Capstone project 2: Milestone Report 2

Project Title:

Application Recommendation System for Biodegradable Polymer

< Abstract >

The goal of this project was to make an application recommendation system that gives some hints about potential applications for a given polymer by a user. A biodegradable polymer was chosen as a polymer kind for this project. The data acquisition, wrangling, and the exploratory data analysis were reported at Milestone Report 1. In this report, the modeling part was described.

Two application recommendation systems were created. One system accepts one patent and another accepts two IPC codes as input by a user, and they provide 10 potential applications. The recommendation systems use two features to extract potential applications, a network and sentence similarity, to maximize the accuracy of recommendation. The network has information about patents and IPC codes, and is used to extract the neighbor patents of an input patent (or two IPC codes.) Sentence similarities of patent abstracts were calculated by cosine similarity. The similarities are used to prioritize the neighbor patents so that the system can pick up the most similar patents and therefore the most prospective applications.

< Table of Contents >

1. Modeling

Network analysis and a sentence similarity were combined for the recommendation systems to maximize the accuracy of recommendation. A network allowed us to easily extract patents sharing IPC codes (*What is IPC code?* See Milestone Report 1) with an input patent. Although the extracted patents were guaranteed to share at least one feature (IPC code), the number of the extracted patents could be large (because some patents had many IPC codes). That was why the sentence similarity of the abstracts was used to prioritize them.

First, the sentence similarities between patents were calculated. Second, the network was built. Finally, the recommendation system was created.

1.1. Sentence similarity

As I reported at Milestone Report 1, the abstracts in the dataset are written in 19 kinds of languages (see 2.6. Language variety). Because I understand English and Japanese and they are the most major languages, I have decided to use the abstracts written in English or Japanese. The Japanese abstracts were translated into English (see

3.1. Translate Japanese to English.) As a result, the sentence similarities were calculated between 7,355 patents (90%.)

Tf-idf vectorizer was chosen to transform the tokens into a vector so that the importance of each word was standardized. A unigram, bigram, and trigram were used because compound names were often composed of several words. For example, methyl methacrylate, poly (1,5-dioxepan-2-one), or poly (ethylene -co- acrylic acid). They should be considered in the calculation. To reduce the complexity, the tokens appearing less than 3 patents were ignored. The tokenized abstracts of 7,355 patents were vectorized on the conditions. As a result, the vectors had 45,438 dimensions.

Next, the sentence similarities were calculated by cosine similarity to the vectors. Cosine similarity is often used to measure the similarities between texts. To evaluate the accuracy of the similarities, the number of common IPC codes was counted. IPC codes are not exhaustively assigned, but the tendency would be shown. All of the patents (7,355) were sorted by the similarity scores to each of the patents. Then, the scores were averaged at each rank. The top 20 were shown in Figure 1. The higher similarity scores, the more common IPC codes. The similarity calculation tended to be accurate. The similarity scores were used for the recommendation system.

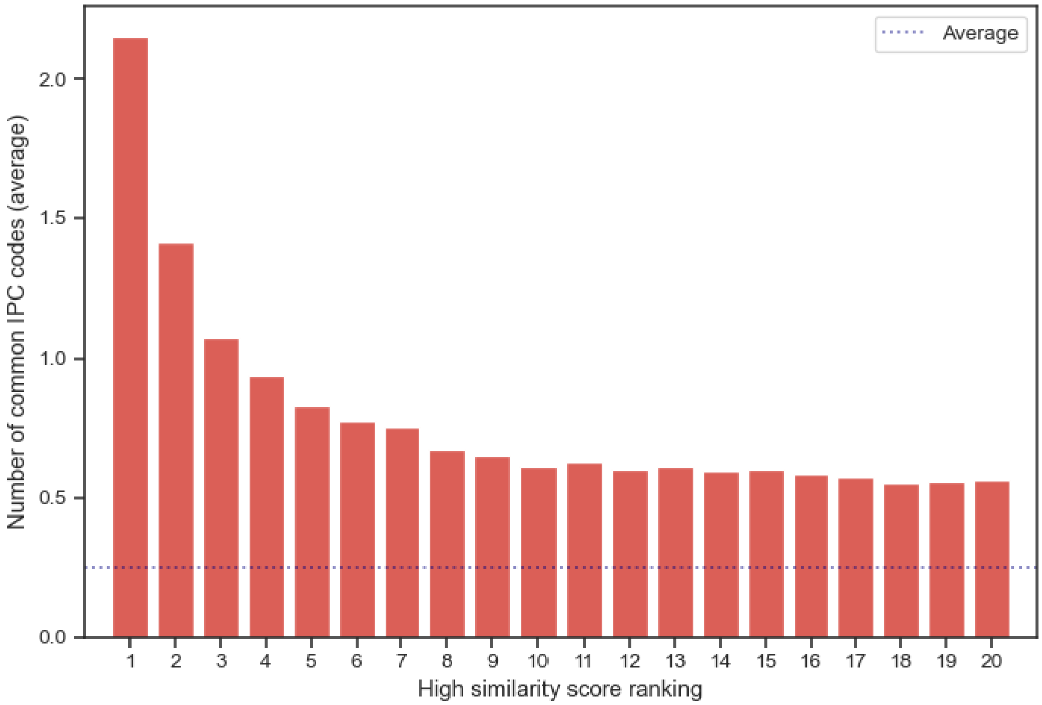

Figure 1. Average number of common IPC codes (Top 20)

1.2. Build a network

The second feature of the recommendation system was a network. The network was composed of nodes and edges. There were two kinds of nodes: patent nodes and IPC code nodes. A patent node was connected to IPC code nodes by edges if the patent had the IPC codes. There were 8,182 patents and 4,344 IPC codes in the data, and the total number of nodes was 12,526. The number of edges was 46,012 in total.

Some IPC classes had been chosen as applications (see Milestone Report 1). Each IPC code node was labeled using the list, and 2,153 out of 4,344 IPC code nodes were categorized as applications. Because the recommended applications would be chosen from the existing application IPC codes in the data, the system would show 10 out of 2,153 applications each time the system runs.

As an example, I introduce a part of the network around 'JP271085664.' The patent 'JP271085664' had four IPC codes, and the four IPC codes were connected to 2,313 patents in total. Here, I would like to visualize the network. Because it's very big, I picked up one IPC code, 'A61L 31/00', and visualized the network having nodes of 'JP271085664', the four IPC codes and the neighbor patents of 'A61L 31/00' (Figure 2). 'JP271085664' was connected to the 4 IPC codes: 'A61L 31/00', 'C08K 5/521', 'C08L 101/00', and 'C08L 67/00'. Then, 'A61L 31/00' had a connection with 97 patents. According to the network plot, 'C08L 101/00' and 'C08L 67/00' have some connections with the patents that 'A61L 31/00' was connected to. This network was a part of the original graph. The whole graph would be highly connected more with one another.

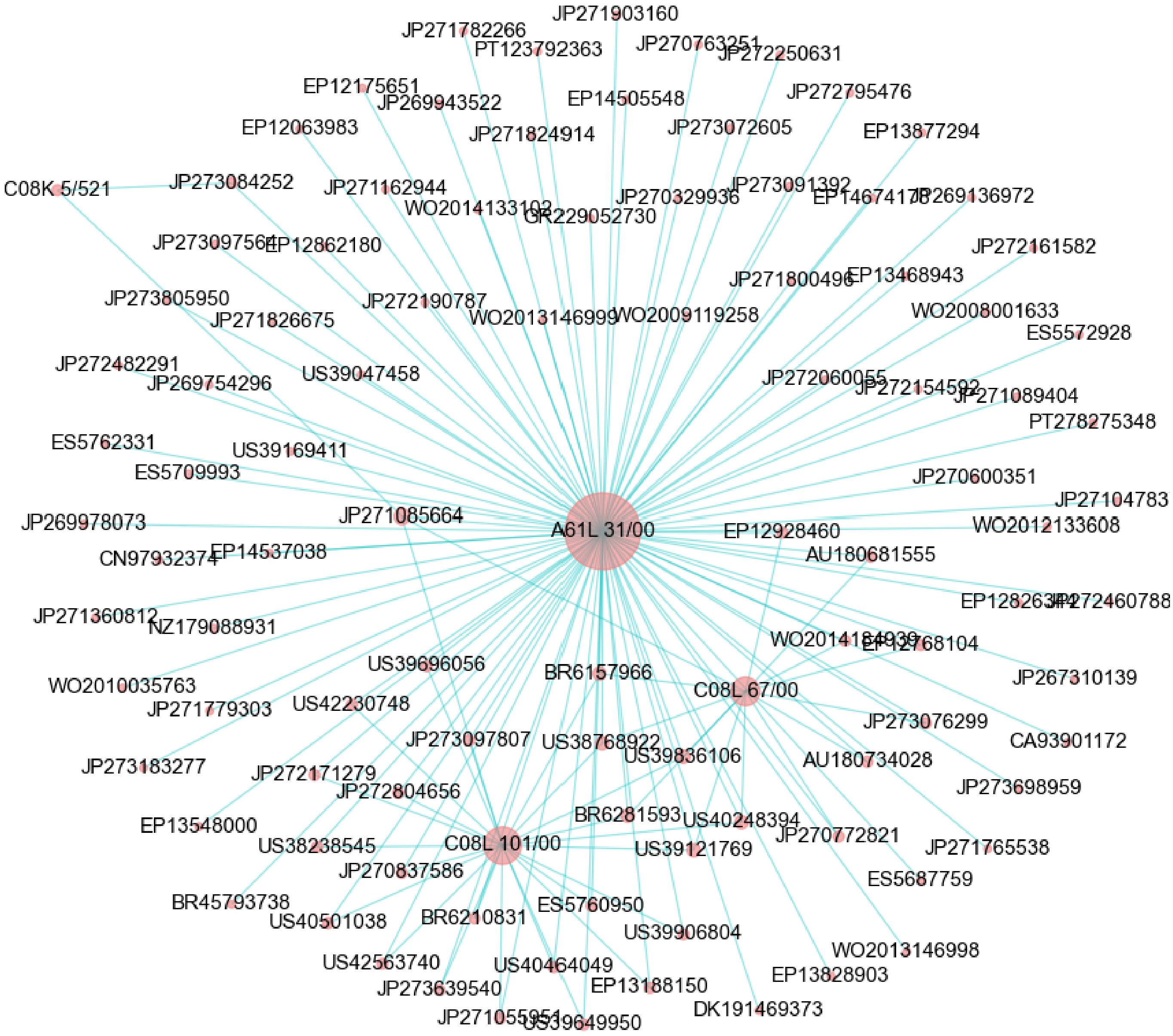

Figure 2. Network around JP271085664

The graph G had two kinds of nodes (patent nodes and IPC code nodes.) Next, I made a new graph having only patent nodes. The new graph was made by projecting the relationships between patent nodes and IPC code nodes. For example, if a patent (P1) had some IPC codes (IPC1 and IPC2) and another patent (P2) had some IPC codes (IPC1, IPC3), it was said that P1 and P2 were indirectly connected through IPC1. In this case, the new graph had an edge between P1 and P2. Because the new graph had only the patent node, there were 8,182 nodes.

Degree centrality is a rate of the number of edges to the potential number of edges. For example, if it is 1, the node is connected to all the other nodes, and if it is 0.5, the node is connected to the half of the other nodes. The degree centrality distribution of the patent projection was visualized in Figure 3.

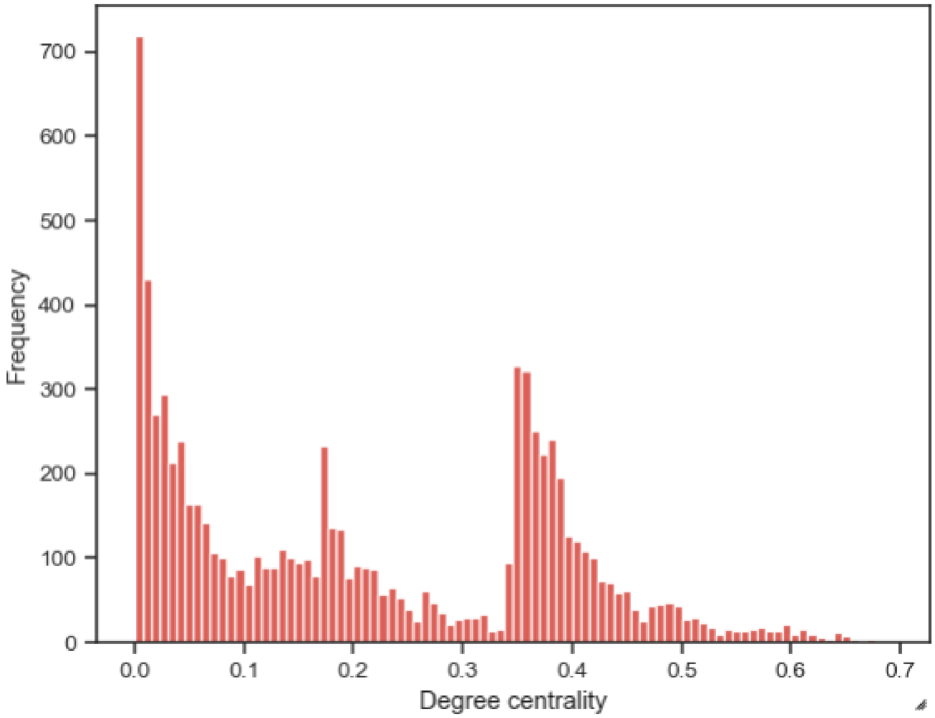

Figure 3. Degree centrality distribution of the patent projection graph

There were three peaks around 0, 0.18, and 0.36. Because the graph had 8,182 nodes, 0.18 meant more than 1,470 connections and it was a lot. That was why the sentence similarity was used to prioritize them.

On the other hand, if a degree centrality is very close to 0, the patent has only a few edges (connections with the other patents). Then, the neighbor patents might or might not be connected to the other patents. If not, the network is closed. There were 46 subgraphs in the original graph. There were one huge subgraph and 45 small subgraphs. The small subgraphs had only one component, that is, each small subgraph was composed of one patent. They do not have any neighbor patents, and it is impossible to extract the applications from the neighbors. On the other hand, all patents are connected by 'C08L 101/16' actually. Then, in the case, recommended applications would be extracted from patents having the most similar abstract.

1.3. Create a recommendation system

Two versions of the recommendation systems were created. The first one accepts one patent from a user and recommends up to 10 applications. The second one accepts two IPC codes from a user and recommends up to 10 applications.

1.3.1. Create a recommendation system accepting one patent as input

When a user already published a patent in the biodegradable polymer field and wants to know other potential applications for their polymer, this recommendation system would be used. The patent that inputted by a user should be in the data.

For example, suppose that Company_A published Patent_1 about a biodegradable suture using polyglycolide, and they want to know that their polyglycolide could be used for the other applications. They would input Patent_1 in the system and would get 10 potential applications. The patent could be a competitor's patent.

Two cases should be recognized to create the system successfully. The first case was that the input patent did not have any neighbor patent. Remember that 45 patents formed individual subgraphs. In this case, the network would not be able to be used to extract similar patents, but the sentence similarity would be. Another case was that the input patent did not have an English or Japanese abstract (10%). In this case, the extracted neighbor patents by the network would not be able to be prioritized. I decided to show a warning and use the unsorted patent list for the following steps. These two cases could be combined. The handling was shown in Table 1.

Neighbor patents	English or Japanese abstract	Sign	Handling
Yes	Yes	-	Extract neighbor patents using the network and prioritize them by the sentence similarity
No	No	Error 2	Exit the system
No	Yes	Warning 1	Extract similar patents in similar order by the sentence similarity
Yes	No	Warning 2	Extract neighbor patents using the network and use the neighbor patents without sorting

Table 1. How to deal with exceptional cases

The workflow of the recommendation system was shown in Figure 4. When a patent is given by a user, the neighbor patents are extracted. The neighbor patents are sorted by the similarities of the abstracts. When prioritizing the neighbor patents, some patents might not have an English or Japanese abstract. In this case, the average score of the input patent is used as the similarity score. Then, the application IPC codes are extracted from the top patents. If 10 applications are not collected at one cycle, the neighbor patents of the neighbor patents would be extracted from the network, and the new neighbor patents are used to extract more applications.

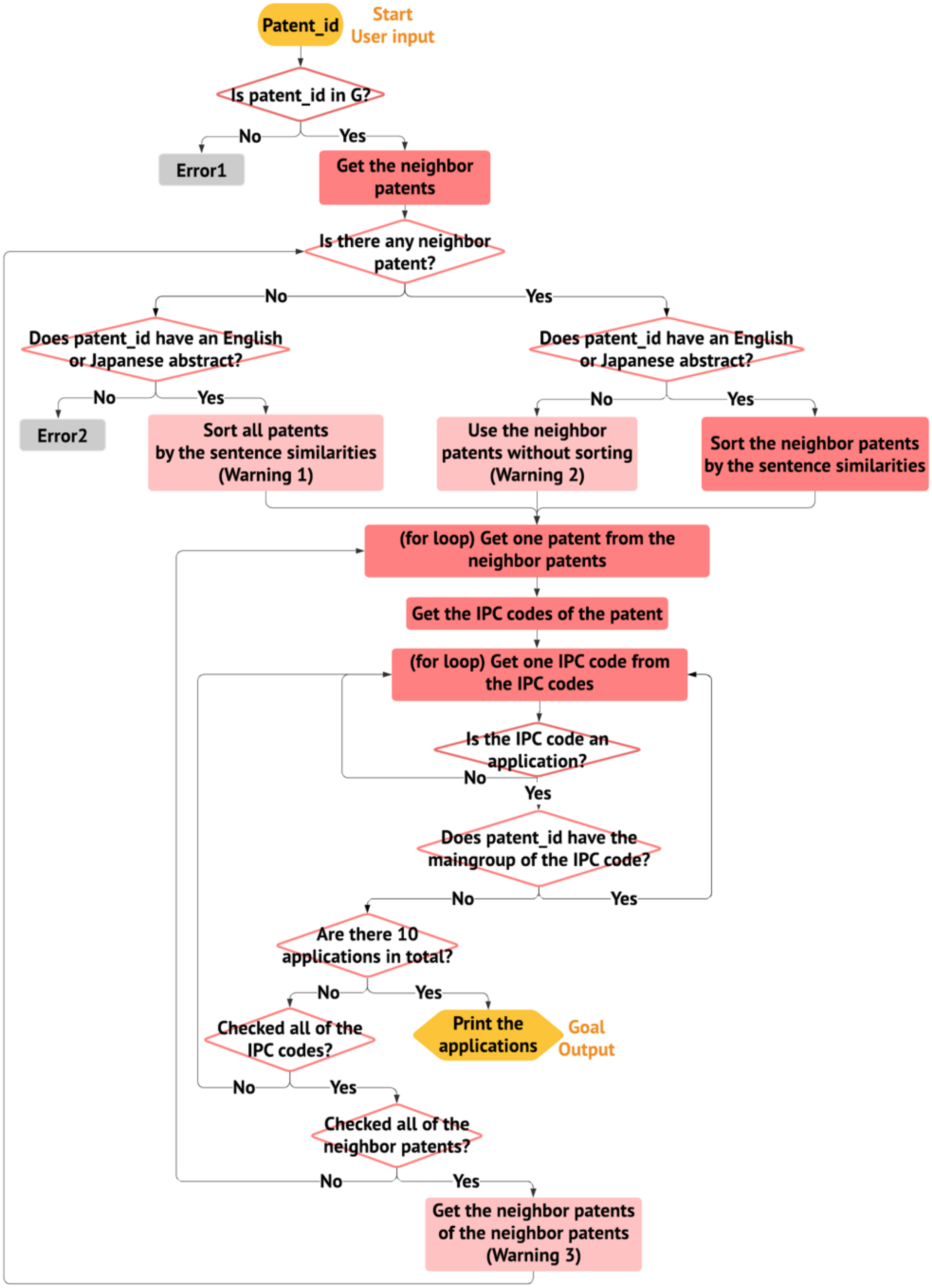
Patent_id
Start
User input
Is patent_id in G?
No
Yes
Error1
Get the neighbor patents
Is there any neighbor patent?
No
Yes
Does patent_id have an English or Japanese abstract?
No
Yes
Error2
Sort all patents by the sentence similarities (Warning 1)
Does patent_id have an English or Japanese abstract?
No
Yes
Use the neighbor patents without sorting (Warning 2)
Sort the neighbor patents by the sentence similarities
(for loop) Get one patent from the neighbor patents
Get the IPC codes of the patent
(for loop) Get one IPC code from the IPC codes
Is the IPC code an application?
No
Yes
Does patent_id have the maingroup of the IPC code?
No
Yes
Are there 10 applications in total?
No
Yes
Checked all of the IPC codes?
Print the applications
Goal
Output
No
Yes
Checked all of the neighbor patents?
No
Yes
Get the neighbor patents of the neighbor patents (Warning 3)

Figure 4. Workflow of the recommendation system accepting one patent as input

1.3.2. Create a recommendation system accepting two IPC codes as input

When a user does not have any target patent and wants to search for some applications from IPC codes, this second recommendation system would be used. The IPC codes inputted by a user should be in the data.

For example, suppose that Company_B is developing a biodegradable suture using polyglycolide (they do not have the patent yet), and wants to know other potential applications for their polymer. They would search the IPC codes of polyglycolide (C08G 63/06) and a biodegradable suture (A61L 17/06, A61L 17/08, or A61L 17/12 (IPC is a hierarchic structure. I recommend to try several IPC codes that are related to what you want to search).) Then, you input two IPC codes (e.g. C08G 63/06 and A61L 17/08) into the system and would get up to 10 potential applications. The workflow of the second recommendation system was shown in Figure 5.

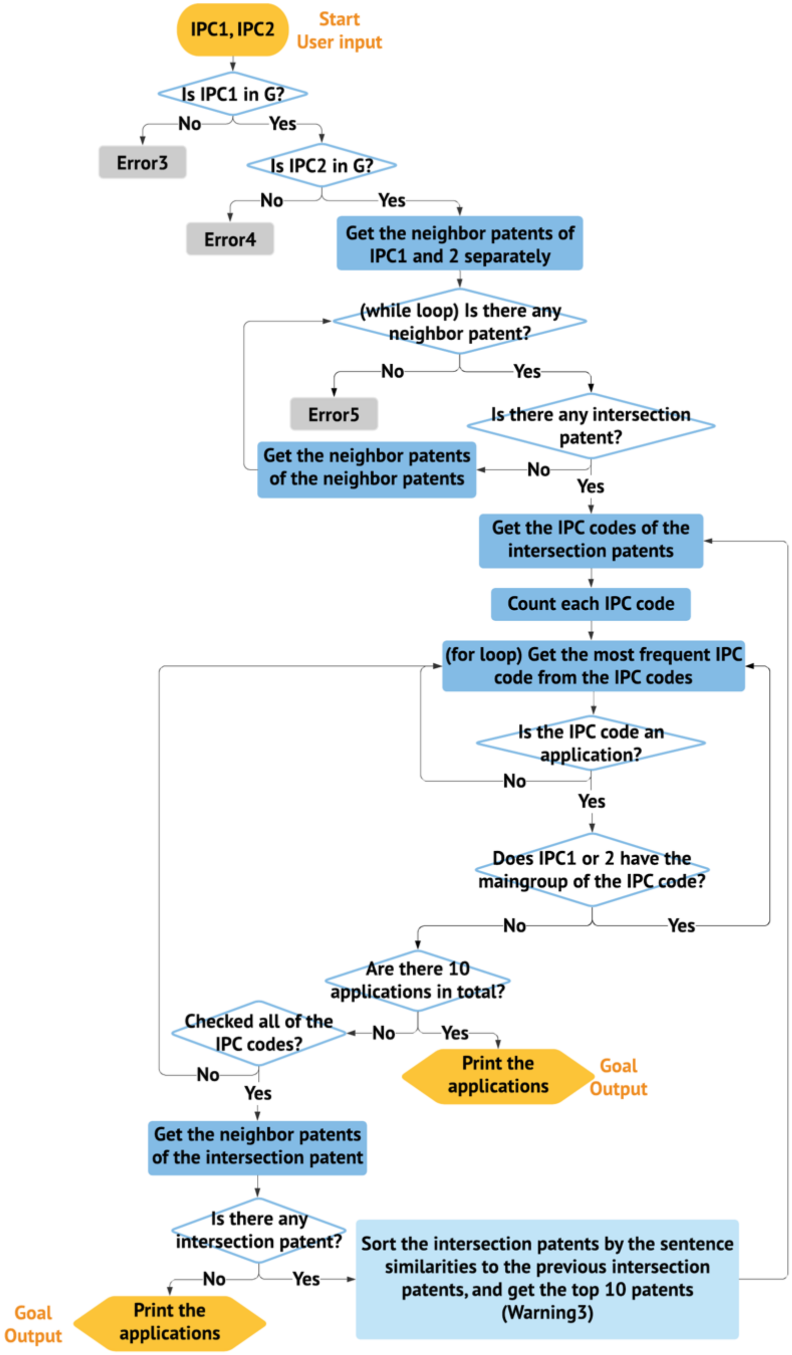
IPC1, IPC2
Start
User input
Is IPC1 in G?
No
Yes
Error3
Is IPC2 in G?
No
Yes
Error4
Get the neighbor patents of IPC1 and 2 separately
(while loop) Is there any neighbor patent?
No
Yes
Error5
Is there any intersection patent?
Get the neighbor patents of the neighbor patents
No
Yes
Get the IPC codes of the intersection patents
Count each IPC code
(for loop) Get the most frequent IPC code from the IPC codes
Is the IPC code an application?
No
Yes
Does IPC1 or 2 have the maingroup of the IPC code?
No
Yes
Are there 10 applications in total?
Checked all of the IPC codes?
No
Yes
Print the applications
Goal
Output
No
Yes
Get the neighbor patents of the intersection patent
Is there any intersection patent?
No
Yes
Sort the intersection patents by the sentence similarities to the previous intersection patents, and get the top 10 patents (Warning3)
Goal
Output
Print the applications

Figure 5. Workflow of the second recommendation system accepting 2 IPC codes as input

When two IPC codes are given by a user, the neighbor patents are separately searched, and the intersection patents are extracted. This time, what a user inputted are IPC codes, and the intersection patents cannot be sorted by the similarities. Then, I decided to use a majority voting. All of the IPC codes of the intersection patents are searched and counted, and the IPC codes are checked if it is an application from the most frequent IPC codes to least. When 10 application IPC codes are acquired, the system outputs the result. If 10 applications are not collected at one cycle, the neighbor patents of the neighbor patents are extracted from the network, and the new intersection patents are sorted by the similarities to the previous intersection patents. Then, the top 10 intersection patents are used for the next cycle.

2. Next steps

The interface is going to make so that a user can input a patent or IPC codes. Also, some unit tests will be prepared. As a summary, I will write a final report and make the presentation slides.