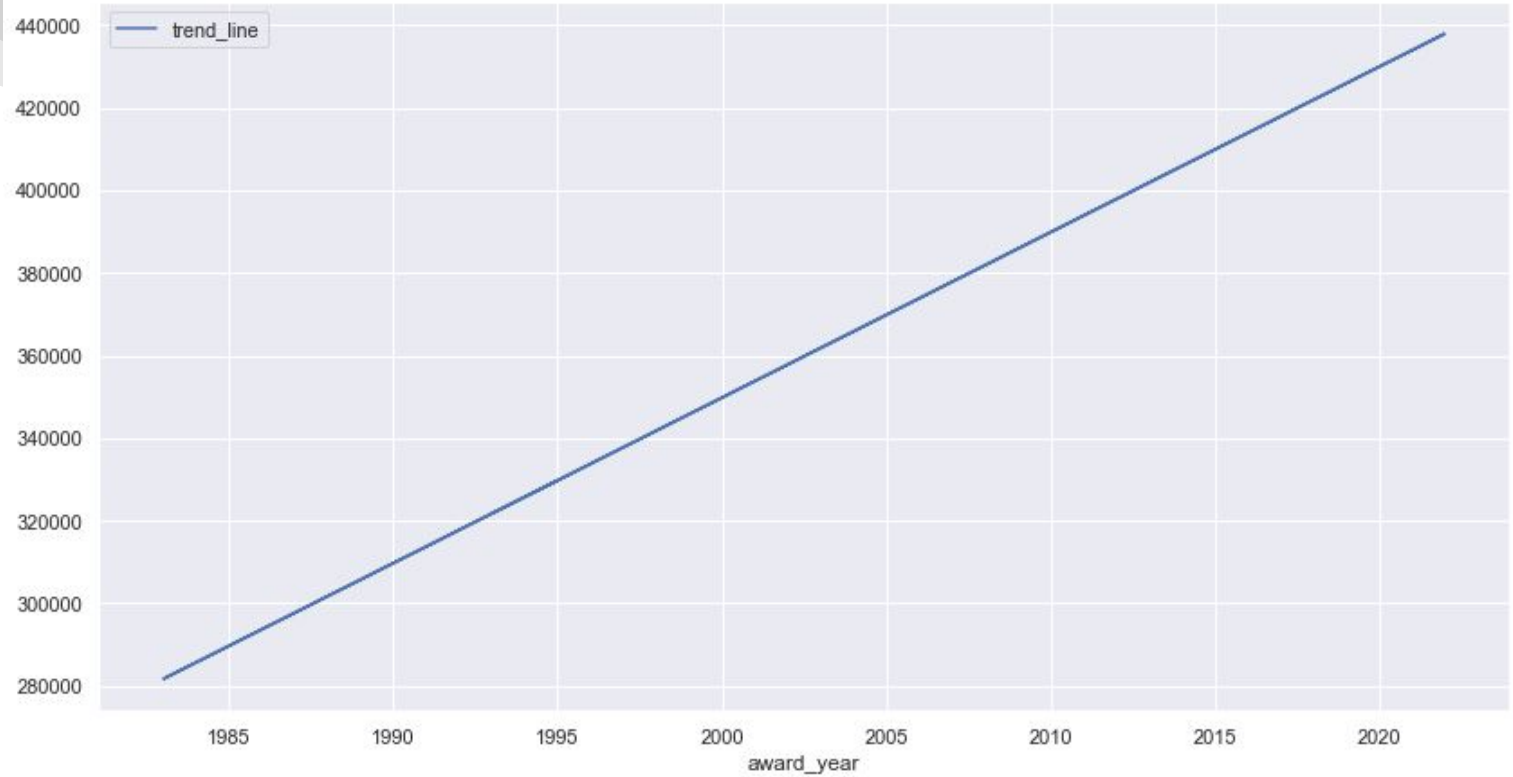
SBIR_AWARD_DATA

```
with a as (
select sum(award amount) as sum, award_year, company
          from "NewResult"
          GROUP BY company, award_year),

   trend line as(
       select slope,
              y_max - x_max * slope as intercept
       from(
           select

coalesce((nullif(sum((award_year-x_bar)*(sum-y_bar)),0)/nullif(sum((award_year-x_bar)*(award_year-x_bar)),0)),0) as
slope,
                max(x bar) as x max,
                max(y_bar) as y_max
            from(
                select award year, avg(award year) over() as x_bar,
                       sum, avg(sum) over() as y_bar
                from a
                )data1
            )data2
   )
select a.*,
      (a.award_year * (select slope from trend_line)+(select intercept from trend_line)) as trend_line
from a;
```

Sbir_award_data: Award Distribution Over Years



SBIR Award Trend Line



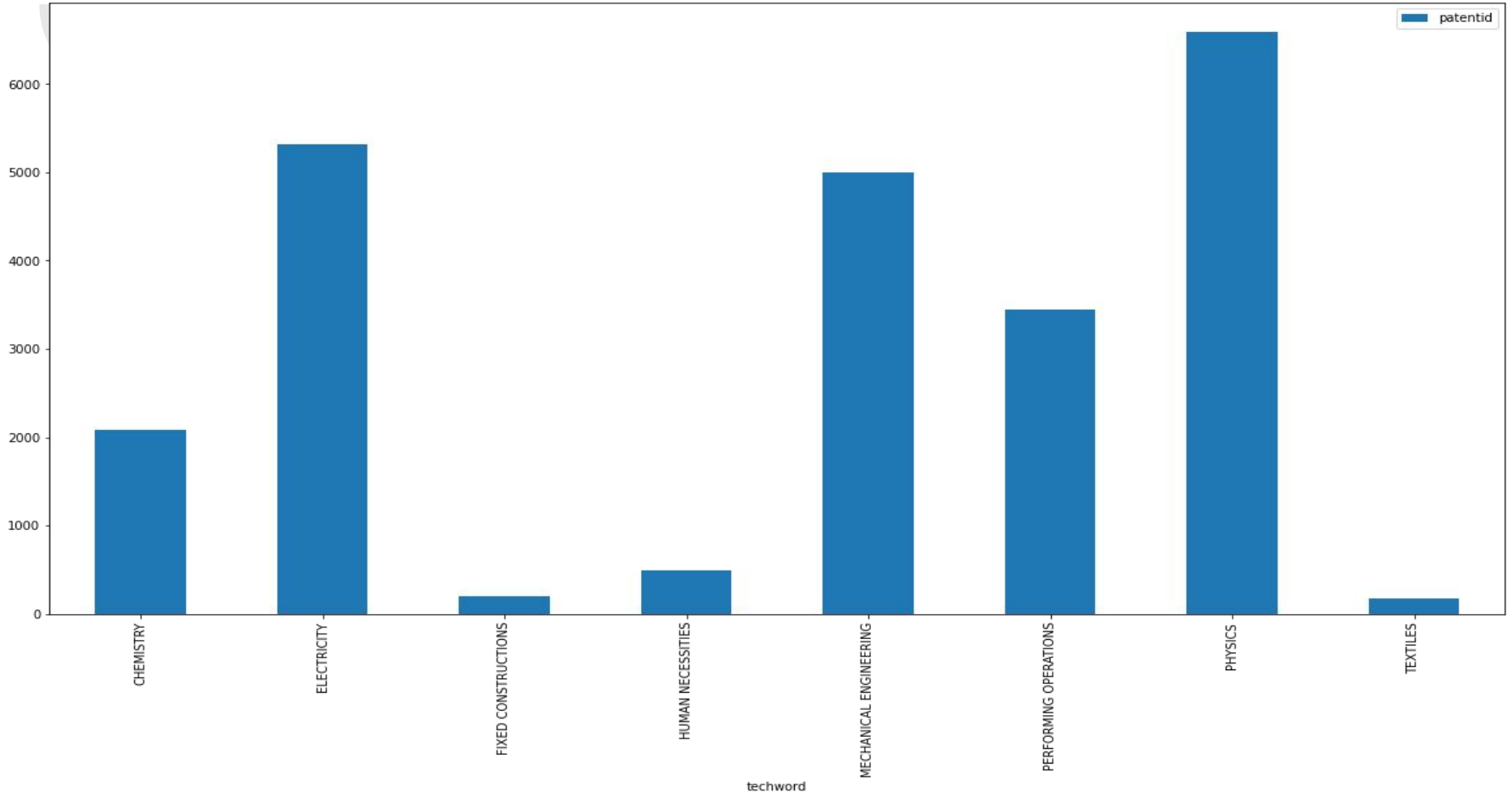
Patentdb

```
with b as
(with a as
(select translate(classname, '[{][}]', '') as classname
from "patentdb (1)"
where classname is not null)
select classname,
(string to array(classname, ';'))[1] as first_word
         from a
   ),
c as (
select patentid, assignee_name_current from "patentdb (1)" where classname is not null
)
select (string to array(first word, ','))[1] as techword,patentid,
translate(assignee_name_current,'{}','') as currentCompany
from b,c;
```

OPTIMIZED:

```
select  (string_to_array((string_to_array(translate(classname, '[{][}]', ''), ';'))[1], ','))[1]
as techword,
patentid, translate(assignee_name_current,'{}','') as currentCompany from "patentdb (1)
```

Patentdb : Finding the number of patents under each technology



Graph

Using the sbir_award_data, graphs using Company, Agency, Branch, Award_Year and Record_Id have been created in Neo4j. The relations are AWARDED (Branch awarding the Company), AWARDED_IN(year in which the company is awarded) and IS_PART_OF(Branch within the Agency)

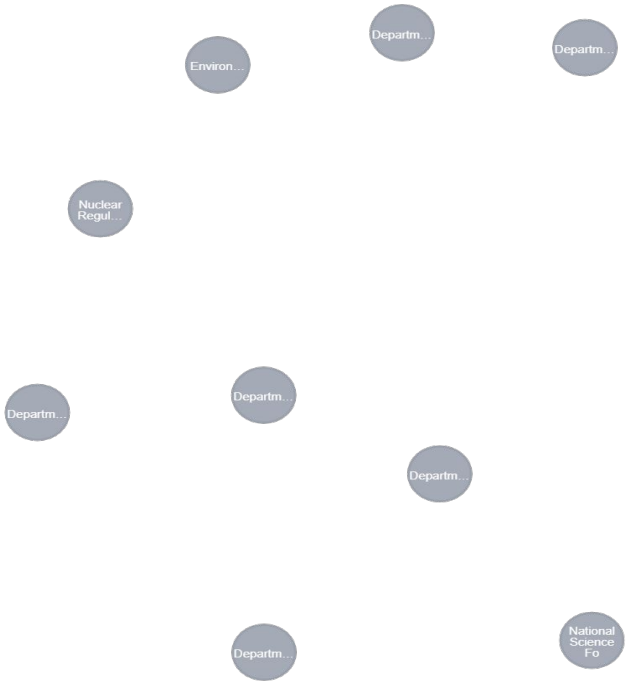
Using the cypher query language,

- We found out the number of awards for each company and the unique branches that are awarding each company.
- Year wise analysis of number of times a branch, agency are awarding companies.

Graph Formation

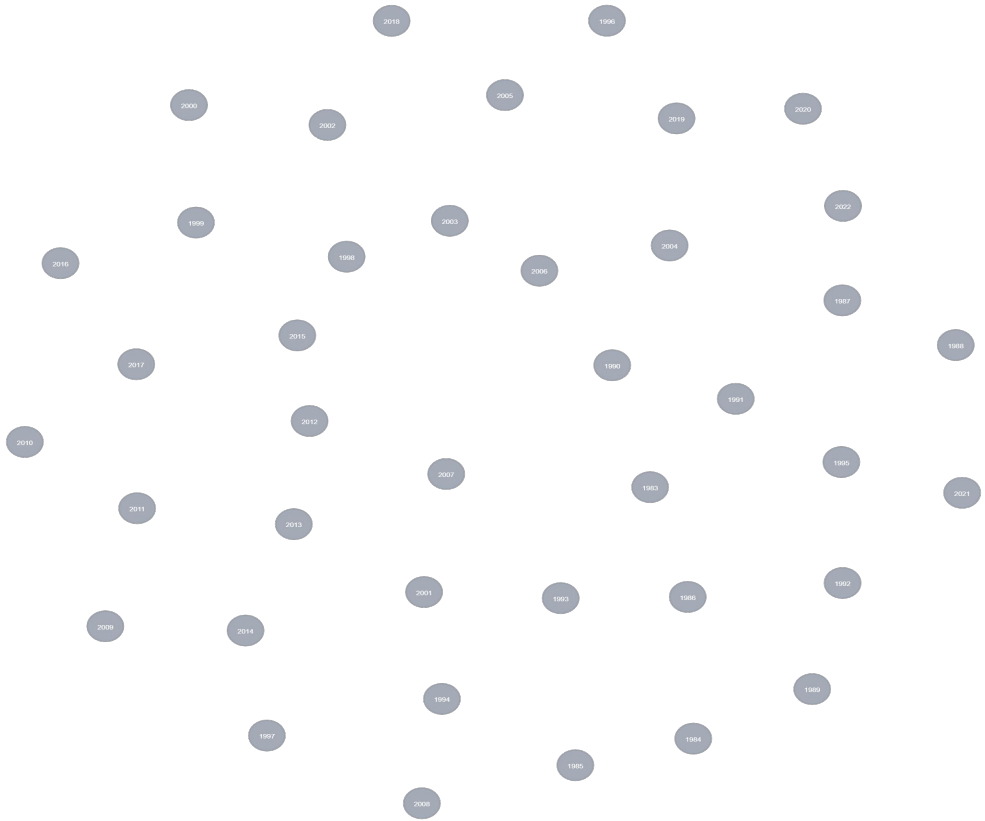
```
LOAD CSV WITH HEADERS FROM "file:///Result_12.csv" as row

with row where row.agency is not null AND row.branch is not null AND
row.company is not null AND row.award_year is not null AND  row.recordid is not
null

MERGE (a:agency{name:row.agency})

MERGE (b:branch{name:row.branch})

MERGE (c:company{name:row.company})

MERGE (y:award_year{name:row.award_year})

MERGE (i:recordid{name:row.recordid})

MERGE (b)-[:IS_PART_OF]-(a)

MERGE (b)-[:AWARDED]-(c)
MERGE (c)-[:AWARDED_IN]-(y)
```

Agencies

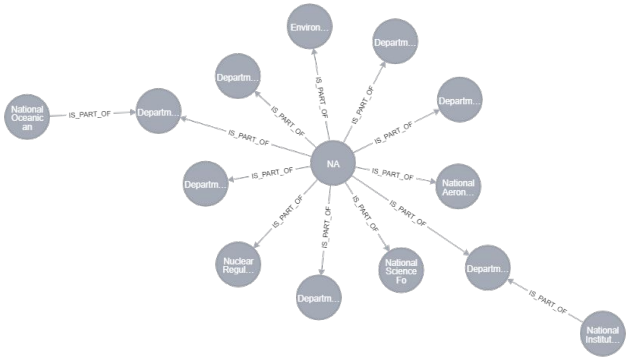


Branches

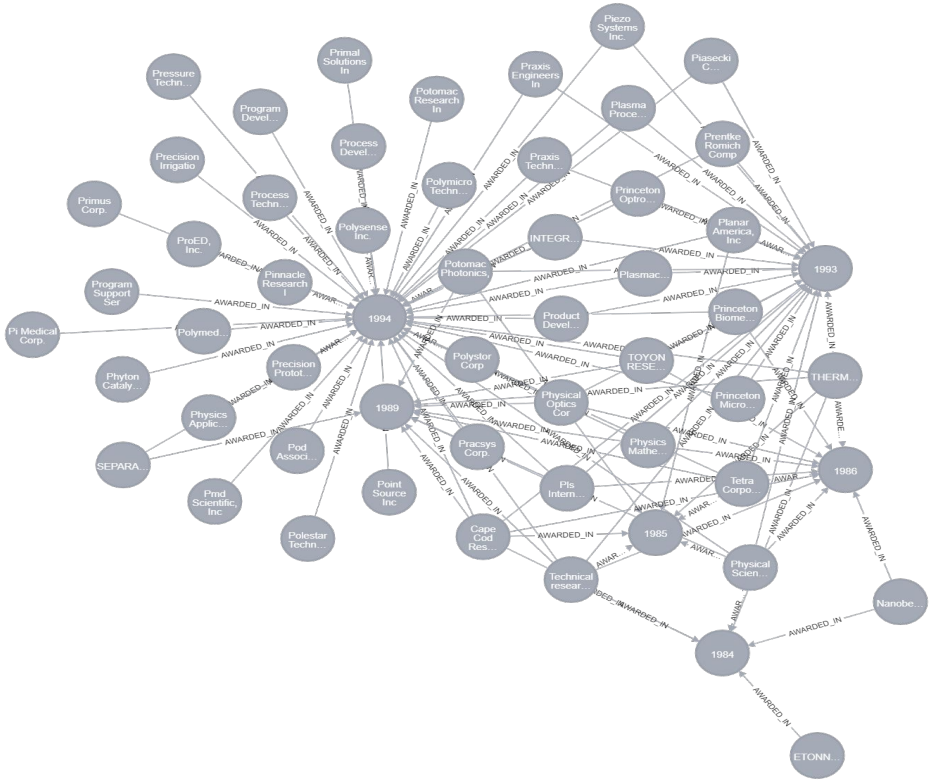
Years



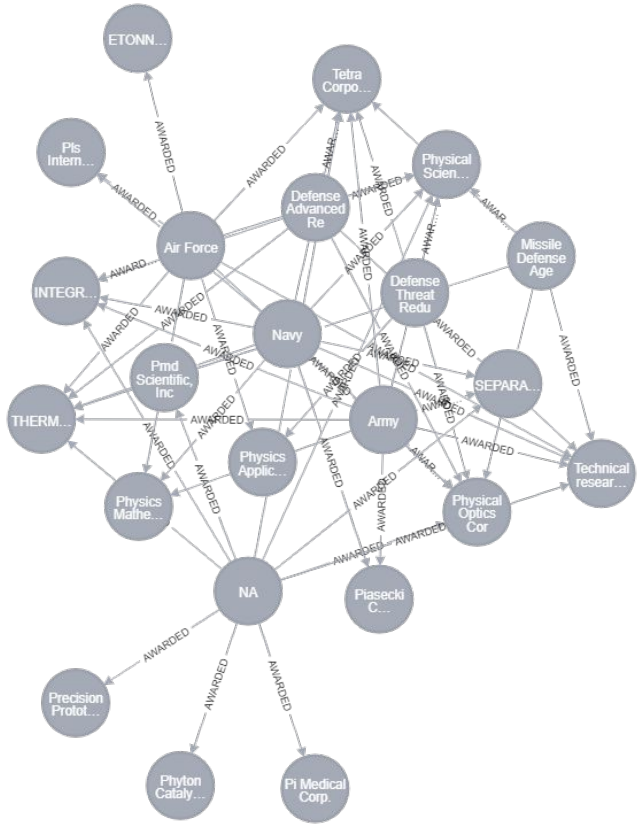
branch-IS_PART_OF_agency



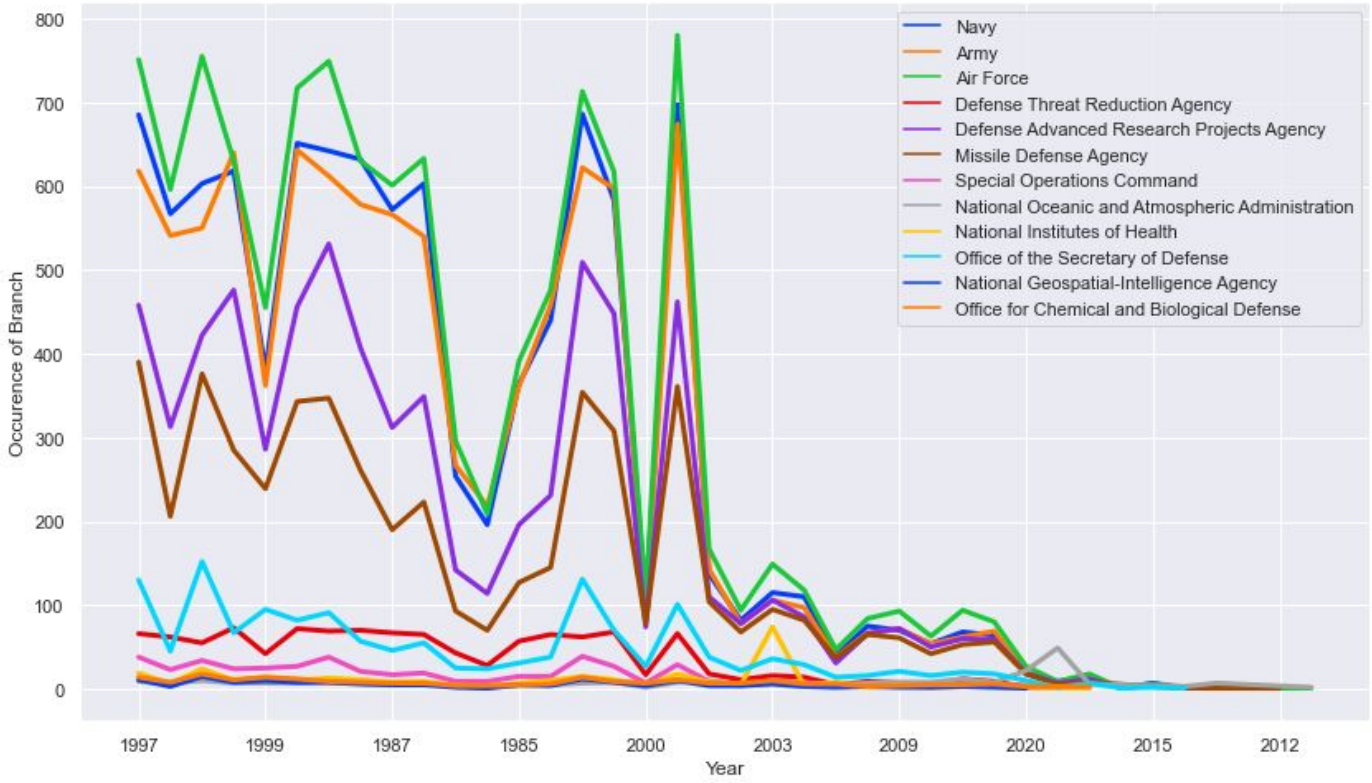
branch-AWARDED_IN-year



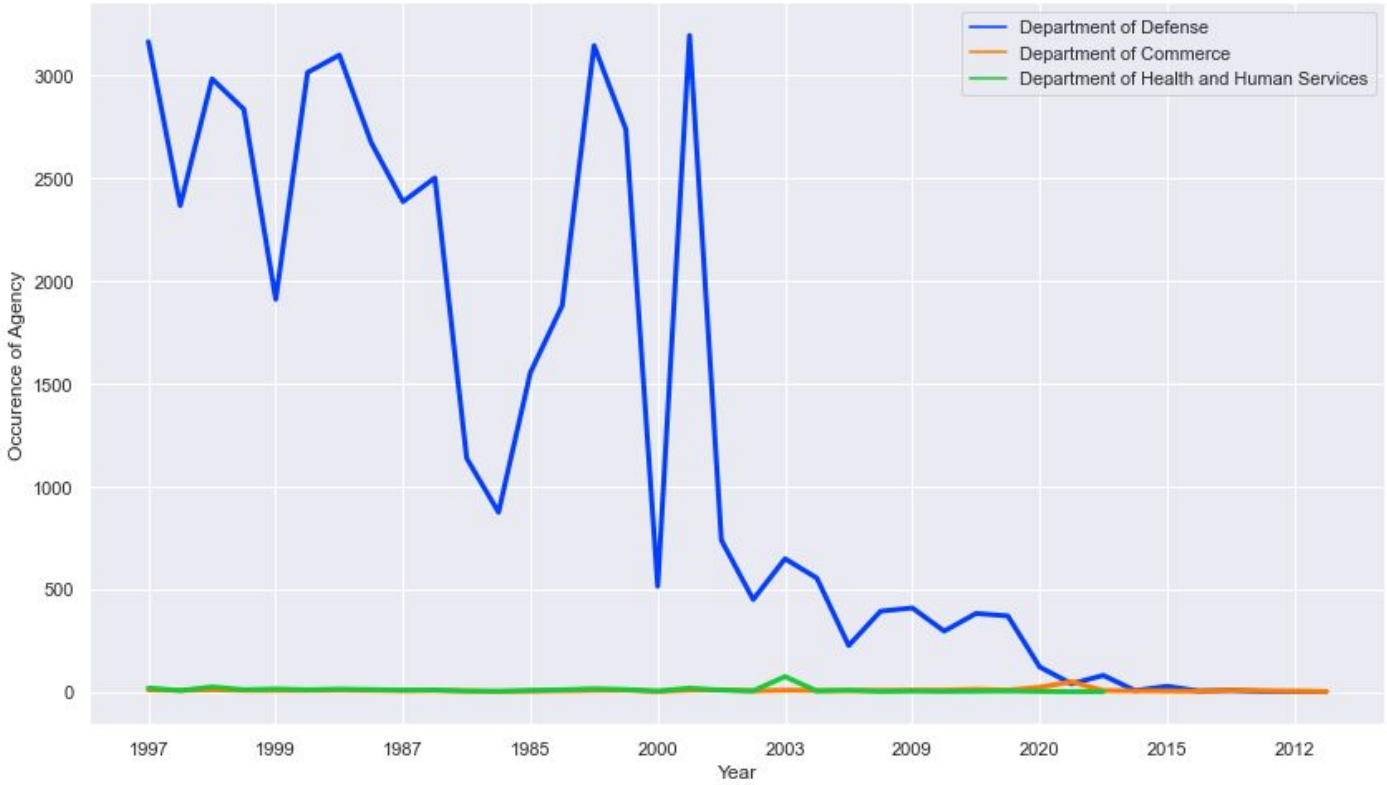
branch-AWARDED-company



Year wise analysis of Branches



Year wise analysis of Branches



Conclusion

- As the years have progressed the SBIR Award Amount has increased. The two companies who have benefitted the most are Physical Optics Corporation and CREARE LLC.
- The most used technologies by the companies over the years were Physics and Electricity, so we can invest in the small companies working on them, which we can see on the SBIR data results.
- We can use Postgres for Data Analysis
- Similarly, we can use Neo4J for graph related data analysis

WebScraping

From the results of the previous queries, we webscraped the data related to SBIR companies that had the most number of awards to find some keywords using XPath and beautifulsoup.