Patents as Technology Innovation and Market Value Indicator

Final Report – Final Project
CMPT733 Big Data Lab 2

Mak Hoi Victor Hau
mhhau@sfu.ca

Yunhye Joo
yunhyej@sfu.ca

Motivation and Background

Understanding the relation of patent collaterals, patent filing and stock price can potentially help analyze the value of businesses and patents.

We aim to create an integrated comprehensive platform that integrates patent data, including full text and applicant's background, patent collateral and assignment data, stock price and business listings. With visualization, we can analyze the potential market value of patents to help companies make collateral decisions.

Businesses can save tremendous time if data are ready and analysis can be generated to aid decision-makers to determine whether to move forward with purchasing patents and at what costs they should afford.

Stock market traders and civilians can decide when to buy and sell stocks with one more factor assisting them to make such decisions.

Our goal is to eliminate the time for patent analysts to analyze the market values of patents by providing a visualization and analysis tool that generate desired results efficiently. We hope to reduce the time taken for the lengthy process to organize patent data and retrieve related patent data and other information. Many research and development (R&D) businesses consider patent collaterals an essential aspect and factor for the company to continue growing healthily and remain the top in terms of market share.

There is more and more related work on researching about the background of the applicants of the patents, such as what country, which institution, or which technology company files the highest number of patents per year. The research on patent collaterals/assignments and prediction of the value of new patents remain non-mainstream in patent research.

Problem Statements

The challenge of creating a platform that can predict market value of new patents lie on the fact that there is not much available data associated with a new patent. Often, new patents before being assigned to companies belong to the individuals who research and invent the methodology, tool, product, or design. Little are known about the background of individual researchers and how advanced and cutting-edge the patent is if only relying on publicly available data online and on the web. One can only look at reverse/backward citations to determine how innovative a new patent is by analyzing the previously filed patents that the new patent cites.

Thus, we want to perform topic modeling to analyze the relation of patents and try to find out the cluster centroid of patents. We want to observe how close a newly filed patent is to the existing patents. However, there is no existing simple and ready-to-use tools, dependencies, packages and functions that we can make use of easily to make a complicated network of patents that the meaningful distance of a new patent with an existing patent can be calculated straightforwardly, and that this distance represents how innovative a new patent is, and more ideally, calculate the estimated market value. Even latent Dirichlet allocation (LDA) can only find out the similarity score of a new patent to a certain topic after performing topic modeling on the patents. We can only find out how close it is to the centroid of such topic by finding out the patent that has the highest score within that topic and compare the similarity score with the query patent. Here we can make use of the full text, abstracts, illustration descriptions, titles or backward citation information for LDA model construction and topic modeling.

As the results, we focus in building a comprehensive tool that can visualize the topics

of LDA modeling given a patent category (IPC) or all categories. We want to visualize graphs combining stock market, assignment data, and patent data. We also want to find similar patents and assignments given a patent document, query words, or a topic. We also want to find words that are closely related in the sense of topics and view the context of topics. We want to create visual graphs that the user can drag and see the relationship of topics, patents, and companies.

What we plan to build from the beginning is that by entering the full text or abstract of the new patent, similar patents can be shown not just by finding words sharing similar meanings but also finding related patents topic-wise. We want to visualize how closely related certain keywords and topics are to each other, in order to analyze the performance and market values of companies and patents.

Overview of Algorithms and Data Science Pipeline

The overview of the data science pipeline from obtaining raw data to making visualization is as follows.

First, we use a script to crawl and download all weekly patent bibliography and assignment data from the official USPTO Bulk Data website. Next, after we extract the zip files, we obtain .dat and .XML files in three different generations of formats according to filing years. We parse the files and retrieve all relevant and important information and store the information into two different tables, one for patents info such as the applicant's background information, the title and abstract, IPC codes, and application date, and one for assignment information including the details of the assignor and assignee and the date of assignment

Next, we then perform NLP techniques on the titles and abstracts of the patent from the patent database, which include filtering away non-alphabets, removing stop words and lemmatizing to get rid of the plurals and link different parts of speech of the same word.

We then construct three different word2vec models using the tokenized titles only, tokenized abstracts only, or both tokenized titles and tokenized abstracts, which we call as text. With these word2vec models, we then construct corpus dictionaries and bags of words, one set for titles only, one for abstracts, and one for text. Then, we use 15 CPUs in our Google Cloud Platform instance to build Latent Dirichlet Allocation models using the multi-processor LDA construction function from Gensim. We have made both 40 and 100 topics for the LDA models and ended up having six LDA models. At last we also create an index for each LDA model for easy and fast reference to be used later in querying.

Thus, with a dictionary, a bag of words, and an LDA model that each is built from the same data set, we can construct an interactive LDA visualization using saliency and relevance metric formulas. The area of each topic is mapped onto a 2D graph and is represented by a circle of variable size. The visualization tool enables the user to point to the topic circles and view the frequency of the occurrence of the keywords inside that topic. The user can also change the relevance metric parameter to adjust the results of the ranking of words in each topic.

We also separate all the patent data into nine categories according to their International Patent Classification (IPC) categories. The categories include A: Human Necessities, B: Performing Operations and Transporting, C: Chemistry and Metallurgy, D: Textiles and Paper, E: Fixed Constructions, F: Mechanical Engineering such as Lighting, Heating, Blasting Engines or Pumps, and Weapons, G Physics, H Electricity, and Y: all other new technologies that cannot be categorized into any group aforementioned. We construct nine sets of word2vec models, corpus dictionaries, bags of words, and LDA models, using data from each of the categories and the procedure we just briefly introduced. We perform the same procedure as above to the nine categories of data.

Our system also supports keyword and sentence querying to find similar patents not just with similar words but also words and content that are closely related to in the sense of topics, patent subjects, technologies and semantics. In the following explanation, we assume we want to find similar patents using 40 topics and in all categories. The query is first properly stripped

and turned into lowercase, with stop words removed and lemmatized. Next, the dictionary of text is loaded such that a vector of the frequency of words in the query that map to the dictionary is generated. Then, the LDA model of text with 40 topics is loaded and a vector of topics with numbers from zero to one indicating how closely related the query is to each of the 40 topics is generated. We can then use the index we generated at the last step of model construction to calculate a vector of similarity score for each of the patents to the query. As we already have the bag of words in the model, we do not need to query the database to perform all the steps mentioned. Each similarity score ranges from zero to one and is independent to each other, which means that the similarity score of all the patents do not add up to one. As the index of this vector is starting from zero, we need to look at the row number of the patent database when we query say the top 100 results to show the full details on the web frontend to the user. We get all the information such as patent number, application and execution date, title, abstract, IPC number, applicant's name and address, citation and assignment history from the database by looking up the row numbers and render on the web frontend. The whole process from sending the query to showing the results only takes three to five seconds. We also query the assignment database by using the patent number of the most relevant patents to append background of the assignee and assignor and execution date next to the corresponding patents. More than one assignment data of the same patent can be retrieved and correctly shown.

We also make use of real-time and historic stock price API by Alpha Avantage and the list of symbols and industries of companies listed on NYSE, NASDAQ and other listings, altogether with our assignment database table to plot graphs that juxtapose the stock price curve and the patent histogram and other information. We can make graphs that are by year, by industry sector, by company, by company per sector, or many other combinations for analysis of the market trend and to find an explanation of the relation of patent assignments and collaterals and the business value.

We also constructed knowledge graphs for each of the nine categories. We first create dense document topics, then calculate marginal and joint probabilities and find associated topics. Then interactive knowledge graphs, which allows users to drag a topic to see the connection clearly and highlight other linked topics, is then generated.

The users can also check words not just with similar meanings but also topic-wise by inputting positive and negative keywords, which mean those that the results should be close to and far away from, into the word2vec models.

Parsing

As the data zip files are packaged by weeks, we programmed an automatic crawling tool that downloads all patents and assignment data efficiently. We programmed an automatic Python script to download and store all the patent and assignment data zip files in the Google Cloud compute engine instance from the official USPTO Bulk Data online portal. We created a list of the names of all zip files with the help of XPath on the official webpage to find the name of all the links. After that, we crawled the data by first analysing the format of different generations of .dat/.xml files that we have downloaded and unzipped.

The XML formats of the patent data are divided by three generations as the USPTO office has revamped the format twice since 1976. In the automatic parsing code we made, the program reads the file name of the zip files to determine which generation of format the files are structured as. We use bibliographic data, which is patent data without full text, to make parsing process faster and the storage space in the Google Cloud Compute Engine instance smaller. The total number of patents since January 1976 to March 2019 we parsed is exactly 6,710,862. The size of the full text data is over 70 GB.

The assignment data consists of size over 23 GB. The patent database and assignment database resulted by parsing are in the size of 6.57 GB and 1.07 GB respectively.

Generation 1 files are stored in .dat formats instead of .xml formats. These files use short abbreviations such as PATN, ASSG to list out

the details. In such cases, the program uses a buffer to parse the XML data line by line. When the tag is read, it matches which category it refers to. We save the sector tags such as “CLAS” (class), “ASSG” (assignment), “PAT (patent), “INVT” (invention), “ABST” (abstract). Any information read will be determined by the tag in front of the information and also the section such tag is within. For instance, the program reads the tag “TTL” which means title, then if the tag is within a tag of “PATN”, then the program can confirm this is the title of the patent. The title is then stored in the buffer. The parser is eventually passed to another function to load the buffer into three chunks and commit these chunks to the database.

Generation 2 files are XML files of that start with “pgb”. This generation of files are those published in January 2001 to December 2004. The program identifies the location of the main patent data XML structures by checking if the line starts with “<?xml”. If the lines start with “<!DOCTYPE”, “<!ENTITY”, or “]>”, it does not contain any useful information and the program needs to skip such lines. Otherwise, the program reads the line and finds all useful information that will be stored in the patent database by locating elements layers by layers. First, it locates the top-level section by finding the tag “SDOBI”. Then, it looks at “B100” to find the information of the published patent. Patent number and publication date are retrieved by finding “B110/DNUM/PDAT” and “B140/DATE/PDAT” under “B100”. Next, it looks for “B200” to find the patent application date by reading the information with the tag “B220/DATE/PDAT”. Other information, including the title and abstract, IPC codes, citations, applicant’s details and claims, are retrieved in a similar manner. A dictionary is used to store all the information so that after parsing an XML file, the information in the dictionary is stored in the database.

Generation 3 files are those published since 2005 with file names beginning with “ipgb”. A similar strategy is involved in parsing the XML structures and finding out related information by looking at the enclosure of tags and parent tags. We store meaningful data that we later make use of in a buffer, before storing the data from the buffer to the database once reaching certain number of patents in the buffer.

Model Construction on Google Cloud Platform and Data Size

We created multiple Google accounts with limited free student credits and initialized at least 10 different compute engine instances to train the LDA and other models simultaneously. Each instance consists of 4 to 16 CPUs and we have used up to 15 CPUs (n-1 CPUs, reserve one for other processes) as workers to train LDA models in parallel in each compute engine. Even though we make use of 15 CPUs with tens of GBs of RAMs per instance and have chosen the optimal instance spec that saves costs yet provides the largest space. We use buckets as storage space for model files, each in several GBs, to transfer files among instances. We upload files using “gsutil cp {zip file name}.zip gs://{bucket name}” after authentication is processed using “gcloud auth login” in the instance SSH terminal. As it takes several days to finish training all the models on one set of patent data, we run the python processes at the background by using “setsid” and “screen”. (Other ways include “& disown”.) We also use the GCP instances to crawl and parse data. It takes more than a week to finish the crawling and parsing process. We have collected 6,710,862 patents and their detailed information in our database, resulting in a SQLite database of patent data with size around 6.57 GB and SQLite database of assignments data with size around 1.07 GB. The original raw patent data files including full text from USPTO Bulk Data portal are over 70 GBs in total size.

At the initial stage, we tried to construct LDA and word2vec models, and perform NLP procedures on the full text of all the patents. However, we let the process run over two nights and the program could not finish creating the word2vec models, which was the first step after parsing. Thus, we switch to make use of only titles and abstracts to construct models. We construct models based on only titles, only abstracts, and both titles and abstracts, which we call as text. In fact, we began parsing the full text of all patent data since 1976 but decided to stop it after running the process in GCP for over two days.

Every time we started a new Google Cloud compute engine instance, we need to set up the virtual environment either using pip or

anaconda. We have tried either ways on different instances. After setting up a virtual environment, we can safely download and install all relevant packages and dependencies for our crawling, parsing, model construction, real-time running, and web frontend to work. Examples of packages needed are Gensim and Flask. After setting up all dependencies and virtual environment, we also have to set up juptyer notebook to run on the remote cloud instance by following lengthy processes.

We have spent several weeks on crawling the data. Without GCP, our project cannot be finished.

Choices of Topic Modeling Methodology

We have chosen to use Gensim and its Word2vec and LDA model construction functions to build the skeleton of this project, which mainly focuses on creating topic modeling by using LDA. We prefer LDA over LSA and pLSA because all these algorithms have major drawbacks that make LDA as our final choice.

Latent Semantic Analysis (LSA) is not that efficient in representation and it lacks interpretable embeddings, which means that we have no ideas what the actual topics and their keywords for the topics (group of documents) the algorithm separates the documents into are, and the components may be arbitrarily negative or positive.

pLSA, which stands for Probabilistic Latent Semantic Analysis, only adds a probabilistic treatment of words and topics on top of LSA, resulting in major drawbacks, including overfitting due to the linearity of parameters and number of documents, and being unable to assign probabilities to new documents.

Gensim

We make use of Gensim and choose word2vec over any other tools and packages as the results generated by Gensim is accurate as Gensim follows strictly the corresponding methodologies by saving our time to code the NLP techniques and modeling provided by Gensim. Gensim provides word2vec modeling, LDA modeling, bags of words and dictionaries generation, and NLP procedures such as lemmatization and removal of stop words, which are all necessary to create a sounding topic modeling and related visualization.

Evaluation

Using LDA to perform topic modeling is one of the common approaches in the data science industry and research within the recent several years. It makes use of Dirichlet priors to return acceptable generalization for word-topic and document-topic similarity distributions. Briefly explaining, LDA creates the best distribution over distribution with the highest probability. In other words, it chooses the most likely distribution over several probabilities of certain distribution. Which set of distribution of similarity of topics is most likely? LDA is the best among LDA, LSA and pLSA to perform topic modeling. LDA has an advantage of generalizing new documents with ease. The document probability of pLSA consists of a fixed point that requires the document to be discovered in the training database in order for the probability of similarity to be given as a result. On the other hand, LDA can always generalize a new document by referencing and sampling from the Dirichlet distribution. Given that LDA is effective and popular and is a built-in function in Gensim, undoubtedly, we choose LDA over the other two methods aforementioned.

We properly preform NLP procedures to remove the stop words and lemmatize words, which means removing plural and articles, and to eliminate spaces, numbers, and non-alphabets after lowercasing all words in the text of patents in the training phase and of the query in the running (testing) phase.

We follow the usual procedures of data cleaning, ETL, and building LDA models after building bags of words and dictionaries, which are made after building word2vec models.

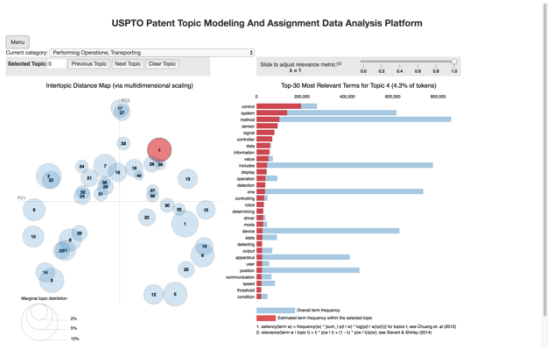
We built the web frontend properly following common practice to prevent errors when users enter irregular queries and W3C measures for usability, such as providing descriptions for missing illustrations.

Data Product

We have created a Flask web frontend that let anyone to analyze patent data, assignment data and stock price interactively. There are several features: finding similar patents and assignments based on a query (that can be from a word to a whole document), viewing the distribution and grouping of topics of patents in each of the seven ICP categories on a 2D plot, finding similar words topic-wise, viewing interactive visual graphs that show the connection of topics, and viewing the context of topics. One can connect locally or through hosting it using cloud services such as Google Cloud Platform or Amazon Web Services. After downloading all prerequisite software in the requirement.txt, one is ready to use our web frontend. Once launched with python in terminal by typing "python main.py" in the subdirectory that contains the Flask frontend, one can go to a web browser and make use of this interactive tool without needing to wait for days and nights for the models to be trained as everything is prepared and ready for the user to use right away. If hosted locally using a terminal or shell, one can go to localhost:5000 on the web address bar as the default Flask port is 5000. On the frontend we created, one can always navigate among different functions using the sidebar at the left by toggling it.

The user can always choose different categories, and this applies to all functions. For instance, the user can choose patent ICP category Physics, the webpage is then reloaded immediately no matter which page the user is on. Then either an LDA visualization, results of patent or word similarity, or a visual graph under that topic is shown.

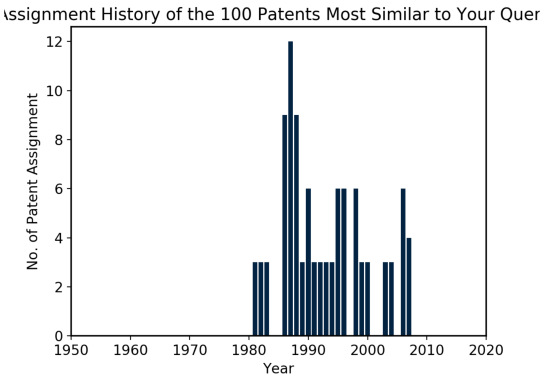
LDA visualization page allows the user to interact with the circles to see the top words appearing in that topic. The user can freely browse and switch in between topics as the topics are all plotted on a 2D graph with two axes. The user can change the relevance metric parameter to rearrange the frequency of occurrence of words in a certain topic. The user can switch categories to visualize topics that are under a certain category or all categories.



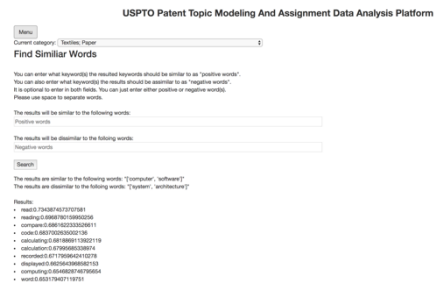
The user can enter a word, several words, or paragraphs of words to query similar patents. The graph showing the patent assignment activities of the top 100 related patents are plotted according to years. A detailed table of 100 mostly related patents in terms of topics and semantics are shown. As described in the Data Science Pipeline, all the related information including patent number, filing date, execution date, applicant's name, applicant's address and country, title, abstract, full text, forward and reverse citations, IPC categories and generation, assignment data (if any) including details of the assignor and assignee and the execution date, and even the .dat/.XML structure generation are shown. These data are obtained from querying the patent and assign database altogether after obtaining the top 100 most relevant patents by loading models efficiently.


Patents similar to these keywords/sentences include:

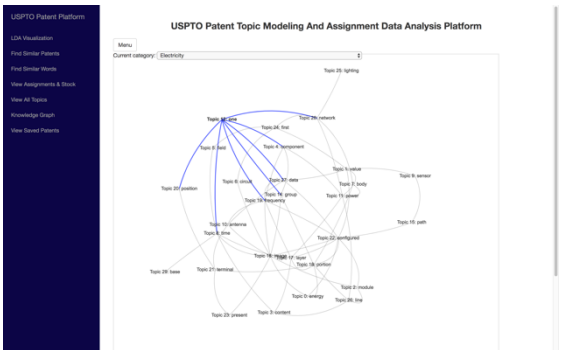
Patent Number	Patent Title	Abstract	Applicant	Application Date	Application Number	Applicant's City	Applicant's State	Applicant's Country	Publication Date
04187978	Mailbox protective apparatus	Apparatus for protecting cantilevered mailboxes from damage caused by collision with passing vehicle...	None	19780519	05907755	None	None	None	19800212
04218338	Process for recycling gaseous reactants used for _regenerating a hydrocarbon hydroconversion catalyst	A hydrocarbon hydroconversion catalyst is regenerated by (a) combustion, (b) chlorination or oxychlo...	Institut Francais du Petrole	19780620	05917224	Rueil-Malmaison	None	FR	19800819
04506749	Underhood muffler for tractor	An underhood muffler mounted in the engine compartment of a motor vehicle supported on the turbo-cha...	Allis-Chalmers Corporation	19811230	06335595	Milwaukee	WI	US	19850326
04109297	Conduction system for thin film and hybrid integrated circuits	A metallization scheme for interconnection of elements in thin film and hybrid circuits is described...	Bell Telephone Laboratories, Incorporated	19770325	05781294	Murray Hill	NJ	US	19780822
04347969	Tray	A tray which may be set up in the field and locked into position without the use of glue or staples....	Weyerhaeuser Company	19810508	06261831	Tacoma	WA	US	19820907
04450713	Method for measuring hardness of rubber and plastics and a hardness tester for use therein	The invention relates to a method for measuring the hardness of rubber, plastics, etc. and a hardnes...	Sumitomo Rubber Industries, Ltd.	19820616	06389156	Kobe	None	JP	19840529
04549245	Capacitance-type material level probe and method of	A capacitance probe and method of probe manufacture	Berwind Corporation	19840921	06652853	Philadelphia	PA	US	19851022



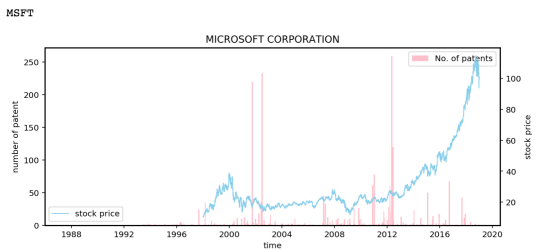
The user can enter positive or negative words, and the number of words entered can be zero to infinite, as our word2vec models can handle any combination of positive and negative words given that at least one word in total is given. Our frontend smartly filters out words that are not in the pre-trained dictionary after all the words are lowercased and spaces and non-alphabets are removed, such that no errors are prompted.



The user can drag and drop the topics on the visual graphs to see the lines linking other topics very clearly and interactively.



The user can also view other visualization such as juxtaposing stock price and assignment data, or visualizing patent filings of companies in a certain industry.



Actual result of our frontend can be viewed at the video https://youtu.be/wacQ3G0qRjw.

Lessons Learnt

We have learnt how to build an interactive Flask frontend that uses Python to run and load the pre-trained LDA models, word2vec models, and relevant models to find similar patents and retrieve related assignment data in real time, using as few as milliseconds to load the results after going through the process aforementioned in the Data Science Pipeline section above.

We have learnt how to script a Python program to automatically crawl, download and save all the patent and assignment data zip files in the Google Cloud instances.

It is our first time to use Google Cloud Platform (GCP) to run instances and perform time-taking processing such as LDA model construction, NLP procedures and parsing of data. We also make use of bucket storages and understand different configurations of specs of instances.

We have successfully combined data from several sources to create meaningful visualization for data analysis. Our sources include official USPTO Bulk Patent and Assignment Data, Alpha Avantage Stock Market API, and NYSE, NASDAQ and international listings available for the public. We are able to perform ETL work and data cleaning to select necessary information to be stored in databases and recognize slightly different names of the same company in the registry so we can accurately show the patent filing and assignment data of a particular company.

We have gone through the whole process from obtaining raw data to creating topic modeling visualization. Crawling, parsing, NLP techniques, construction of word2vec models, dictionaries, bags of words, LDA models and indices are all time-consuming steps that each take days to complete on more than hundreds of gigabytes of patent data, as we perform the procedures above for different sets of data. This is what big data is. This is something we cannot run on our own MacbookPro even though ours are with the top specs of the product line. Multi-processors (or multiple GPUs depending if the processes are CUDA-compatible) and the cloud infrastructure that provides high-end top specs containing tens of CPUs and GPUs are a must to run all the procedures. Without cloud

services, this project is unable to be completed individually without using lab environment provided by university. We did not choose to use the cloud cluster in SFU because there is not enough storage space to store all the patent data and the Spark and Hadoop infrastructure is in high demand that the resources are on a competitive basis. We often need to wait for hours for the CPUs and GPUs to be freed for us. We believe our choice of utilizing GCP is wise.

We have also successfully built a Flask frontend that integrates all the features we planned to provide and our pre-trained models so that a querying for similar patents can show up instantly with graphs rendered and tables loaded.

In the future, we would want to deep dive into biosimilar markets to focus on analyzing how an expired medication patent is being utilized in the pharmaceutical industry. Interesting results could be dug out by comparing expiry of drug and scientific methodology patents with the business value and the market values of the drugs and other products the company newly creates.

We would also want to make use of more advanced techniques or other more complicated and cutting-edge methods to perform topic modeling of patents and find out relations of business giants and small firms, and being able to predict market value of a newly filed patent given there is no backward/reverse citation and collaterals data with it.

Given more sufficient time, we could train LDA models for each company, or categorize patents by companies instead of topics, so that we can create visual graphs of patents of a certain company, or visual graphs of relation among different companies.

Summary

We have been trying to identify the relationship of the patent assignment and stock market price. Clearly, the correlation of the stock market price and patent assignment is not significant enough to have any conclusion being made and grounded. There is no clear relation of stock market price and the purchases of patents. We have found another interesting fact that before a new stock is listed, there is always a huge amount of patent assignment activities involved right before the company begins to be listed publicly. This coincides with the common sense that the company purchases or is granted huge number of patents to begin their new businesses with. A new company may be established based on patents that are originally owned by other big firms or scientists and research institutions. Sometimes, an entrepreneur with a unique vision would collect all the related patents and begin building an innovative commercial product or providing a new service to customers and clients.

Otherwise, patent filing and assignment does not contribute to the stock price or the market value of a company. Usually, a research and development company, especially in the technology sector, such as IBM, Samsung, and Alphabet (Google), keeps inventing and filing new patents every year. The number of patents and the stock price cannot form a formula that apply to all years and all companies.

Yet overall, we have created an integrate platform that can help integrate patent and assignment data and provide the functionality of searching relevant patents and assignments topic-wise and semantically similar, and visualizing graphs of patent and assignment data by year, by company, by sector, or other combinations of search criteria. We have achieved the goal of making an efficient tool to save tremendous time to organize existing patent data, acting as an assistant tool for patent researchers for patent analysis.

We have also made use of it to try to find the relation of market value of a company and assignment being assigned to the company. In the future we can try to find interesting findings of other relation and criteria by integrating more data and creating visualization in a different way.

Existing patents that are filed cannot reflect and predict the failure or success of the future of a company. Often time, when a patent is filed, it is already scientifically well proven, and experiment is well performed. Many research analyses that fail to have its hypothesis be proven and inventions that fail to work are never filed. The patent database neither includes the work wasted behind the curtain,

nor reflects the other factors that determine the success and importance of a patent, which include costs, difficulty to reproduce, marketing and promotion, and the acceptability to the target audience. We hope that we can find out more relation of patents values, and eventually be able to build an automatic tool that predicts the market value and the success of the product, or part of the product, a patent that is ready to be filed makes of.