

Final Project – BSc in Software Engineering  
**Patent visualization tool for venture capital**

**Author**

Sverrir Heiðar Davíðsson

**Instructor**

Steinn Guðmundsson

## Abstract

This study explores whether publicly available patent and patent application data could be used for early detection of promising companies with respect to venture capital financing. The intended application of this data relies on the ability to search for patents based on their relation to fields of technology or possible scope of usage independent of patent classification and appearance of a particular word in the patent. Using a bag-of-words (BoW) representation of the patent abstract and t-SNE for visualization, we showed that clusters of patents could be identified. The method was tested in two case studies that attempt to show that similar patents will overlap and dissimilar patents will form separate clusters. Lastly we want to create an interactive tool that shows how these methods can be applied to create value in a venture capital setting by looking at the prominent holders and distribution of patents in a given subfield of a technological domain.

## 1 Introduction

Venture capital is a type of financing that supports promising companies early in their lifetime in hopes of high return on investment if the company manages to grow substantially. Any tool that could help investors find companies at an early stage could be of great value. With machine learning and data mining it might be possible to use the existing data on patents or articles to see when a company is in the early stages of investing in or researching a particular technology of interest. In this study we will only be using available patent data due to its availability and structure. The reason patents are of interest to investors is because they indicate that their owner has at the very least been showing interest in the patented technology and at most can signal an intention to develop or create a product for production. This is of interest to venture capitalists specifically in the context of an emerging or promising technology.

A patent is an intellectual property that grants exclusive right for a product or process for a limited time. (WIPO, n.d.) A patent contains a precise description of the invention and is comprised of text, pictures, tables, references to other patents along with bibliographic data. Each patent contains a section of text called an 'abstract'. The abstract of a patent is a short summary that should communicate the essence of the invention. (Government of Canada, 2016) This is in contrast with the body of text contained in the patent with more detailed specifications and description of the invention. For this study we will not be looking at the entire text of the patent but instead only the title and abstract, since they should be able to convey the most relevant information of the patent.

One of the drawbacks of conventional patent search engines is their reliance on specific search words and existing patent categories. A patent can be indirectly beneficial to the manufacturing or use of another technology and can be categorized differently than the technology it hopes to support. For example a prosthetics company could own a patent in a textiles category that they intend to use for manufacturing of prosthetics components. Therefore, potential companies for investment might slip under the radar of a hopeful investor that is searching only for words specific to that sector. The investor can also not trust that any given category is representative of a specific field of interest.

The purpose of this study was to show that it is possible to use publicly available patent data to search for patents based on their field of technology. We attempted to show this with the following steps, in order. First, we found an appropriate and publicly available dataset for patents. Next, we extracted useful features from the abstracts of these patents. Then we applied the t-SNE algorithm to visualize similarity of patents and tried to show that it was possible to find clusters of patents with two case studies.

## 2 Data

In this study we used the bulk-data on granted patents provided by PatentsView, due to its accessibility and manageable size. Several other open-access data points or bulk data for patents are made available by the United States Patent and Trademark Office (USPTO). Most patent data is not of relevance in this paper, like drawings, references and the full text of the patent. We are mostly interested in the abstracts and bibliographical information, since the abstracts will be used for searching and the bibliographical data is used to refine and aggregate the results of the search. We might be more interested in recent patents or patent application data in the context of venture capital and we will be interested in knowing what company each patent belongs to, among other things.

PatentsView is a service supported by the USPTO intended to increase the value, utility and transparency of the US patent data. (PatentsView, n.d.) This service provides both APIs and bulk data for patents and patent applications. Patent applications are different from patents in that they have not been accepted and only allow the holder to claim that their tech is 'patent pending', while a granted patent would allow its owner exclusive rights to that intellectual property. At present, support for granted patents is greater than that of patent applications, therefore we will mainly consider this subset of the data in this paper although, with respect to venture capital, the patent application data is more interesting since there is often a long time from the application of a patent and the granting of the patent where a company can continue to grow in their respective field and most of the positive indicators of interest to venture capital hold true for patent applications, regardless of whether or not the patent is granted.

The "patent" data set contains data only on granted patents. (PatentsView, 2019) This dataset has around 7.1 million patents and is roughly 5GB in size. This is a major limitation in this study, since some of the methods used would require the use of a 7.1 million by 7.1 million matrix, if all the data was used, which with only 32 bit values for each cell in the matrix would require around 201 terabytes of RAM and then there is also the computation required to both fill the matrix and then to use the values. For this study we used a subset of this dataset which contains only patents from after the date of 1. Jan 2018. This subset contains roughly 535 thousand patents. This would still require around 1 terabyte of RAM to use, so we split the data into even smaller subsets when we want to use it.

## 3 Feature extraction

The primary features used in this paper are a bag-of-words (BoW) vector representation of the vocabulary of each patent. The BoW representation is a method of turning a collection of strings into numeric vectors. These vectors are  $M$  dimensional where  $M$  is the number of unique words used in the whole collection. Often with BoW, the values of the vectors are usually either a count of how many times each word appears in each string or made with some other function to represent the importance of each word in the abstract. Each patent, represented by a vector, is then compared

to every other patent by calculating the *cosine distance* (or *cosine similarity*) between the vectors. This leaves us with a *pairwise similarity matrix* that we can use with the t-SNE algorithm.

### 3.1 Trimming the data

Before we use the data, we can process it a little to get rid of irrelevant, redundant or useless information. Instead of using every unique single word in the abstract of each patent, we want to filter out words that don't help us comparing the patents together. We begin by *stemming* the words, which is the process of reducing words to their *word stem*. For example, we don't want the words 'medicine' and 'medicinal' to have separate values, since they convey an almost identical meaning. With stemming, both words are reduced to their common stem 'medicin'. Then we eliminate so called *stop words* like 'the', 'an', and 'which', since they can appear in any patent and don't give us any clue as to how similar two patents are. We also want to get rid of words that only appear once, since the information won't be relevant to comparison with any other patent. A higher threshold for the minimum number of appearances of a given word can be used since, for example, if only two patents out of a thousand contain a word, the word might be increasing the required memory and runtime of our application disproportionately to its contributed value.

### 3.2 bag-of-words

Once we have processed the data, we want to transform our collection of abstracts to a matrix where each row represents a patent and each column represents a unique word. The most important part of this step is how we calculate the values we place in the matrix. The objective here is that, if two patents describe a similar invention, their respective rows should reflect that by being similar by some measure. One way of comparing two vectors is by calculating their *cosine similarity*. If two vectors are parallel their cosine similarity is 1. If two vectors are perpendicular, their cosine similarity is 0. Since the length of an abstract is not relevant to its meaning, the cosine similarity is a good measure for this use-case. With this method we would transform the strings "x x y", "x x y y x" and "y y x" would return the vectors [2,1], [4,2] and [1,2] respectively. The first two will have the same direction but different magnitudes meaning their cosine similarity will be 1, while the last two will have similar magnitude but a different direction, so their cosine similarity will be less than 1.

$$\text{similarity} = \cos(\theta) = \frac{v \cdot w}{\|v\| \|w\|}$$

Usually, the values in a BoW vector are a count of how many times a particular word appears in a string. There are some fallbacks to this method. We can assume that the more times a word appears in an abstract, the more relevant it is, but some words appear more often than others in general and are less likely to matter to the content of the patent than other words that are mainly used when they are highly relevant to the content of the patent. To incorporate this characteristic, we use *term frequency-inverse document frequency (tf-idf)*. (Leskovec, Rajaraman, & Ullman, 2011) The tf-idf value is larger the more times a word appears in a specific abstract and is offset by how many abstracts contain the word. We end up with an NxM matrix A, where N is the number of patents in the subset and M is the number of unique words used in the representation.

### 3.3 Similarity matrix

The input into this step is the NxM matrix A from the previous step. From this step we want to make a new matrix, B, that is NxN where N is the number of patents in the subset and the values  $B_{ij}$  are the cosine similarity between patent  $i$  and  $j$ . This matrix is called a *pairwise similarity matrix*.

### 3.4 Visualization

Since the main objective of the paper is to show whether the data could be useful for venture capital, we will be using t-SNE, due to its usefulness in visualizing high-dimensional data. A more complete tool could include a different technique for clustering or analysis but the emphasis here is placed on showing that such efforts could be worthwhile. The t-SNE algorithm is a non-linear dimensionality reduction technique, that tries to conserve distance between points, with emphasis on keeping points that are close in the original vector space close in the reduced vector space. So, if we have successfully managed to transform our collection of abstracts into a vector space where locality means similarity in content, we have shown that it is possible to explore and search for patents in this vector space based on similarity.

### 3.5 Scalability

If we look at the total number of unique words in our vocabulary as a function of how many patents we include in our analysis, we get a curve that can be approximated as a scalar multiple of the square of the number of patents in our collection, as seen in Figure 1.

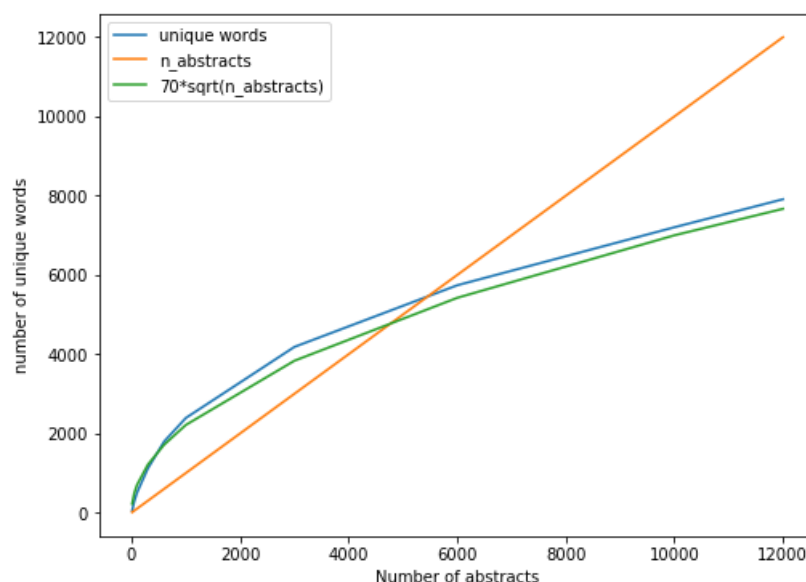


Figure 1: The relationship between the number of unique words in a collection of abstracts and the number of abstracts in the collection. This relationship can be approximated, at least for a number of patents in the range of 1 to 12000

We can see from Figure 1, that after approximately 5000 patents, the number of patents is larger than the number of unique words in our vocabulary. Since the values in both the BoW representation ( $N \times M$  in size) and the similarity matrix ( $N \times M$  in size) are stored as 64-bit float values. After 5000 patents the similarity matrix would be our limiting factor, since the size of an  $N \times M$  matrix is less than the size of a  $M \times M$  matrix. If we want to limit the memory of either matrix to a reasonable 4GB. We get the following:

$$4'000'000'000 * 8 \text{ bits} = 64 \text{ bits} * N^2$$

$$N = 22'360$$

So we have a limit of roughly 22'360 patents in terms of storage. Another probable limiting factor is computation. If our method takes too long to return results, it becomes less useful. To create the similarity matrix, time complexity has to be  $O(N^2)$ , since each patent has to be compared to  $N-1$  other patents. At best we can have roughly  $\frac{N(N-1)}{2}$  comparisons since cosine similarity metric is symmetric, so we don't have to compare each pair twice. The time complexity of the 'exact' t-SNE

is  $O(n^2)$  (Maaten & Hinton, 2008), unless we use the Barnes-Hut approximation method which has  $O(n \log n)$  time complexity. With experimentation we get roughly 10 seconds of computation on a decent laptop for  $N=1500$ , with the Barnes-Hut method. We will therefore not be using subset much larger than  $N=2'000$  in the application but can use larger numbers, up to  $N=22'000$  for case studies and examples.

## 4. Results

To see if this approach produces meaningful results, we looked at two case studies where we design the input in a way that it should produce predictable results if the method works as intended. One case study is intended to show that *dissimilar abstracts will form separate clusters*. The other case study is supposed to show that *similar abstracts will form overlapping clusters*.

### 4.1 Separation of between dissimilar clusters

To create a subset of abstracts where we should see *little overlap* between clusters, three different subsets are combined where each is comprised of patents where a particular phrase or word occurs. We want to look at cases where the word appears *in the abstract*, as well as cases where the word occurs *in the title*, since the appearance of a word in the title might signify even more relevance than its appearance in the abstract. In this case study, we looked at patents with abstracts that contained the words 'drugs', 'footwear', and 'speech'. The number of patents in each subset where the word appears in the abstract were 831, 552, and 1409 respectively and where the word appears in the title the sizes were 217, 364, and 571 respectively. There is that possibility for selection bias when we leave the search terms themselves in the abstracts and allow them to affect the results. If the search term is the only common factor in each subset, our method would not be much more useful than a simple word-search. We can check to see if this is the case by applying our method with and without the search term in the abstract before applying our method. The end result is shown in Figure 2, where each dot represents a patent and colors represent the subset each patent was taken from.

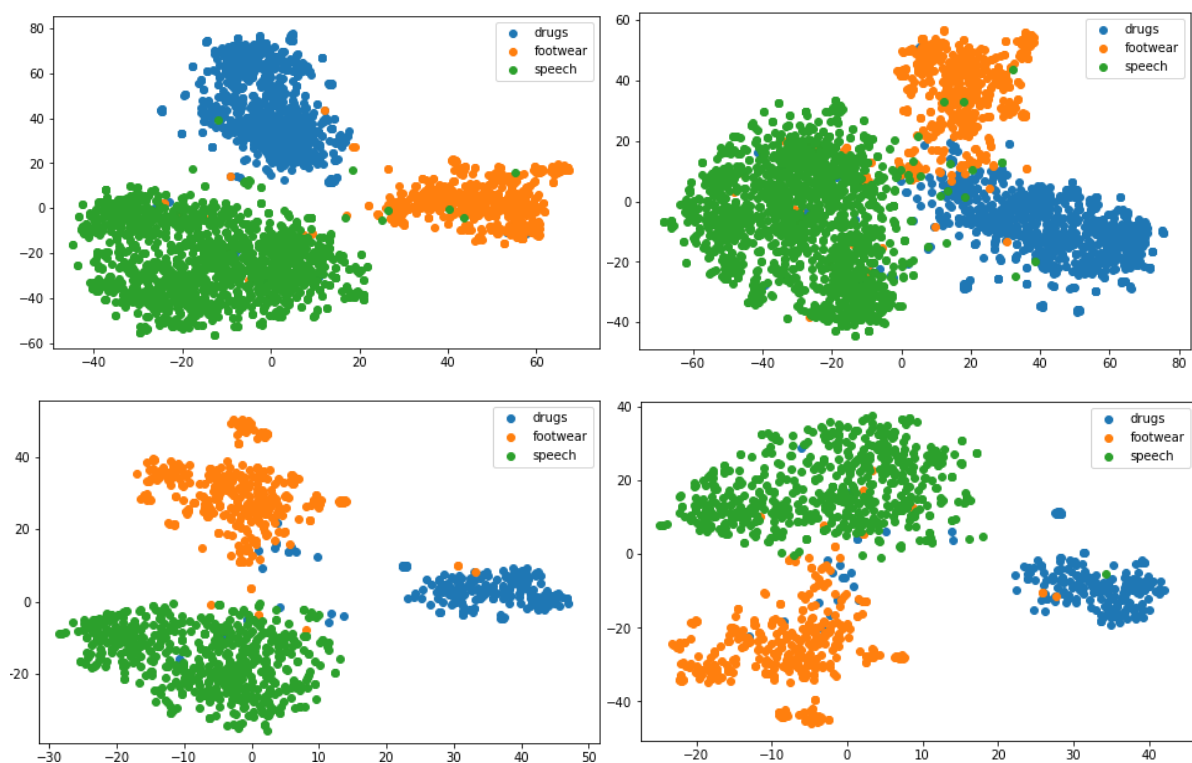


Figure 2: First case study (Dissimilar subsets) Left: search terms were removed from the abstracts before vectorization. Right: Search terms were not removed. Top: Search terms appear in abstract. Bottom: Search terms appear in the title

From Figure 2, we see very clear clustering with negligible overlap between the subsets in all cases. We can therefore reasonably assume that, at least in extreme cases such as these unusually dissimilar subsets, our transformation was able to create a vector space where locality is somewhat tied to similarity of the abstracts content. Whether the word appears in the title or the abstract did not seem to have much of an effect and neither did removing the search terms from the abstracts beforehand.

#### 4.2 Overlapping of similar clusters

The second case study was designed in such a way that we expected there to be significantly more overlap in the clusters. The phrases used to create these subsets were ‘*drugs*’, ‘*surgery*’ and ‘*physiological*’. We expected to see three overlapping clusters for since all are likely to include similar medical and biological terminology.

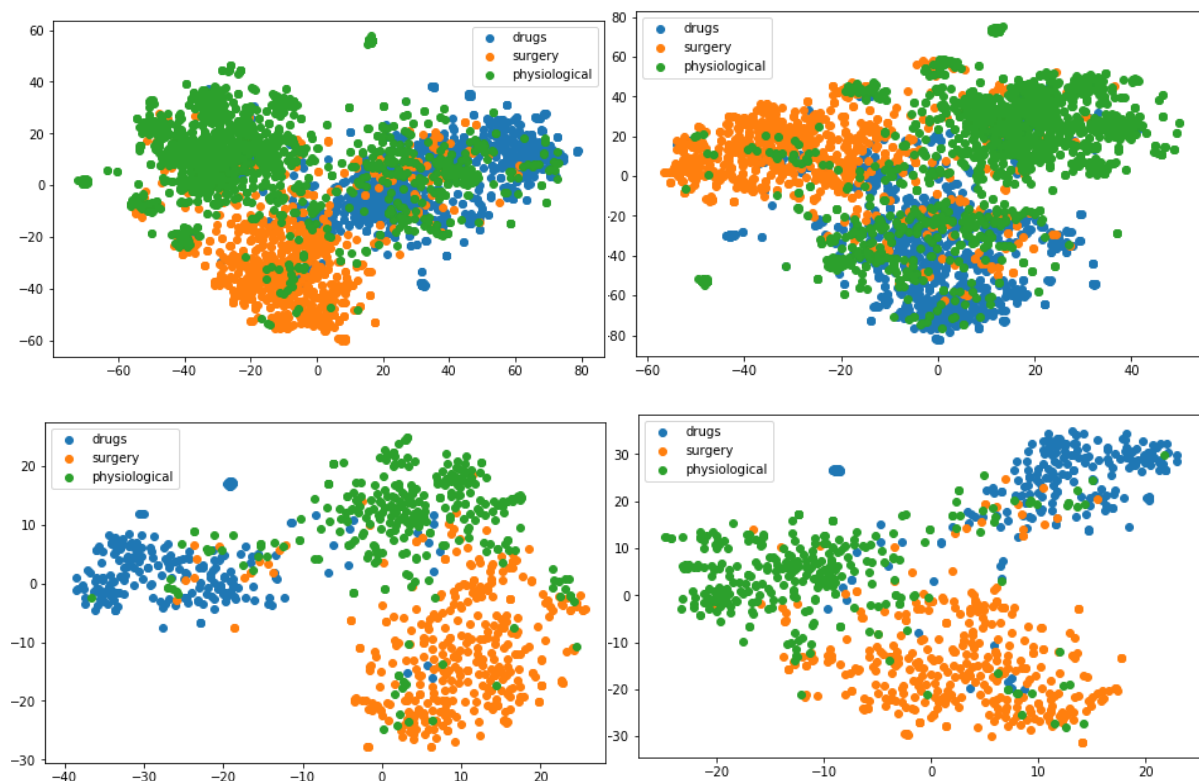


Figure 3: Second case study (similar subsets). Left: search terms were removed from the abstracts before vectorization. Right: Search terms were not removed. Top: Search terms appear in abstract. Bottom: Search terms appear in the title

The result from the second case study can be seen on Figure 3. We see different results from the first case study. We see that when we included all patents with our search term in the abstract, we get much more overlap, but when we only use those with our search terms in the title, there is significantly less overlap. A reasonable explanation is that if a word appears in the title, it is because the word is highly relevant to that specific patent and our samples would therefore be more specialized. In all four plots we see more overlap between clusters in our second case study, than in our first, indicating that our assumption is right; our transformation conserves similarity as locality in the BoW vector space

## Conclusions

What we wanted to show in this paper was that patent abstracts could be used to find similar patents. We attempted to show this by transforming the abstracts into a BoW vector space where similar patents would be close to each other and dissimilar patents would be far apart. We then used t-SNE to visualize the locality in this vector space. With two case studies we showed that:

- Similar patents will form clusters together.
- Similar clusters will overlap more than dissimilar clusters.

We have therefore shown that it could be possible to create a useful tool for venture capitalists to search for patents and explore domains of patents without being limited to simple word searches and patent categories.

## References

- Government of Canada. (2016, 01 25). *Writing a patent application*. Retrieved from Canada.ca: <https://www.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/wr01434.html>
- Leskovec, J., Rajaraman, A., & Ullman, J. D. (2011). *Mining of Massive Datasets*.
- Maaten, L. v., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2579-2605.
- PatentsView. (2019, October 08). *Data Download Tables*. Retrieved from PatentsView: <https://www.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/wr01434.html>
- PatentsView. (n.d.). *About*. Retrieved from PatentsView: <https://www.patentsview.org/>
- WIPO. (n.d.). *Patents*. Retrieved from wipo: <https://www.wipo.int/patents/en/>

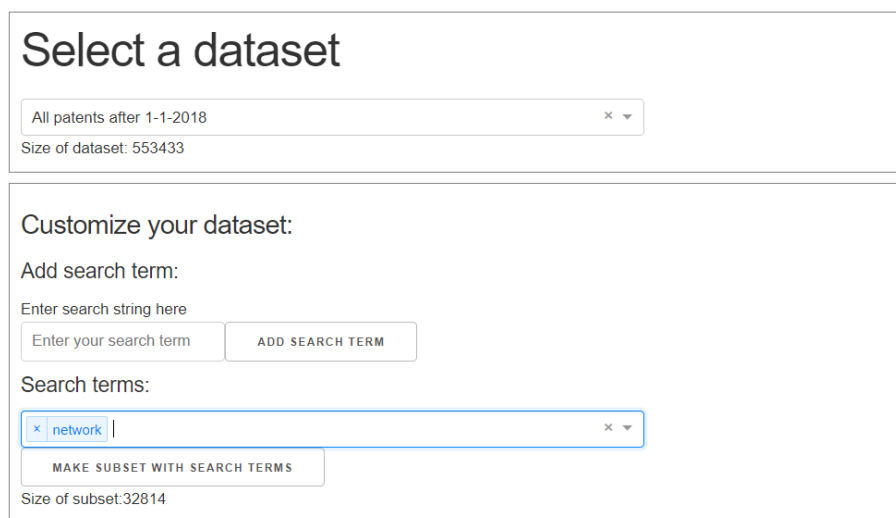


## Appendix

### Demonstration of practical application

Once we saw it was possible to cluster patents in this way, we tried to implement these methods in simple interactive tool that can demonstrate the usefulness of this method on a real-life example. The example used here is the intention to investigate the distribution of patents in the subfield of *'network patents specifically relating to social networks'*.

First the user searches for all patents, with and after the year 2018, containing the word network in the abstract. The tool finds 32'814 such patents. The tool automatically reduces the size to 10'000 before the next step by random selection to reduce computation time and memory required.



Select a dataset

All patents after 1-1-2018 x

Size of dataset: 553433

Customize your dataset:

Add search term:

Enter search string here

Enter your search term ADD SEARCH TERM

Search terms:

x network x

MAKE SUBSET WITH SEARCH TERMS

Size of subset: 32814

Figure 4: User interface for patent space visualization tool

The user then goes to the next tab, specifies some keywords for labeling the patents. The user picks the words 'neural', 'wireless' and 'social', since the user wants to see how much these subdomains overlap. The user finds that these subsets separate themselves quite a bit in the 'network' domain. The user then hovers over some points in dense clusters, attempting to find other 'network' subdomains.

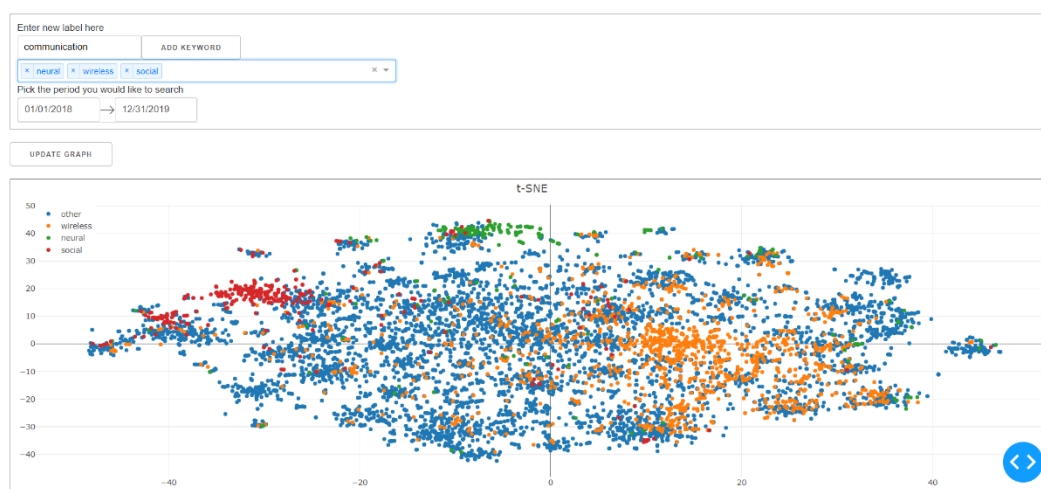


Figure 5: User interface for patent space visualization tool with the results of the search and patents colored based on if they contain a certain term or not



The user finds the words 'communication', 'packet' and 'virtual' to be common in some clusters and adds these terms to be labeled. This gives a much more detailed picture of subdomains and their relation to social networks.

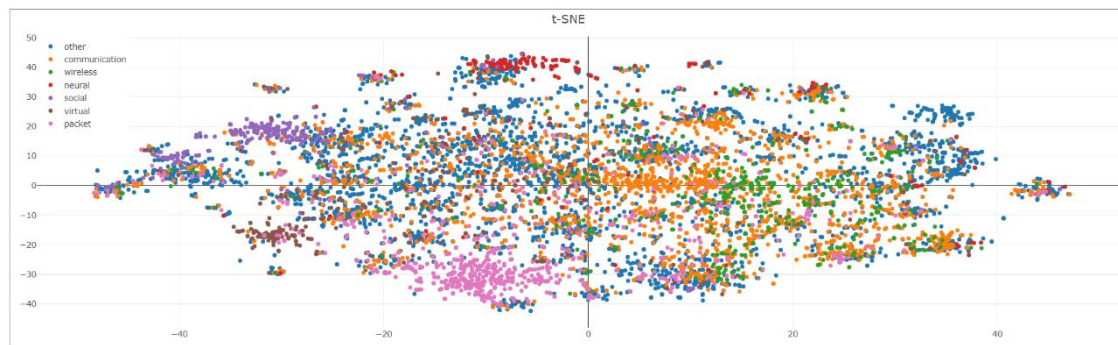


Figure 6: User interface for patent space visualization tool with the results of the search and patents colored based on if they contain a certain term or not, with more colors and more terms that show the clustering of patents with certain terms

The user then finally drags a shape around the most prominent clusters of patents labeled with the 'social' search term and then goes to the next tab.



Figure 7: : User interface for patent space visualization tool with the results of the search and patents colored based on if they contain a certain term or not, with more colors and more terms that show the clustering of patents with certain terms. Here we also see highlighted, a region with patents containing the words "social"

There, the user is able to see the most prominent patent holders in this domain. To no surprise, the biggest number of patent in the selected are belongs to *Facebook*, then followed by *IBM* and *Google*.

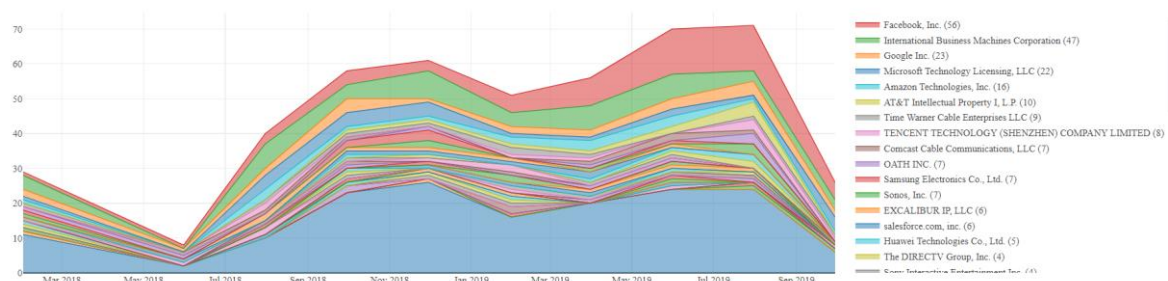


Figure 8: The second part of the UI with a timeline of the highlighted patents from the previous image and a breakdown of which companies own these patents.

The user is interested in seeing how the acquisition of patents by Facebook has developed and if there is some trend that can give insight. The user double-clicks on Facebook's label in the legend and is presented with a similar graph where Facebook's patent history has been isolated.

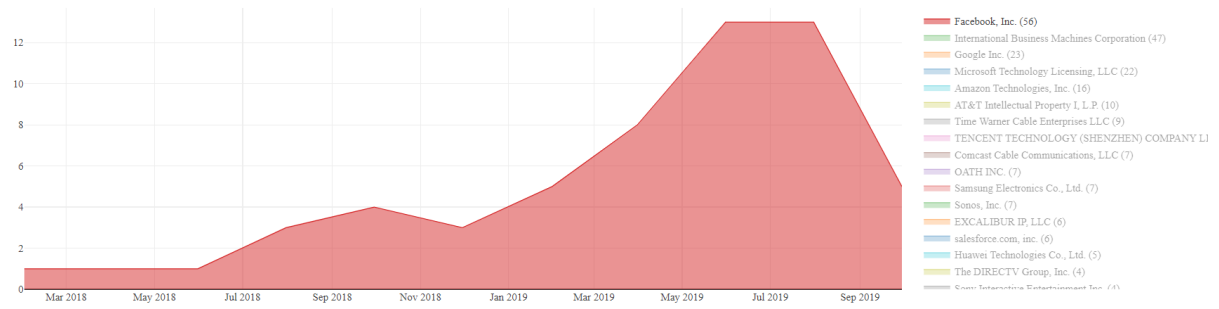


Figure 9: The second part of the UI with a timeline of the highlighted patents from the previous image, but now one of the companies (Facebook) is selected and we see how the number of patents they own over time has grown in recent years.