Capstone Project 2 Milestone Report

Semantic Search for Patents

Summary

Semantic search on patents has been explored for years and is to discover other patents with similar contents instead of keywords. It can increase the efficiency of finding patents with similar contents for patent attorney, R&D departments and patent software organizations. This report explains a new approach of automatically discovering semantically relevant patents with AI/ML models with the small datasets. This reports takes a small manually selected seed focusing on a topic, such as “purify water” and “filter water”. Then the key information is extracted from the seed and expands seed to a larger document sets. Semi-supervised model is applied to classify this larger document sets with the small human-selected anti-seed.

Project Overview

- **Project Background**

Patent Semantic search is to search patent document based on the meaning of search words instead of keywords. Currently, the rule-based software engineering is the most common approach in real products. However, it requires very sophisticated techniques such as software engineering, computing linguistics and domain experience. To keep minimal manual efforts and improve efficiency in search, some data scientists are also leveraging Neural Network in supervised machine learning for semantic search. For example, Hamel Husein uses Deep Learning to build semantic search for arbitrary objects. But the huge amount of labeled data are needed, which also prevent AI/ML from further use in semantic search.

- **Project Scope**

Inspired by Google’s research work, we can create a AI/ML semi-supervised model to automatically explore semantic patent search with similar schema. Given that patent contains many technical jargon, we select a topic “purify water” and “filter water” to simplify the project. A small seed dataset containing “purify water” and “filter water” are manually selected and

labelled. And synonym phrases would be extracted from this dataset. Then these phrases, as keyword in patent abstract, would extract more patents.

- **Project Benefits**

This approach can utilize the advantage of less manual efforts and dependency on domain expertise. Moreover, it requires smaller labelled data and improve the data collections. It will reduce the barrier of applying ML/AI in patent semantic search.

- **Targeted Customer**

This can improve the efficiency and user' experience in patent searching. Targeted Customer includes Google Patent Search Group, Patent Search Software Company or other organization heavily related to patent search.

- **Approach**

1. Data Wrangling.

   This step is really fundamental to the whole process. Unlike other data science project, the project retrieves key information from seed analysis and search more data in Google Patent Public dataset. Ultimately, we have three datasets: seed (labelled), anti-seed (labelled), expanded seed(unlabelled)

   Seed: Starting from seed data (classified patents including patent number and title only), we extract abstract through Google Bigquery from Google Patent Database for further analysis. So the seed patents are added to 1285 including "purify water" and "filter water".

   Anti-seed: The anti-seed dataset, opposite to seed dataset, is also manually selected from patents containing "algorithm" without "purify water" or "filter water". These dataset are manually labelled.

   Information Extraction. Three methods: Bigram(tfidf), Doc2Vec and Word2Vec are tried to extract key information. The result from Bigram seems more relevant to the ones from Doc2Vec. From the top 20 features of tf-idf can be used for keywords for expanding data in Bigquery. Another option is that we can leverage pre-trained word2vec model from 5.7M (by Google) to obtain the similar words to "purify", "filter" and "water". These words can be also used as keywords to expand data.

   Seed Expansion. The expanded data can be retrieved based on keywords in abstract. The expanded data are treated as unlabelled data.

2. Data Analysis. Since the key information from seed dataset is a phrase with two words, unigram and bigram are our focus in EDA. And the length and words of abstract are also visualized.

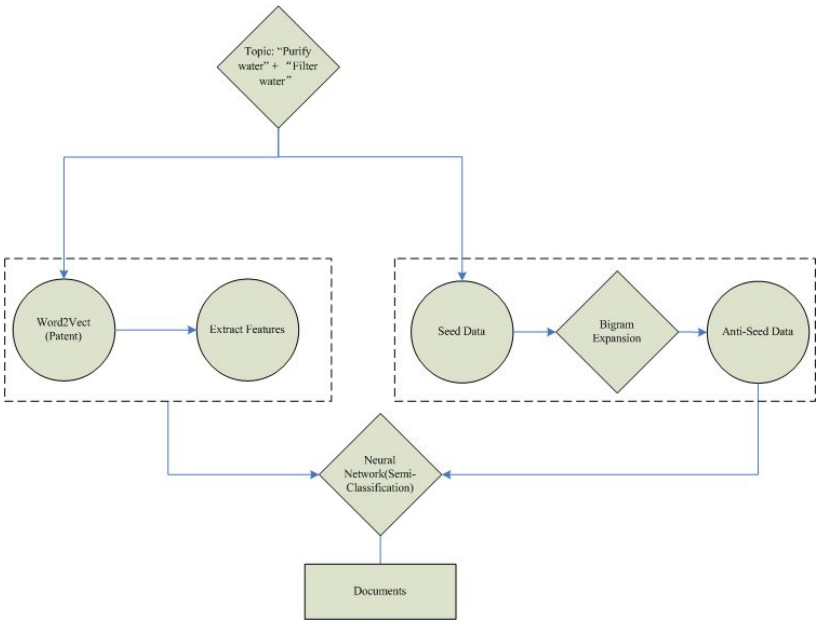
3. Feature Engineering: The input would be texts from patent abstract. So we would convert them to numbers by tfidf (unigram, bigram) or word2vec.

4. Model Algorithm. Supervised and semi-supervised algorithms will be used for this project. Supervised classification algorithm is used for baseline and semi-supervised algorithm is applicable to this problem due to small labelled dataset and large unlableled datasets. The semi-supervised algorithm is designed as two steps: The first step is to build a classifier with labelled data (seed and antiseed). It is very similar to supervised model. We are using labelled data to train a classifier. The second step is to label the expanded patent (unlabelled data) with the classifier. The trained classifier is to predict the unlabelled data.
   In this step, we may start from Naive Bayes as baseline classifier and try other classifiers as Random Forest, XGBoost and Neural Network (LSTM) with/without pre-trained word-embedding model. Some other models can be adopted too such as Pseudo Model.

5. Model Evaluation. Before evaluate our model, we use a heuristic approach of assigning any document contain the keywords relevant to “purify water” or “filter water” as 1 and others as 0. The unlabelled dataset is very unbalanced. So f1 score is the main metric to indicate how accurately our model can label the patents having information. The optimal result is to have lower false positive and lower negative.

6. Hyperparameter Optimization. After the models would be compared, we will select the best model and optimize the model’s hyperparameters. ML model and AI model would be separately optimized by Gridsearch and Hyperas.

The drafted schemas follows as the figure below:



- **Data Source**

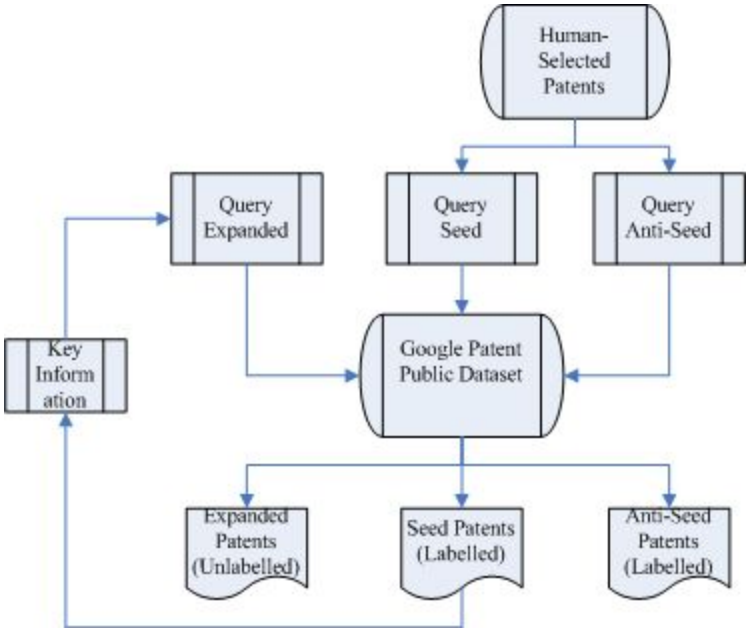
The data mainly comes from Google Public Data. We are using Bigquery to extract data**.** All patents are strictly limited in US patent database. Only abstracts in patents will be analyzed since this project focuses on the meaning of patents,. Here is the data link.

- **Deliverables**

A Final Report, PowperPoint Slides, Source Code at Github. I might also publish it at my medium blog.

Data Wrangling

In this project, data wrangling are repeated to extract data for EDA analysis and modelling. This fundamental step is to expand human-selected patents and results in three kinds of datasets: Seed, Anti-seed and Expanded Patents. The first two are human- labelled and the last one is unlabelled. The steps are shown in the figure below.



Here are details on every step.

- Seed: Fundamental dataset contains “purify water” or “filter water”. At first, we tried 87 patents relevant to “purify water”, which is insufficient to extract key information for next step. Then the seed is increased to 1285 and human-labelled dataset. Through Bigquery, we find abstracts based on patent number and title. When the seed data is larger, it can cover more topics to extract more information. Seed dataset is labelled as “1”.

	publication_number	title	abstract
0	US-6652816-B2	Apparatus for generating ozone and anion	An apparatus for generating ozone and anion ca...
1	US-4655910-A	Liquid filtering device	Liquid filter devices, particularly for irriga...
2	US-3996136-A	Pump-filter for bilge water	A housing for a bilge pump-filter has a lower ...
3	US-6036178-A	Device for mixing air and water in a water pur...	A device for mixing air and water in a water p...
4	US-5813245-A	Pressure relief circuit for refrigerator conta...	A refrigerator having a water filtration and d...

- Anti-seed: This dataset has the opposite polarity to seed. Patents including “algorithm” are searched to increase polarity. Through Bigquery, the keyword “algorithm” in abstract is searched and exclude any containing “purify water” or “filter water” to minimize the overlapping with seed. Google’s work recommends the range of anti-seed number from 10,000 to 40,000. So we select 20,000 for anti-seed datasets and labelled as “0”
- Key Information Extraction: The key information would be extracted from the seed datasets. Three approaches for extraction were tried: Tfidf (bigram), Word2Vec,

Doc2Vec. The results show Tfidf can produce the expected results. So we choose top 10 phrases. More phrases would have broader coverage on results but need more time for searching patents.

```
['water purification',
 'reverse osmosis',
 'waste water',
 'water filter',
 'purified water',
 'filter cartridge',
 'filter water',
 'present invention',
 'water treatment',
 'filter element']
```

We can see that the keywords are expanded from “purify water” or “filter water” to other words such as “water treatment”. Meanwhile, there are also unwanted information such as “present invention”. Since little human-intervention is involved, we still use these keywords to find patents and have machine learning algorithm prune the results.

Pre-trained Word2Vec model for patents trained by Google can match single words only. For example, top 10 of words close to “purify” found from Word2Vec are

```
[{'word': 'purifying'},
 {'word': 'purification'},
 {'word': 'purified'},
 {'word': 'adsorbent'},
 {'word': 'adsorbing'},
 {'word': 'adsorbed'},
 {'word': 'adsorb'},
 {'word': 'purifier'},
 {'word': 'treat'},
 {'word': 'purity'}]
```

Top 10 of words close to “water” are:

```
[{'word': 'soluble'},
 {'word': 'dispersible'},
 {'word': 'hot'},
 {'word': 'sea'},
```

```
{'word': 'tank'},
{'word': 'steam'},
{'word': 'jet'},
{'word': 'cold'},
{'word': 'swimming'},
{'word': 'purified'}]
```

We can combine these similar words by the order of “purify water” as keywords to expand more patents with containing. We also tried Doc2Vec. But it does not provide the ideal results. In this project, tfidf for patent expansion is mainly discussed and used. The technique of Word2Vec for expansion is very similar.

- Seed Expansion:
  With the extracted keywords from the previous step, Bigquery searched more patents. Patents as expanded patents are expanded 1285 to 1176846. Among them, most patents are apparently irrelevant to seed patents, such as the topic containing “present invention”.

Three datasets are prepared for the following steps:

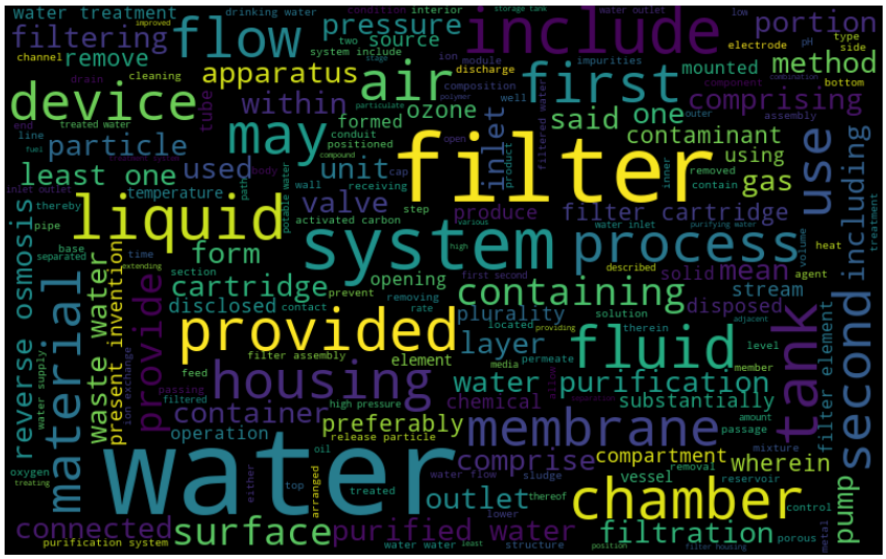
Dataset Name	Dataset Size	Labelled
Seed Patents	1284	Yes
Anti-seed Patents	20,000	Yes
Expanded Patents	1,176,846	No

Data Cleansing

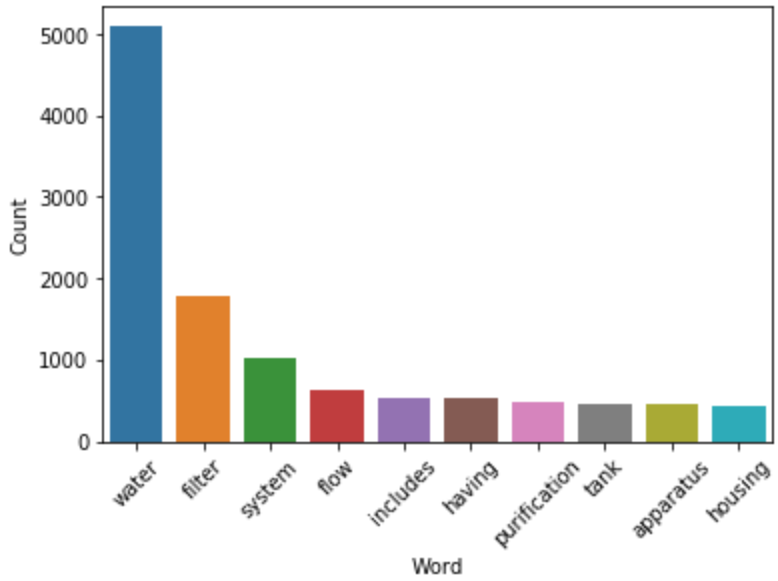
The three datasets we are dealing with are all texts in abstracts of patents and our purpose is to recognize and classify the expanded patents. Because the extraction of three datasets are performed individually, there may be duplicates among those three datasets. Those duplicates are removed. The capital letters are changed to lower cases. Patent often contains non-alphabetic characters. So non-alphabetic characters (number and punctuation) are removed and stop words such as “am”, “is”, “to ” etc. are removed. This applies to all three datasets.

Data Visualization

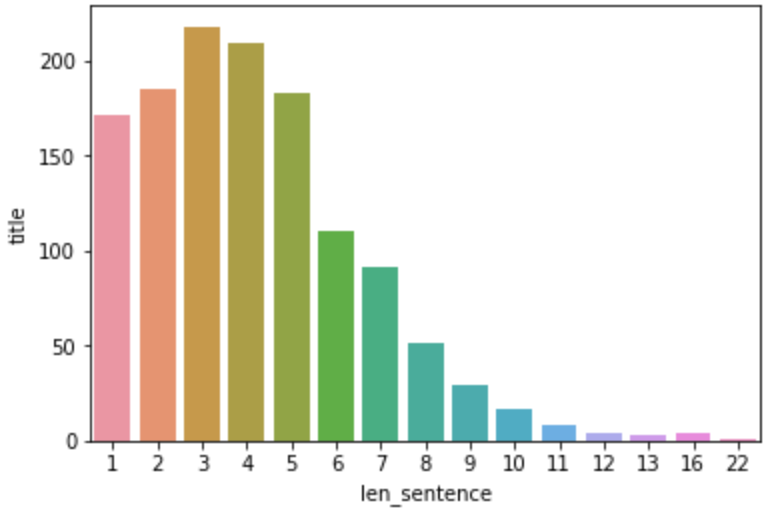
Let's dig deeper about the dataset visualization. We are taking seed patent as an example here. Technique would be similar and you can find complete results in the powerpoint slides in this folder. The below chart indicates the most commonly words in texts in WordCloud.



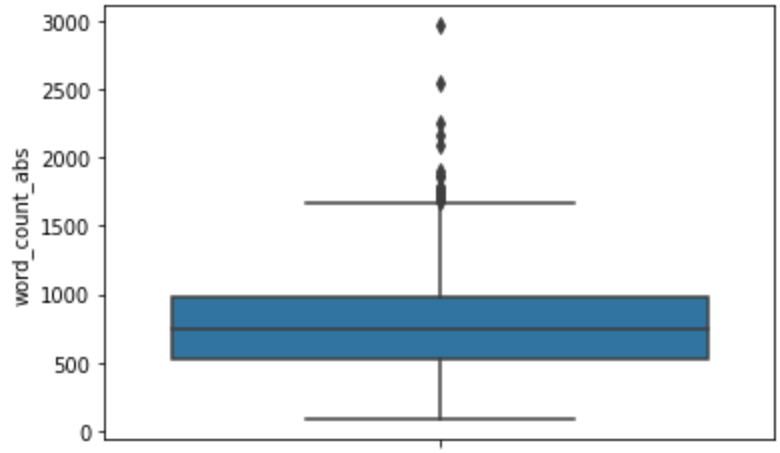
The chart below shows the top 10 words in abstracts. We can see the word "water" is top 1.



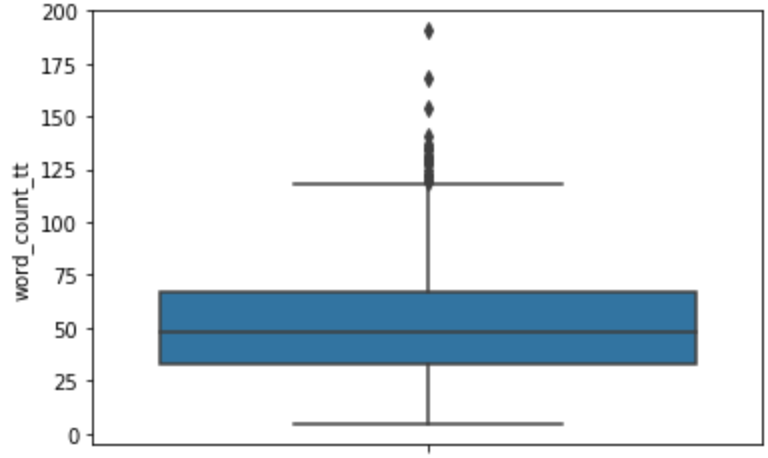
We can see most abstracts would be less than six sentences.



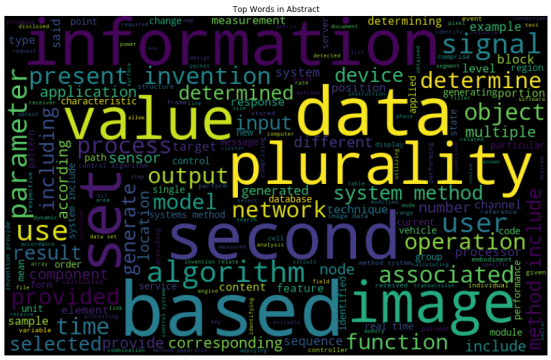
The median of word number in abstracts is around 750.

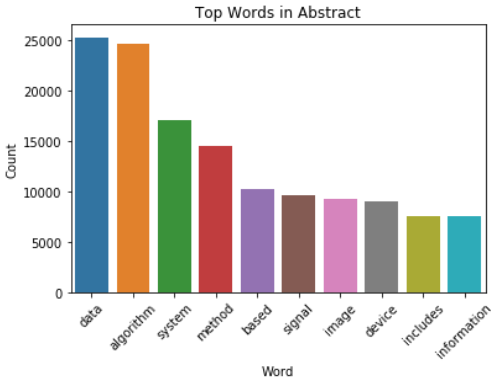


The median of word number in title is around 50.

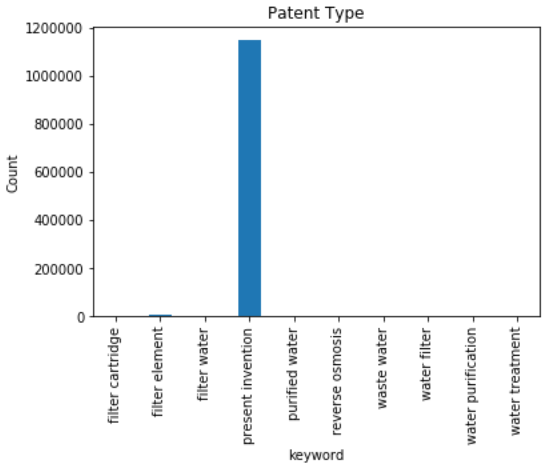


Antiseed focuses mainly on data and algorithm





Expanded dataset is very unbalanced.



Feature Engineering

Since we focus on the contextual meaning in abstracts, we chose three feature engineering approaches: Tfidf with unigram, Tfidf with bigram and word embedding. Prior to data transformation, number is changed to “-NUMBAER_” and punctuations, non-alphabetic character and stopwords are removed.

Noticeably, in word embedding, we do not convert the abstracts directly like many publications do. Most abstracts have less than 1,000 words after text cleansing and size of word vector would be set as 1,000 to include most abstracts. However, the large word vector often has a great impact on the computation speed of model. To improve this, we used a pretrained model trained from 5.9 million patents by Google to represent our datasets. Then some words not included in the pretrained would be ignored and the representation of each abstract length is

reduced. This technique might decrease accuracy slightly, but the size of word vector is decreased dramatically from 1000 to 461. And no top words can be assigned when words are tokenized. The size of labelled can be reduced to 21824 * 461.

Labelled dataset is composed of seed and anti-seed while expanded dataset is seen as unlabelled dataset. We can see an example of word vector:

```
Clean text: apparatus generating ozone anion capable respectively
controlling ozone generator anion generator disclosed apparatus includes
ozone generator ozone generator driving portion second power source
supplying power ozone generator driving portion anion generator anion
generator driving portion circulating fan circulating generated anion
power source supplying power anion generator driving portion selecting
valve selectively discharging anion generated anion generator water
purifier number ozone container activated carbon filter removing smell
ozone introduce selecting valve water purifier controlling portion
controlling anion generator driving portion ozone generator driving
portion operating portion inputting external instruction controlling
portion timer controlling reserving function operating portion
Word vector: [48   307  4335  4225   345   423   357
  4335   514  4225   514    90    48    24  4335   514  4335   514   353
    35    22    72   103  1047    72  4335   514   353    35  4225   514
  4225   514   353    35  3022  1219  3022   329  4225    72   103  1047
    72  4225   514   353    35  1064   127   462  1909  4225   329  4225
   514   125 11431   174  4335   232  1200   334   263   757 16936  4335
  5519  1064   127   125 11431   357    35   357  4225   514   353    35
  4335   514   353    35   278    35  3646   416  1007   357    35  2367
   357 12025   337   278    35]
```

Modelling

- **Algorithm Introduction**

  Two modelling method is introduced in this project: Supervised and Semi-supervised algorithms. The first, such as Naive Bayes, Random Forest, XGBoost, is very common to be executed in many occasions. Semi-supervised here is not often mentioned and we emphasizes this algorithm in this project.

1. Supervised algorithm

   Five classification algorithms applied to this project are Multinomial Naive Bayes, Random Forest, XGBoost, Neural Network(DNN) and Neural Network(LSTM). Multinomial Naive Bayes is built for the baseline model. DNN is for tfidf sparse matrices and LSTM is for word embedding. LSTM also leverages the pretrained model by 5.9M patents to provide weights for embedding layer. These two types of neural networks will be discussed later.

2. Semi-supervised algorithm

   We are using Pseudo Label for semi-supervised algorithm. It contains three steps:

   - ❖ Labelled dataset is fed to train a classifier, then a random small sample (0.3) of unlabelled dataset is predicted as pseudo label dataset by the classifier. The sample rate can not be too large. It is recommended between 0.1 and 0.5.
   - ❖ The pseudo label dataset is concatenated with labelled dataset as new training dataset
   - ❖ The first step is repeated. The difference is that the new training dataset is to train the classifier and the entre unlabelled dataset is predicted.

- **Baseline Model**

  Due to the simplicity and speed, Multinomial Naive Bayes is the baseline model for supervised and semi-supervised approaches respectively. Their ROC Curve is shown as below. We can see Pseudo Label performs better.

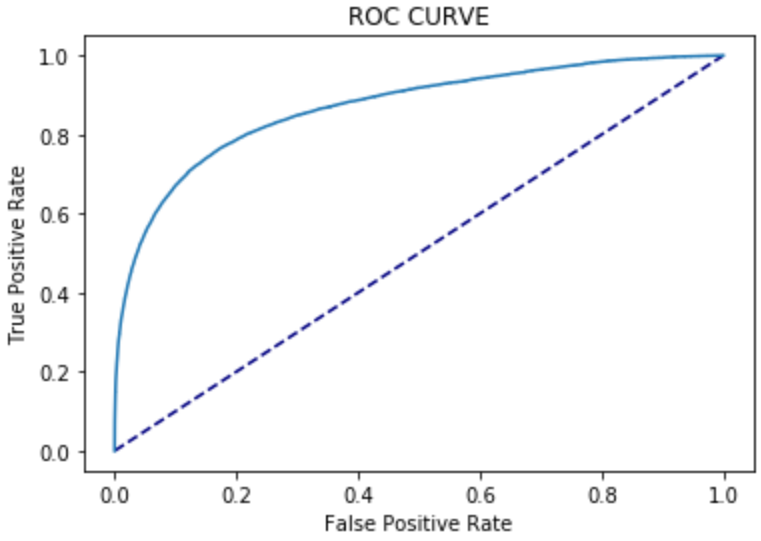


Fig 2. Classification of Multinomial Naive Bayes

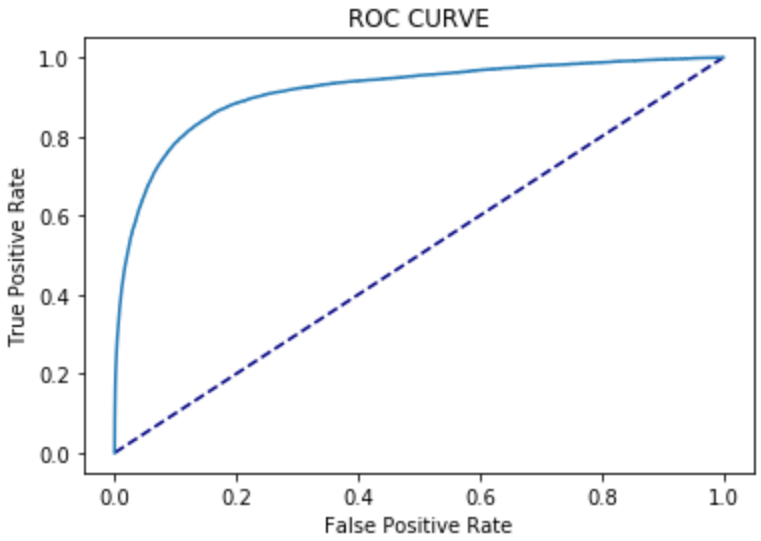


Fig 3. Pseudo Label based on Multinomial Naive Bayes

- **Neural Network**

  As discussed before, two Neural Network are introduced in this project. DNN is fully connected network and takes input as sparse matrices. It uses the size of label dataset as the input layer's size and connects to each layer fully. Sigmoid is placed as activation function in the output layer to predict probability of classification.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 26875)             0
_________________________________________________________________
dense_5 (Dense)              (None, 8192)              220168192
_________________________________________________________________
dense_6 (Dense)              (None, 2056)              16844808
_________________________________________________________________
dense_7 (Dense)              (None, 512)               1053184
_________________________________________________________________
dense_8 (Dense)              (None, 64)                32832
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 65
=================================================================
```

```
Total params: 238,099,081
Trainable params: 238,099,081
Non-trainable params: 0
```

LSTM takes word vector as input. In embedding layer, the weights of pre-trained model from 5.9 million patents. Similarly, the activation function in output layer is sigmoid.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embed_input (InputLayer)     (None, 461)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 461, 300)          33072000
_________________________________________________________________
lstm_1 (LSTM)                (None, 461, 50)           70200
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 50)                2550
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 51
=================================================================
Total params: 33,144,801
Trainable params: 72,801
Non-trainable params: 33,072,000
```

- **Model Evaluation**

  We are borrowing the classification metrics to evaluate the models. So the unlabelled dataset must be labelled manually as true value first. We are using a heuristic way to label unlabelled dataset. The patent having any key information relevant to "filter water" or "purify water" is regarded as 1. Otherwise, the patents not having the information would be labelled as 0.

  Among 1,176,846 expanded patents, 29,277 patents are target patents. Other unwanted 1.1 million dataset costs huge computation power and its imbalance might greatly model

performance, especially for Neural Network. Therefore, unlabelled data are sampled with the ratio of 1:3 between 0 and 1. This ratio can be changed too.

In this example, we are using True Positive Values (How many wanted patents are recognized) and F1 score (low false positive and negative rate). Both with higher values signify the better model performance. The results with various models are shown as below.

Table Performance with Supervised Classification

	TP Values	F1 Score		TP Values	F1 Score
NB(tfidf)	11705	0.5464	**NB(tfidf-bi)**	11705	0.5464
RF(tfidf)	18934	0.6674	**RF(tfidf-bi)**	21042	0.7465
XGB(tfidf)	29244	0.3908	**XGB(tfidf-bi)**	29245	0.3956
DNN(tfidf)	25240	0.6838	**DNN(tfidf-bi)**	26662	0.6198
LSTM(w2v)	12565	0.3016			

Table Performance with Semi-Supervised (Pseudo Label) Algorithm

	TP Values	F1 Score		TP Values	F1 Score
NB(tfidf)	8621	0.4498	**NB(tfidf-bi)**	8732	0.4539
RF(tfidf)	19748	0.7537	**RF(tfidf-bi)**	20582	0.7477
XGB(tfidf)	29236	0.3963	**XGB(tfidf-bi)**	29235	0.3963
DNN(tfidf)	26724	0.6356	**DNN(tfidf-bi)**	24467	0.6541
LSTM(w2v)	12550	0.2988			

From the tables above, the results are very mixed. Generally speaking, Random Forest and Fully-connected Neural Network (DNN) yield better performance. Other models have very huge difference. Neural Network with word2vec has lower performance probably because some key information were removed in tokenizing from pretrained model.

The pseudo label improved Random Forest with tfidf inputs. This might be relevant to the random sample rate. Tfidf for unigram and bigram have no big impact on the performance.

Results

We take DNN as an example and check the result in details. Confusion matrix shows that most wanted and unwanted patents are sorted out. The False Negative is pretty low.

```
[[78265 19304]
 [ 4037 25240]]
```

We can explore more about results. This abstract is identified correctly

'filtration system cleaning fluid wash tank filtration system includes process tank fluid pumped filter element located wash tank filter element formed hollow porous tubes fluid pumped process tank hollow porous tubes process tank fluid hollow porous tubes greater pressure fluid wash tank submerged fluid particulate matter flow pores tubes wash tank cleaned fluid provided wash tank fluid higher concentration contaminants returned process tank'

Example of False Negative:

'system method supporting cache coherency computing environment having multiple processing units unit having associated cache memory system operatively coupled therewith system includes plurality interconnected snoop filter units snoop filter unit corresponding communication respective processing unit snoop filter unit comprising plurality devices receiving asynchronous snoop requests respective memory writing sources computing environment point point interconnect comprising communication links directly connecting memory writing sources corresponding receiving devices plurality parallel operating filter devices coupled correspondence receiving device processing snoop requests received thereat forwarding requests preventing forwarding requests associated processing unit plurality parallel operating filter devices comprises parallel operating sub filter elements simultaneously receiving identical

snoop request implementing different snoop filter algorithms determining snoop requests data determined cached locally associated processing unit preventing forwarding requests processor unit manner number snoop requests forwarded processing unit reduced increasing performance computing environment'

Example of False Positive:

'present invention method producing silicon wafer silicon single crystal comprising double polishing step mirror polishing sides wafer sliced silicon single crystal heat treatment step heat treating mirror polished wafer repolishing step polishing surface sides heat treated wafer provided method producing silicon wafer silicon wafer high quality cop free region oxide precipitate free region sufficiently ensured haze foreign body sticking wafer surface contact trace jig wafer surface produced'

The example of True Negative:

'embodiments present invention recite method system administrating gis data dictionaries embodiment mobile electronic device assigned workgroup method comprises selecting data dictionary comprising desired gis feature type originally intended use mobile electronic device based assigned membership mobile electronic device workgroup method comprises sending data dictionary mobile electronic device wireless communication network'

The model can clearly identified most patents correctly. However, some interdisciplinary patents like data computation in filtering water, might be categorized incorrectly And some other patents not involved in training datasets fail to be sorted out.

Future Work

- Training Dataset: Inclusion of more seed data can be beneficial to the performance. And data can be oversampled to overcome the imbalance for Neural Network model.

- Information Extraction: Besides tfidf, other approaches such as word vector, topic modelling are also worth trying to extract information.
- Information filter: We are using all keywords extracted from labelled dataset, but it might include some irrelevant keywords to topic. An algorithm of pre-filtering those keywords might be added to improve efficiency of patent expansion. So the size of dataset can be effectively diminished and the performance of model can be improved.
- Feature engineering: There are other approaches such as CounterVectorize for feature engineering. Some other features, such as CPC(Cooperative Patent Classification) and Citation Number, might be involved to improve model performance. The pre-trained model for word2vec might be re-trained with more patents
- Model improvement: Other semi-supervised and neural network algorithm (RNN, GAN etc.) would be applied.
- Model evaluation: Right metrics is very fundamental to the success of this project. We choose the number of True Positive and f1 score as key metrics. From the business, It needs to integrate with business metrics to choose the model.
- Hyperparameter Optimization: Sample rate in Pseudo, oversampling rate for unlabelled data and hyperparameters for models need to be optimized after the model is determined.