

**Social Analytics on Stack Overflow Questions**

Li Ling

Li Zuxian

Lyu Xudong

Shen Jiajun

Wang Kuan

Zhong Zhewei

Abstract

In modern days, more and more people, even kids, are beginning to learn at least one computer programming language. It is obvious that the demand for programming language lessons is ascending tremendously. Our project is taking a course consultant as a starting point, aiming to use social analytics techniques including community detection, social frequent pattern & centrality, and topic modelling to explore the popularity of these languages on Stack Overflow. The forum serves as a question and answer site for professional and enthusiast programmers and features questions and answers on a wide range of topics in computer programming. We suppose that Stack Overflow could, to some extent, represent the preference of the public.

# Motivation and Objective

We are aiming to achieve the following business applications.

* To understand the popularity of programming languages for this year.
* To understand which subcategories of each programming language are mostly discussed.
* To understand which applications these popular programming languages are used for.

# Dataset

In the project, we use the questions and answers data of the Stack Overflow from Kaggle. Here is the data resource link, <https://www.kaggle.com/stackoverflow/stackoverflow>. We will mainly use two tables of the dataset, Post\_Questions and Post\_Answers. The whole table includes the data from 2009 to 2020. The time period of data we use is 2009, 2012, 2015, 2018 and Jan to May 2020. Totally we have 1081283 rows answer data and 1022431 rows question in 2020. We use Google Bigquery API with SQL to query data from Google Cloud.

For each data table, we have the following schema.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Description** | **Data Type** |
| Post\_Questions. id | The unique id of the question | Int |
| Post\_Questions. title | The title of the question | String |
| Post\_Questions. body | The body of the question | String |
| Post\_Questions. tags | The tag of the question, for example, C#, python etc. | String |
| Post\_Questions. owner\_user\_id | The author’s user id of the question | Int |
| Post\_Answers. Id | The unique id of the answer | Int |
| Post\_Answers. body | The content of the answer | String |
| Post\_Answers. owner\_user\_id | The author’s user id of the answer | String |
| Post\_Answers. parent\_id | The question id of the answer | Int |

# Approach

## Community detection

In Stack Overflow, users communicate with each other by questions and answers. Their questions focus on different parts of technology they use, so they may form potential communities for different technologies. To detect the community, we run the community detection with following methods.

Firstly, in the original dataset, question posts and answer posts are in different data tables. We write the SQL code to join those two tables on answers’ parent\_id equal to questions’ id.

Secondly, because of high volume of data, we filter out the questions and answers from active users. The definition of active users is the users who post more than 10 questions or answers in a year.

Thirdly, after we get the table about questions and answers, we calculate the frequency of questions and answers relationship between each two users.

Finally, we create the graph for community detection. The graph is weighted and undirected graph. The node is id of each user. The edge represents the Q&A relationship between two users. The frequency of Q&A relationship is counted as weight of the edge. Then, we run community detection with best\_partition function in community\_louvain package.

To identify the theme in each community, we count the tags of the questions posted by the users in the community.

## Frequent Itemset Mining & Visualization

There could be more than one tag in a question. They may be a programming language and one of its sub-fields, a pair of similar languages, different versions of systems etc. In our project, we only focus on Python as it is the most important and popular area based on the results of community detection. Hence, the first step is that a table containing the co-occurrence of tags under Python category would be created based on the data.

Secondly, frequent itemset mining will be carried out based on the co-occurrence results using Apriori algorithm. We set the occurrence threshold to 1% to filter out frequent itemset. Itemset such as {Python, Pandas}, {Python, Machine Learning}, {Pandas, NumPy} and etc are discovered.

Thirdly, an ego-network of “Python” will be built to visualize the above result. To avoid creating an ego-network with only edges between the ego-node and the non-ego-nodes, edges between two non-ego-nodes will be generated if the itemset support percentage is larger than 0.01%.

Fourthly, in order to know the result better, tags including Data Wrangling, Website Related, Machine Learning, Plotting, and Basic Function, are given to each item, indicating its function in Python. For example, Pandas is tagged to be Data Wrangling and Django is tagged to be Website Related.

Finally, by plotting the interactive network visualization with R markdown, we could conduct further analysis on the relation of all generated frequent itemset. The weights of edges are shown by widths. Nodes in different groups are shown in different colors, with the relative position being the relationships of each pair of nodes.

## Text analytics

For our project: **Stack overflow data,** we conducted as series of text analysis through the *Posts\_questions* dataset.

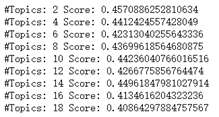
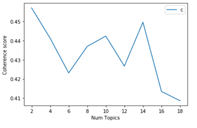
Based on our observation, the report will contain three major findings of our text analysis, which is the coherence value for finding meaningful topic, pyLDAvis and WordCloud visualization.

**(i) Finding a meaningful topic:**

In order to achieve our objectives, we built the topic model to find out our topic which contained a set of words frequently co-occurring together.

We did the Topic coherence model analysis for Pyhton through LDA, and from which, we found that higher coherence value could always indicate model which was more meaningful. Hence, we listed the top 9 meaningful LDA models and calculated their coherence score. To get a high-quality model, we used the Mallet version in topic modelling with gensim, and by doing so, the coherence value of all the models could be displayed by the graph shown below, which was easier for us to observe and analyse which topic was the most suitable one.

We ignored the extreme value from both sides, and we also made sure that our result was reasonable. From the graph, we can see that there are few reasonable peak values from the coherence score of different models, such as: 8, 10, 14. Among these reasonable numbers, the score of the 8 topics model is around 0.437, while the value of 10 topics model is around 0.442. Though the score of the 10 topics model is a little bit higher than the 8 topics one, we can see that the 8 topics model shows an uptrend, which is better than the 10 topics model with a sharp down trend. As it makes better sense to pick the model with the high CV before flattening out or dropping down, we picked the 8 topics model one.



**(ii) pyLDAvis visualization:**

After calculated the coherence score, we analysed visualization graph result to approve our previous conclusion. Thus, we did the models visualization to compare the 8 topics model and 10 topics model by pyLDAvis, which is the feature of building the graph to display the correlation between the topics as well as the topmost relevant terms for each selected topic.

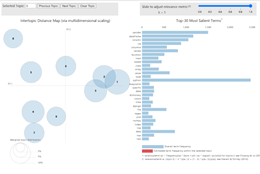
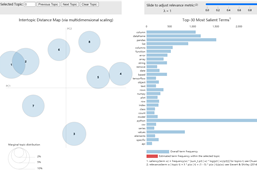
From the graph shown below, on the left hand side, each bubble plot represents a topic. The larger the bubble, the more prevalent the topic is, in another word, a good topic model will have a big and non-overlapping bubble.

On the right side, the words and bars are the Top-30 Most Salient Terms for the 8 or the 10 topics models, from which we can select each bar to represent its corresponding Terms.

A model with too many topics will typically have multiple overlaps, and this is the reason why we did not consider the 14 topics model at the beginning. According to the graphs shown below, the graph on the left shows the 10 topics model, from which, we can see that topic 1, 2, and 3 are overlapped with each other. While from the graph on the right for the 8 topics model, we can see only 2 bubbles are slight overlapped when other topics are evenly distributed across all the four quadrants. Therefore, the 8 topics model is more reasonable and meaningful.

After we selected the 8 topics model, we did a deeper analysis, which is our meaningful LDA model.

As shown in the graph for the 8 topics model, when we click each bubble, it will display the percentage of its contribution. Topic 1 is 16.5%, Topic 2 is 16%, Topic 3 is 12.2%, Topic 4 is 11.9%, Topic 5 is 11.6%, Topic 6 is 10.8%, Topic 7 is 10.7%, and Topic 8 is 10.4%.



**(ii) WordClouds visualization:**

When we select each topic, we can see the top relevant keywords, and these keywords can also be displayed by WordClouds.

WordClouds graph is another way to visualizing our topic model, and we got a few insights from the keywords. Each topic has its most Obvious keywords: Topic 0 is focusing on the basic functions; Topic 1 is related to Machine learning, especially refer to the package or library; Topic 2 is about files; Topic 3 is related to graph or plot drawing, as known as visualisation display; Topic 4 is related to the arrays and matrices, which focusing more on the data structure; Topic 5 is obviously related with pandas, Series and DataFrame, which is more about the data analysis; Topic 6 is about error handling; and the last one, topic 7 is about multi-functions.



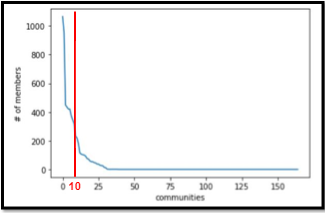
From our analysis and observation, we found that there were some overlapping among users’ post questions, so we could advise Stack overflow can separate those top question topics by different plates, also make their website be more objective, and increase the efficiency for their users. We also noticed that there are some users with related or similar questions, so Stack overflow can also try to create or upload some common Questions or Answers under the corresponding topics.

From the keywords and the text inside of WordClouds and pyLDAvis visualization graph, we would like to suggest Stack Overflow to focus on the most obvious topics, such as the second topic and the fourth topic, which are related to machine learning and visualization. The second topic has some interesting subjects about Tensorflow or Django, while the fourth topic is the most relevant one with plot, matplotlib, calculate, geopandas and so on. Both of them are relevant to the hot technology fields, so we would suggest Stack Overflow to try to push some interesting online courses that cover machine learning or visualization display for different IT background level users in the future.

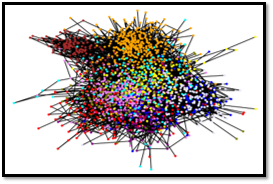
# Results

**Community Detection**

In the result, there are total 200 communities. The number of users in the community drop dramatically as the community id increases. It is a long tail graph. We choose top 10 communities as our target communities to do analysis.

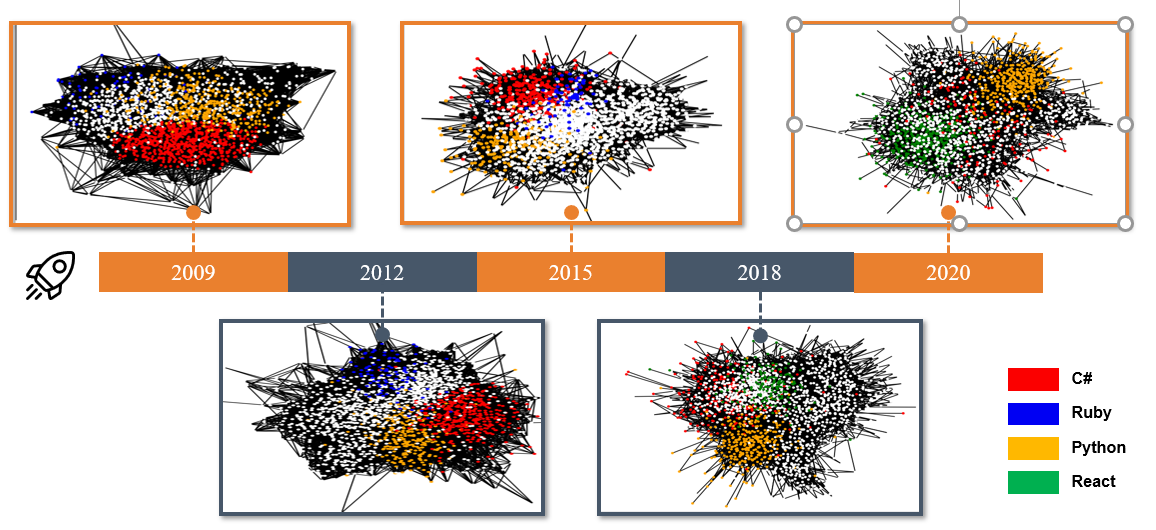


Top 3 communities in 2020 are JavaScript & React as the first, python as the second and Java as the third.

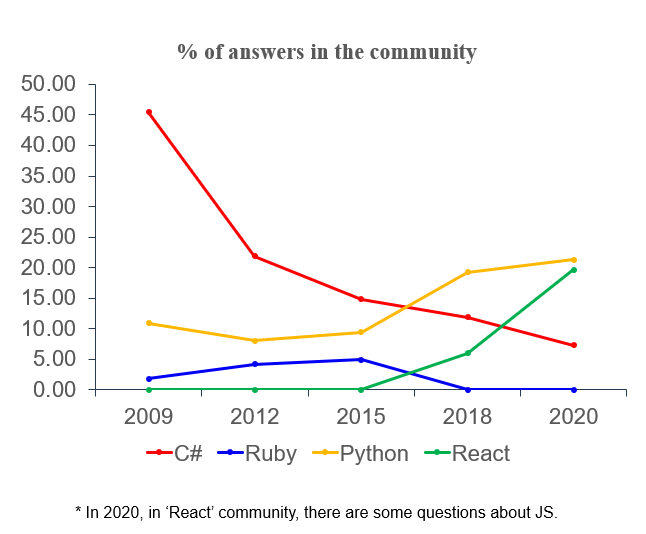
 

The community detection results of year 2008, 2012, 2015, 2019 and 2020 are shown below. Here we only show 4 dominant communities in color and other small communities are all in white.

We conducted 4 years’ community detection in order to study how communities evolve overtime. As you can see from the graphs, as time went by, some communities declined along with the rise of other new communities. Specifically, C# community, representing by red color became smaller and smaller, and Ruby community, representing by blue color, disappeared after 2015. On contrast, the React community appeared in 2018 and has been growing rapidly since then. Besides, the growth of Python community is very significant too.

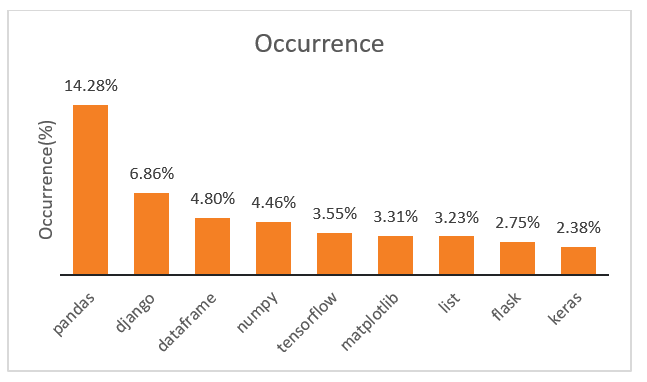


We can also refer to the number of questions and answers discussed in each community to reflect the community evolution. According to the line chart below, the red line shows a downward trend and the blue line drops to 0, meaning that the discussion related to C# and Ruby get fewer and fewer, which reflect the decline of these two communities. On the other hand, we noticed the sharp increase of the green and yellow lines. We learned that React is open sourced in 2013 and gets more and more popular among developers. Python, without surprise, grow up gradually, with the fashion trend of Machine Learning and Artificial Intelligence, because of its simplicity and natural language-like syntax.

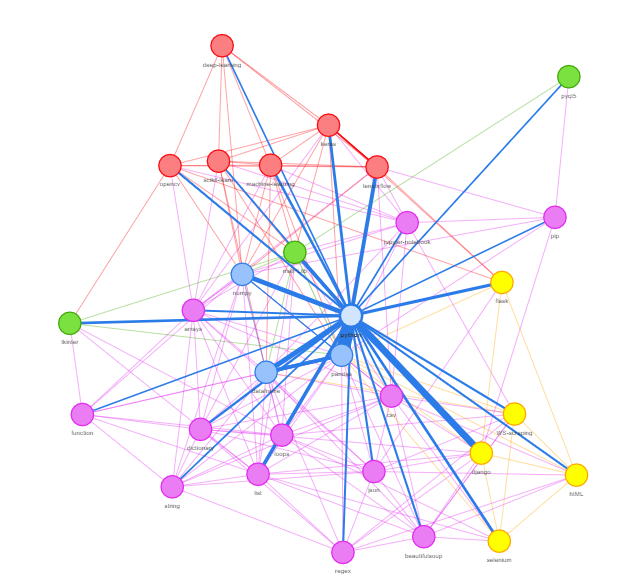


**Frequent Itemset Mining**

The bar chart describes the ranking of percentage of each tag under Python category. It is obvious that “pandas” has the highest percentage of 14.28%, twice more than the second tag “Django”. Except for “Django”, which is more about the web framework, all the top 3 tags (“pandas”, “Dataframe” and “NumPy”) are under data wrangling category. It could be inferred that data processing is the most popular topic under Python among users on stack overflow forums.

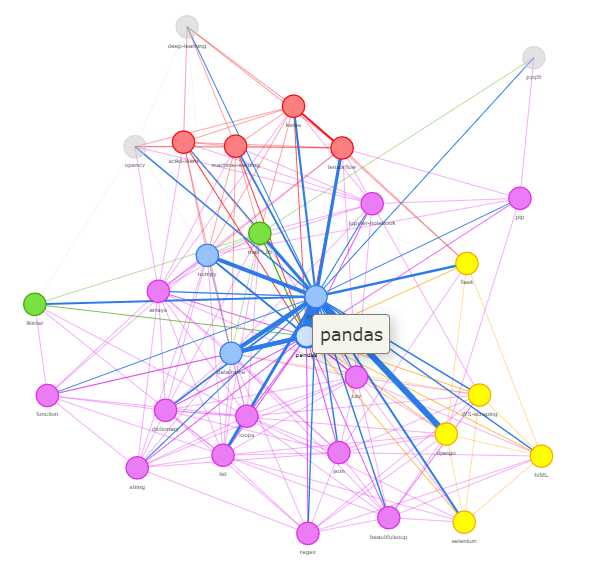
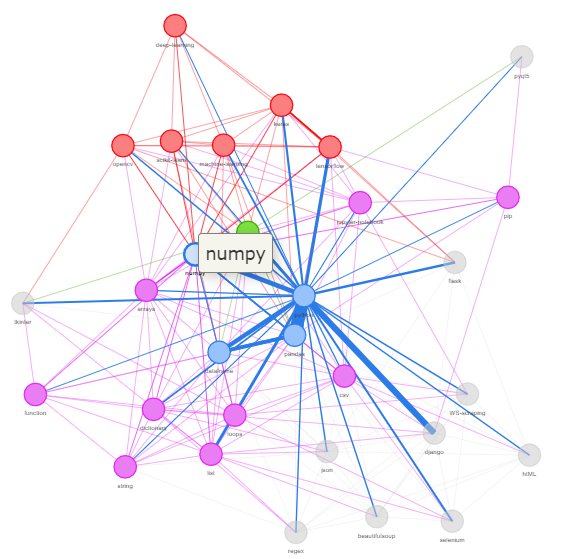
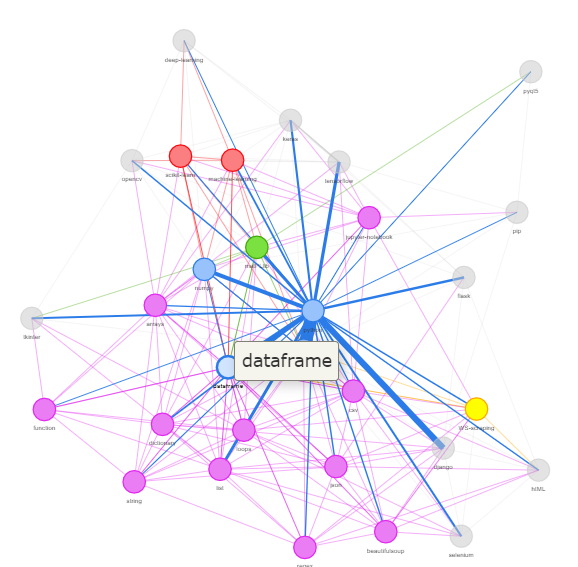


An ego-network of python is built and visualized according to the relative distances among nodes.

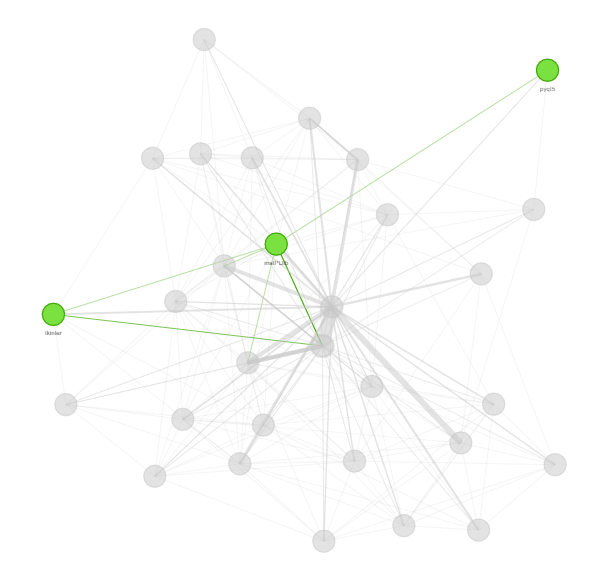
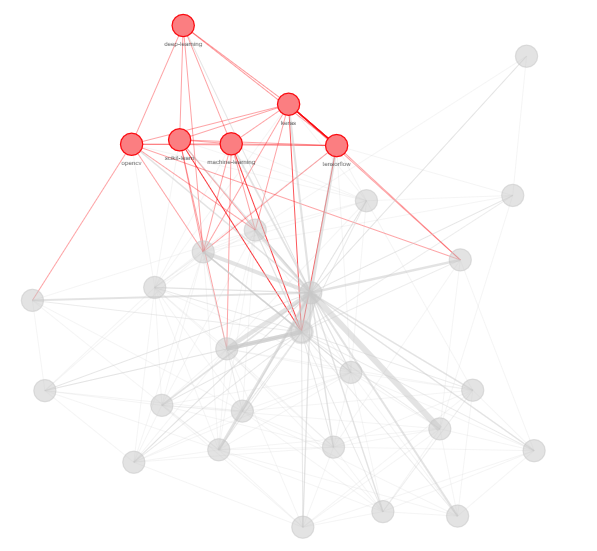
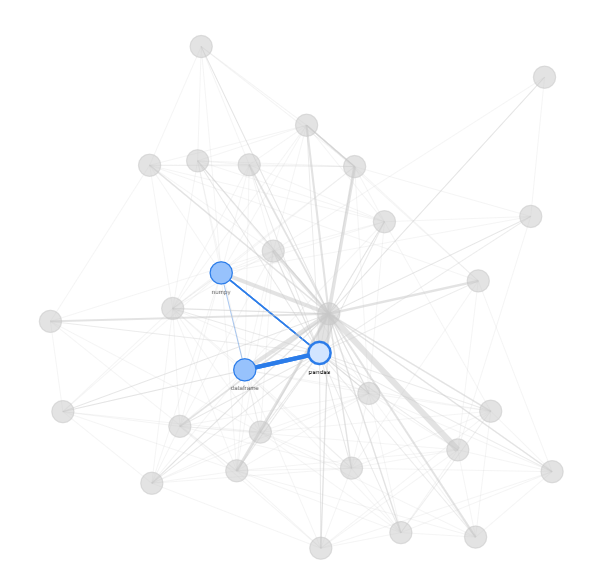


The blue edges are frequent itemset mined from the dataset. The wider the edge, the larger the support. Edges with support larger than 0.01%, will also be displayed here for better visualization. In the middle of the graph is the node “python”. It connects to every node in this network.

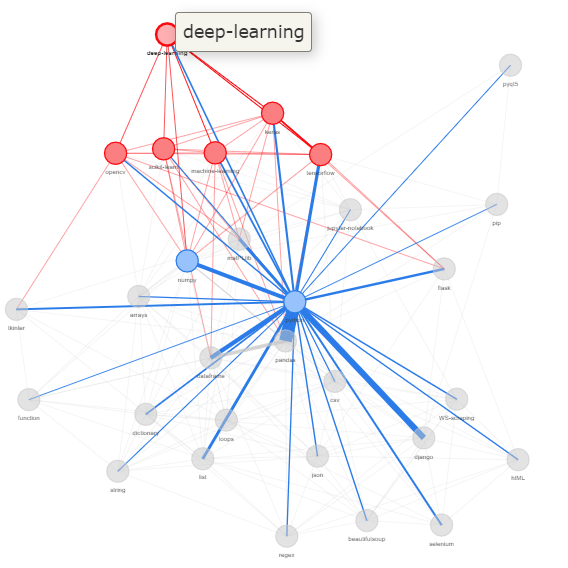
The position of data wrangling group is in the middle near ‘python’. When you choose the three nodes respectively, Pandas, NumPy, DataFrame, basically you can highlight the whole network, showing the importance of data wrangling.



Data wrangling & ML are densely distributed, with great inner connections. however, plotting-related nodes are sparsely distributed, meaning that they don’t have much collaboration with one another.



Also, we could see that the node ‘deep-learning’ is mostly connected to nodes in the ML group. It doesn’t relate much nodes in other groups.



All in all, python is the most popular language nowadays. Data wrangling (blue nodes) is the most important part in Python language. Basic functions (purple nodes) are embedded into every part of python. On the contrary, Machine Learning (red nodes) and Website-related nodes (yellow nodes) are more specialized in their particular fields, related to artificial intelligence and front-end UI design respectively.

# Contributions

Li Ling: Text analysis part presentation and report writing.

Li Zuxian: Social Itemset Mining, Ego-network building, Ego-network visualization.

Lyu Xudong: Query data from Google Cloud with Bigquery; build graph and run community detection.

Shen Jiajun: Social Itemset Mining coding including data pre-processing and graph building.

Wang Kuan: Text analysis coding and conclusion & future work.

Zhong Zhewei: conduct community detection for 5 years with Xudong; Build community network graph.

# Limitation & Future Work

# Limitation

In complex networks, one node may closely relate to multiple groups, that is, there may be overlaps between communities. We may use weighted graph or try other more advanced algorithms to mitigate the problem. The result of topic modeling may not be very clear. Topics may overlap or it may be hard to understand what each topic is about or determine the context in which the words are used based on the top topic words. We can change the number of topics when build the model, do more data cleaning like adding specific stop words and do some feature engineering.

Future Work

We also admin there is a lot of other meaningful work we can do in the future. Firstly, we can identify the most difficult and challenging topics commonly faced through sentiment analysis which is not done in our current project yet. Sentiment analysis will help detect more insights about each question topic. With all the information, we can develop more useful programming courses to meet users’ urgent needs. Secondly, we also can improve our topic modelling by determining more details of subject difficulties for users. Specifically, it means we will find out deep level of sub-topics for python or even specific library, function and error message. Additionally, more than size thousand new questions are asked on stack overflow every weekday. However, currently about 6% of all new questions end up closed. So another future work we can do is to predict closed questions and highlight in website for non-closed but urgent questions to improve our user experience

# References

<https://ieeexplore.ieee.org/abstract/document/6921608>

<https://www.kaggle.com/paultimothymooney/how-to-query-the-stack-overflow-data>