Text

Description automatically generated

**ISSS609 Project Report**

**Application of Text Analytic Method in Automatic Information Analysis from Refinery News Corpus**

***Group 2-1***

|  |  |
| --- | --- |
| ***DONG Fang*** | ***fang.dong.2020@mitb.smu.edu.sg*** |
| ***LU Di*** | ***dilu.2020@mitb.smu.edu.sg*** |
| ***TANG Yue*** | ***yue.tang.2020@mitb.smu.edu.sg*** |
| ***YIN Xiaolan*** | ***xiaolan.yin.2020@mitb.smu.edu.sg*** |
| ***YOU Pingping*** | ***ppyou.2019@mitb.smu.edu.sg*** |
| ***ZHANG Jiale*** | ***jialezhang.2020@mitb.smu.edu.sg*** |

***17 APR 2021***

**1. Introduction**

## **1.1 Background**

The law of supply and demand primarily affects the oil industry by determining the price of the “black gold.” Expectations about the price of oil are the major determining factor in how companies in the industry allocate their resources. Prices create incentives that influence behavior. This behavior eventually feeds back into supply and demand to determine the price of oil. Oil trading is the buying and selling of diverse types of oil and oil-linked assets with the aim of making a profit. As oil is a finite resource, its price can see massive fluctuations due to supply and demand changes. The oil supply comes from refineries. The total supply in a specific period can be calculated by sum of capacity of refineries worldwide. But refineries may under planed or unplanned suspend production which makes a profound influence on supply. The common reasons are maintenance, outage, fire, strike, economic reasons etc.

## **1.2 Motivation and objective**

The data comes from Ampol Singapore’s subscribed industry news. There are a lot of refinery news happening on a single day. To predict the supply, we need to read many refinery news every day. The manual process is firstly read a news and judge whether it is talking about something will decrease the oil production, such as maintenance or halt production. The second process is extracting relevant information from the news. If the news is talking about maintenance, outage etc. We need to record which refinery it is, what is the period, which unit(s) it affected, what is the reason and how many capacities it affected. Therefore, the motivation behind this is how can we turn this manual process into an automated process. In terms of the objective of this project, the first part is to build a LDA model to analyze the potential topics in refinery news; the second part is to create a document retrieval system to help users locate or extract target news; the third part is to build binary machine learning classification models to distinguish news between positive class and negative class; and the final part is to apply the techniques to extract information and then apply that on the supply prediction.

**2.** **Methodology**

In Document retrieval part, we use TF to select the most relevant document given by the keywords.

For Topic modeling, we use LDA to implement topic modelling, following by the analysis of evaluation of the models, topic distribution, label ratio and top n keywords of each topic, and the word cloud. And the challenge my within the choose of which part of dataset for topic modeling, also the model or package to be used for LDA.

In addition, we explore on 3 well established text classification models, namely Naive Bayer Classifier, Support Vector Machines (SVM), and Random Forest with Scikit-learn library. We use unigram and TF-IDF in the document's representation. The corpus includes the combination of news titles, news content and one of news tags - “units affected”, where “units affected” lists out the refinery units if there are any. Lastly, we would use cross-validation and grid-search to fine tune the model to achieve the highest F-beta score.

Moreover, in the part of information extraction, depends on the specific information we needed to extract, the methods are different. Some information can be directly extract using tokenization, string operation and slicing, so we defined several functions then apply those functions on the whole news. For the information cannot be extracted by regular expression, it would be very challenging to use regular expression to obtain the information. To deal with this problem, we finally trained a pre-trained model to learn to recognize our self-defined entity automatically with spaCy .

**3. Solution Details**

## 3.1 Document retrieval

As for the detailed method to do the get the TF for keywords in documents, we defined a function to get keywords frequency in each document given the keywords users may concern about. Specifically, the inputted keywords should be stemmed by applying “stemmer” function in “nltk” before finding the frequency. The advantage of using TF is easy to calculate and very straightforward.

## 3.2 Document classification

Based on the content of every refinery news, we firstly labeled the news as 1 which would influence the oil production and price and labeled the news that has no impact as 0, totally 1322 news were labeled in processed news dataset. Given the importance of “units\_affected” in news which provide the concentrated refinery information about which units will be affected by this news, we extract “unites\_affected” information and added it as a new column in new\_for\_classification.csv. For the news without “units\_affected” information, “unites\_affected” was assigned to 0.

As important features, column “title”, “notes”, “units\_affected” were processed for document classification. Firstly, we expand the contractions and removed special characters in documents. Next, we use spaCy library breaks documents into tokens and remove default stop words from sentences, since “no” and “not” have been removed from stop word list, we also remove them from the input sentence after stop word removal. In the following step, we use spaCy to parse test and assign part -of-speech tag, further lemmatize documents.

Next, we convert text documents into corresponding features by TF-IDF values, Term Frequency - Inverse Document Frequency, which summarizes how often a given word appears within a document. We use TfidVectorizer in Sklearn to calculate TF-IDF vector for each of refinery news. Because Naive Bayes model are a group of extremely fast and simple classification algorithms for training and prediction, especially Multinomial Naïve Bayes performs well in text classification, in our project we start with Multinomial Naïve Bayes model to classify refinery news.

Moreover, because Support Vector Machines (SVM) model works well for highly dimensional space, for example, in text classification. And this algorithm also has high accuracy and is less prone to overfitting due to the presence of a regularization parameter. In our project we use SVM to exploit news classification.

Besides, to get a better prediction outcome, we tried to build third classifier. Because Random Forest Model is considered as highly accuracy and robust method for classification and regression problems. To enhance the performance of classification, we further use SVM and Random Forest Model in scikit-learn library to classify news.

## 3.3 Topic modeling

As the data set for topic modeling, there are two choices, first is the contents of the news and second is the title of the news. We decided to choose the title to perform analyzing, since generally the title is the summary of the news and could highly extract the main topic of it. And the title contains less useless words which may interrupt the results of topic modeling. We tried both contents and titles as the source data, and the perplexity of contents are higher then title, which proved that the title is more suitable for analyzing

There are two popular models used for doing LDA, one is Gensim and the other is sklearn. As sklearn offers a very systematic, efficient framework for analyzing, ensemble methods, evaluation and validation, and hyper-parameter optimization. Thus, we finally use ‘LatentDirichletAllocation’ from sklearn to implement topic modelling. And the important features include the iteration times, which is set to be 10, and the topic number, which is set to be 6.

## 3.4 Information extraction

In this part we mainly use spaCy package to lemmatize contents and train our models. While generally nltk provides access to many algorithms to get lemmatization done, spaCy provides the best way to do it. It provides the fastest and most accurate syntactic analysis of any NLP library released to date. It also offers access to larger word vectors that are easier to customize. So instead of using nltk in this part, we use spaCy.

To get the entity we defined by ourselves, we use transfer learning to predict our user defined NER using spaCy, the pre-trained model we tried is “en\_core\_web\_sm” . Since fine tune the pretrained model needs a lot of annotated training set, but we have limited manpower, we only manually annotated a small part of training set to test if this solution works for our project. The final result shows that after 10 times iteration, the model can predict our user defined NER.

**4. Experiments**

## 4.1 Data pre-processing

We used Jupyter Notebook to implement our experiments. The original dataset we have is a single txt file named ‘refinery’ which contains all news. And what we want is a corpus of news where every txt contains a piece of news, or a csv file that each row represents a piece of news. We first transformed it to a corpus and then used the corpus to form a data frame and export to csv file.

In order to get the corpus, we tried to find some features of the news and apply different processing methods accordingly. By reading every line from the txt file and checking whether it meet some different requirements we can let the machine recognize the features. As the figure shows, we mainly did the four kind of process.

|  |  |  |
| --- | --- | --- |
| Features | Representations | Process |
| ‘100-REFINERY’ | The start of a piece of news. | Recognized it, opened a new txt file (eg. ‘news1.txt’) to store this line and for the following lines do not start with this pattern, appended them to the file that store the start line of the news. |
| Finish by ‘…’, ‘0100’ or ‘pltn’ | Not important. | not included those lines in the output txt files in the corpus. |
| Start with ‘--’ or 'source: ' (except the space) |
| End with ‘GMT’ | Date values. | involved them in the output txt file for further extraction of the date values and add a dot in the end. |
| start with items including a word and a colon | Different part of the content. | Recognized them, replaced all the dot in the content and added a dot at the end for further exploration. If the item name is ‘note’, we also added a dot before it starts. |

*Table 1 Pre-processing: steps of transform to a corpus from a txt file*

|  |
| --- |
|  |

*Fig. 1 Pre-processing: steps of transform to a corpus from a txt file*

Now we come to the construction of the csv file. In the csv file we need the news number that was allocated to each news in the previous txt processing part, the date extracted from the lines contain ‘GMT’, the title in the first line of the single news txt file and the notes, which is the main part of the news.

We traversed all the txt files and each txt file extract one row. The news number was the name of the file. The date values were extracted using regular expression. The main parts of the news were those behind ‘notes:’. We only extracted the first three sentences since they were enough to determine the theme of the news. For those news less than three sentences, we selected all of them.

One more thing we need to do is label the news manually. Although all the news is refinery news, from the business perspective, we are not care about all of them. In terms of the topics, some news is talking about the refinery will undergo maintenance; some refineries are shut down or halted; some refineries are upgrading or expansion; some refineries has restarted from works; some refineries report fire, glitch, or influenced by natural disaster; some refineries have strike. The topics can also be report production or export and etc. In our case, we are only care about those news reports something will have a particular influence on the supply, those are the news labelled as 1, otherwise, the label will be 0. We use cross labelling, which means each news has at least two people to label it, in order to make sure the quality of our labelling work.

The final csv files are shown in the following figure.

|  |
| --- |
|  |

*Fig. 2 Pre-processing: content of result csv file*

## 4.2 Document Retrieval

Given the preprocessed and structured news dataset saved as a “.csv” file, first we tokenized the words by using “nltk” package, preserved the English words with applying Regular Expression, removed stop words, then stemmed the words. And the stemmed word list was reserved in the final data frame for further analyzing. With the stemmed words list, we used “Counter” function to count the frequency of each word for each news line, and the counting results were added as a new column named “count” in the data frame.

As the second step, we defined a function to get TF for stemmed keywords in each document. The frequency for each keyword would be saved as a new column, besides a new column named “Total” will be added to record the sum of all keywords’ counts.

Finally, to display the results of document retrieval, we defined a “summary” function to get the document numbers having the top n frequency of each keyword as well as the total frequency of all keywords. In addition, we defined a “printed\_out” function to print the detailed information of the documents retrieved from “summary” function, including word counts, news number, news date, news title, news label and news notes.

## 4.3 Document Classification

4.3.1 Further data pre-processing for classification

The dataset has well balance on size of positive and negative class, almost of 50% of each. We ran some preliminary models (Naïve Bayes and SVM) on tokenized news body content alone, however, the accuracy scores were not satisfactory. This led us to include “units affected” and news titles into corpus. We went through a series of data preprocessing on corpus, including expanding contractions, lowercase the text, removing non-importance features (e.g., extra newlines, special characters, and stop words). We then introduced Part-of-Sentence (POS) feature to perform lemmatization. After the data preprocessing, 33% of tokenized news would be set aside for model testing, and cross validation (3-fold) would be used to train model’s hyper-parameters.

4.3.2 Assessment of classifiers’ quality

We propose F-beta to measure model performance. F-beta is a harmonized measurement on precision score and recall score. The beta is set as 1.5, which is to favor recall score. For our model, we emphasize recall score (false negative) more than precision score (false positive). False negative refers to the situation where the target news fails to be detected by the model. This would cause inaccurate data fed into the pricing model subsequently. False positive refers to the situation where the random news is picked by the model, which would be translated to more working hours of the data specialist team to review the news. Therefore, we tend to reward recall score more than precision score.

## 4.4 Topic Modeling

First, we got the title list from the csv file and do the lemmatization. Besides the default stop words list, we created a new list which contains two parts and remove the words in this list. First part consists of the words of the countries, organizations and date recognized by NER using spaCy. Second part consists of the words appeared in the previous results of topic modelling but have no real meaning or are not helpful in analysis the topic (for example, ‘refinery’, ‘unit’, ‘star’, ‘complex’.).

We used the ‘LatentDirichletAllocation’ from sklearn to implement topic modelling. The iteration is 10 by default, we did a line chart to decide the most appropriate topic number and learning decay (0.9). However, the best model only contains 2 topics, which is not what we want. Finally, we selected 6 as topic number, which is based on the consulting result of expert with domain knowledge. And the perplexity of the model is 80.32. pyLDAvis is used to visualize the result.

## 4.5 Information Extraction

Given the preprocessed news dataset and news\_for\_classification.csv file, we first get the news are labelled as 1 which are our target news for information extraction. Then we created news\_for\_IE.csv which contains news number, title, date and contend information, and used spaCy to lemmatize the content.

We defined some functions to do information extraction. The functions use tokenization, string operation and slicing. Information like refinery name, owner name, overall capacity, affected unit(s), affected capacity and duration can be extracted directly from our pre-defined functions.

When we extract reason, we use transfer learning. The package used remains spaCy. First, we prepared the train data. The format of training data is a list, each item in this list is a tuple. The tuple contains two parts, the first part is a string, the second part is a dictionary. The key is “entities” and the value is a tuple, the first element is the start index of the entity in this string, the second element is the end index of the entity in this string, the third element is user defined entity name. Refer below to see an example.

|  |
| --- |
|  |

*Fig. 3 An example of the training data for transfer learning*

Then we load the pre-trained model “en\_core\_web\_sm” which is a CNN model. In spaCy, the available pre-trained models are “en\_core\_web\_sm”,“en\_core\_web\_md”,“en\_core\_web\_lg” and” en\_core\_web\_trf”. “sm” means this model is trained on a relatively small corpus (contains blogs, news and comments), “md” means the size of corpus are medium, “lg” means the size of corpus are large and the final one is the largest. The larger the model, the better the evaluation metric. In our case, we only tried the smallest pre-trained model.

After load the pre-trained model, then we append our user defined entity name to the original NER list. The original NER support cardinal, date, event, GPE, language, law, money, ordinal, percent, person, product, quantity, time etc. Then we disable pipeline components we don’t want to change, import packages random, minibatch and compounding from spacy.util, path from pathlib and Example from spacy.training. We shuffle the train\_data, this will ensure the model does not generalize based on the order of the examples. We set the iteration for 10 because the model need loop over the example for sufficient number of iterations, too few iterations may not be effective.

Finally, after train the model, the model can recognize our own defined NER. We save all the information extraction results to the news\_for\_IE.csv.

**5. Results and Analyses**

## 5.1 Document retrieval

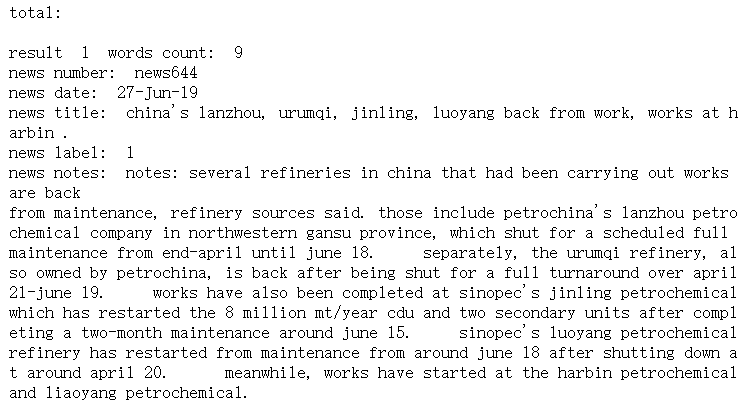
Since our objective is to analyze the influence from refinery events to the oil capacity in the oil market, we intended to retrieve the relevant news documents given the keyword of events such as “maintenance”, “shut” and “restart”. And we set the number of documents as 5. Below table shows the result of document retrieval:

| Keyword | News number | Keyword Counts | News Date | News Title | News Label |
| --- | --- | --- | --- | --- | --- |
| maintenance | news79 | 5 | 23-Jan-19 | China's Daqing Refining to shut for maintenance over Aug-Sep | 1 |
| news200 | 5 | 19-Feb-19 | China's CNOOC Huizhou shuts phase 1 refinery for maintenance | 1 |
| news720 | 5 | 18-Jul-19 | PetroChina's Dushanzi to shut for maintenance July 20 | 1 |
| news85 | 4 | 23-Jan-19 | Baytown's largest maintenance program underway; will add light, sweet crude capacity: ExxonMobil | 1 |
| news207 | 4 | 21-Feb-19 | Italy's Augusta to start two-month maintenance on Feb 21 - source | 1 |
| shut | news483 | 8 | 7-May-19 | Sinopec Hainan shuts residual oil hydrotreater for maintenance | 1 |
| news733 | 8 | 23-Jul-19 | China's Jinzhou back; works at Qingdao, Harbin, Dushanzi | 1 |
| news301 | 7 | 18-Mar-19 | Japan's JXTG shuts Chiba CDU, 13% of capacity for maintenance | 1 |
| news230 | 6 | 26-Feb-19 | ExxonMobil Singapore shuts units for planned work | 1 |
| news267 | 5 | 7-Mar-19 | China's Sinopec Qilu shuts CDUs, residual hydrotreater in Mar | 1 |
| restart | news332 | 7 | 26-Mar-19 | Bilbao restarts FCC | 1 |
| news663 | 7 | 2-Jul-19 | Swiss Cressier restarting after maintenance | 1 |
| news612 | 6 | 13-Jun-19 | Final units at Italy's Augusta restarting after major maintenance: source | 1 |
| news199 | 5 | 19-Feb-19 | Italy's Sarroch restarts various units: source | 1 |
| news225 | 5 | 25-Feb-19 | Bilbao restarting conversion units | 1 |
| Total Keyword Counts | news644 | 9 | 27-Jun-19 | China's Lanzhou, Urumqi, Jinling, Luoyang back from work, works at Harbin | 1 |
| news483 | 8 | 7-May-19 | Sinopec Hainan shuts residual oil hydrotreater for maintenance | 1 |
| news720 | 8 | 18-Jul-19 | PetroChina's Dushanzi to shut for maintenance July 20 | 1 |
| news733 | 8 | 23-Jul-19 | China's Jinzhou back; works at Qingdao, Harbin, Dushanzi | 1 |
| news200 | 7 | 19-Feb-19 | China's CNOOC Huizhou shuts phase 1 refinery for maintenance | 1 |

*Table 2 Retrieved documents given the keywords “maintenance”, “shut” and “restart”*

As we can see from the retrieval results, the highest frequency of “maintenance”, “shut” and “restart” are 5, 8 and 7, found in 3 news, 2 news and 2 news, respectively. And the highest total keyword count is 9 in news644, “China's Lanzhou, Urumqi, Jinling, Luoyang back from work, works at Harbin”. Besides, all retrieved document labels are 1, meaning all of them are targeted documents, which could be used to check if the keywords are effective enough to obtain documents related to our purposes.

Take news644 as instance, we can obtain the result from “printed\_out” function as below figure shows. We can see, in this document, “maintenance” appears 4 times, shut appears 3 times, and “restart” appears 2 times.



*Fig.* 4 *The printed result of document retrieval: the document having the highest total keyword count*

In this way, users could quickly get the documents they want from hundreds of thousands of news texts, as well as directly read the news contents to collect extra details.

## 5.2 Document classification

Below are the results of three modes, where SVM has the highest F-beta score. We would further drill down to the model results and check how well it classifies the news. For text mining, there are many research works prove the good performance of SVM with linear kernel classifier. We therefore did not explore other kernels. As for parameter C, it is to regularize the complexity of model. We derived 0.88 as the optimal value for parameter C from 3-fold cross validation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Parameters | Precision score | Recall score | F-beta score  (beta = 1.5) |
| Naive Bayes | None | 0.79 | 0.95 | 0.84 |
| SVM | C= 0.4, Kernel = 'linear', degree = 3 | 0.88 | 0.93 | 0.9 |
| Random Forest | min\_samples\_split = 5, n\_estimators= 60 | 0.83 | 0.92 | 0.86 |

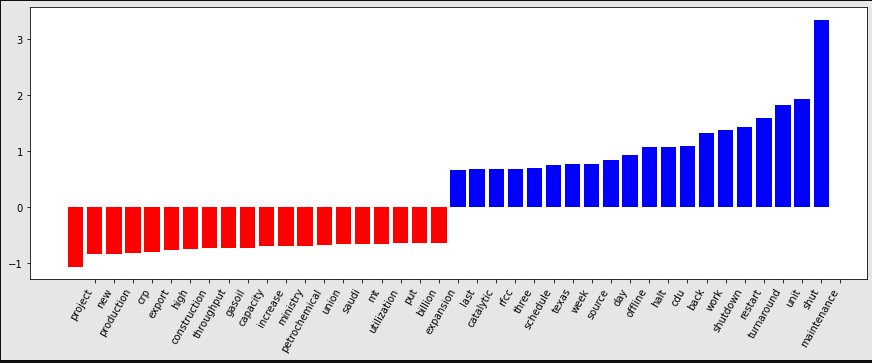
*Table 3 The Model performance of Naïve Bayes, SVM and Random Forest*

As observed in confusion matrix of SVM, out of 210 news, models failed to pick up 15 positive news and misclassified 30 negative news.

|  |
| --- |
|  |

*Fig. 5 The model performance of SVM*

SVM has one attribute “.coef\_”, which returns the weights of features (words) in the model. With this help of this attribute, we generated the top 20 influential words from each class, where blue colored bars represent the positive news and red colored bards are for negative news. We can see that, “maintenance” is the most prominent feature, followed by “shut”, “unit”, “tournaround”, “shutdown”, “work”, and “back”. Apart from words (“unit”, “back”), all the words mentioned are quite intuitive and related to the nature of positive news. On the other hand, there are not many words standing out in negati. Apart from word “project”, which has a slightly higher weight, the rest of words share the similar weight. This explains why false negative rate is higher. More negative keys need to appear together to out weight the positive words.



*Fig. 6 The top 20 most influential words for each class*

We inspected the negative false news, trying to investigate why the misclassification occurred.

|  |
| --- |
|  |

*Fig. 7 The misclassified text file index– false positive and false negative*

Below are sample examples on news 96, 202, 274 and 1242. We can further summarize the false negative error as 3 categories, rare words, ambiguous words, and mislabeled text. Firstly, by applying human judgement, words “rehabilitation” and “blast” would be hint for us to label them as positive news. The probable reason the model failed to pick up words is due to their low occurrence. “Rehabilitation” is used 9 times, and “blast” is used twice only in the corpus. Secondly, word “strike” appears in both positive and negative training data. If a strike causes a negative impact on oil production, the news would be labeled as positive, otherwise the reports on strikes only would be labeled as negative. As shown on the above picture on 20 influential words of SVM model, “strike” related word - “union” has high weightage contributing towards the negative class. Lastly, for news 1242, it was mislabeled as positive news. To a certain extent that the wrong labels cause the noise to model and affect its performance.

|  |
| --- |
|  |

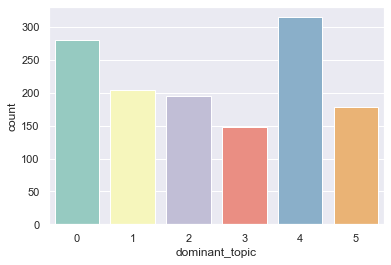
*Fig. 7 The excerpts of news*

## 5.3 Topic modeling

After topic modeling, we generated the word cloud of the high-frequency words in the six topics, and the document counts for each topic, as below figure shows:



*Fig. 8 Word cloud of the six topics*



*Fig. 9 Document counts in each topic*

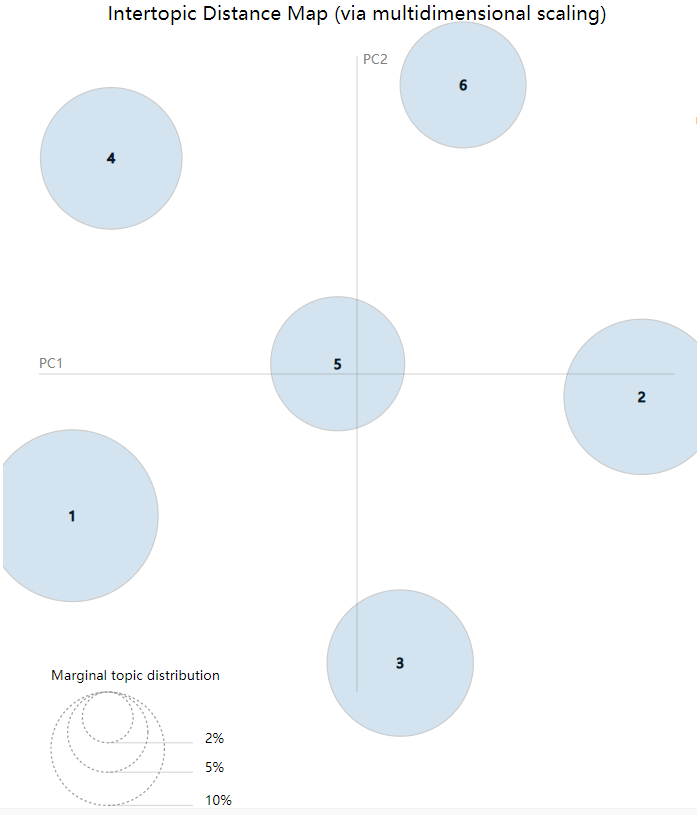
According to above figures, we could get results shown as below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Domain Topic | Most Frequent 5 Words | Main Content | Document Count | Target Label Ratio |
| Topic\_0 | restart, work, crude, carry, set | The refinery plant restarts to work from halting | 281 | 49.5% |
| Topic\_1 | upgrade, run, fuel, production, rate | The upgrade and extension project of production rate in fuel refinery complete or delay | 204 | 28.4% |
| Topic\_2 | plant, capacity, operation, offline, processing | The refining operation capacity of refinery plant | 195 | 40.5% |
| Topic\_3 | shut, flare, maintenance, power, shipment | Oil refinery shut down because of issues such as maintenance or flare | 148 | 73.0% |
| Topic\_4 | Maintenance, plan, start, work, turnaround | The maintenance of refinery plant starts or completes | 315 | 82.5% |
| Topic\_5 | export, report, product, oil, new | Export of energy production such as oil or gasoil | 178 | 23.6% |

*Table 4 Result summary of topic modeling*

Within above topics, topic 3 and topic 4 have the highest target label ratio, which represents the percentage of target documents within the total number of documents in the topic, meaning these two topics are highly probable our targets for analyzing refinery news. Besides, from the main contents summarized by high-frequency words, it is reasonable that topic 3, indicating shut down of refinery plant, and topic 4, indicating the maintenance of refinery plant, should be our target topics, since these kinds of event will lead to the halt of producing oil and further influence the capacity of oil production in the whole market.

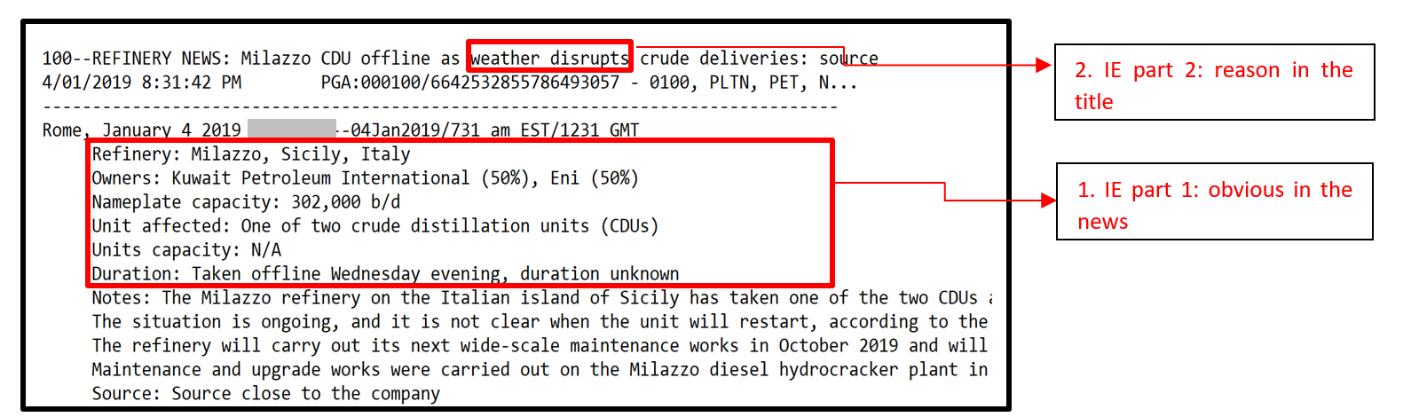
Besides, we generated an inter-topic distance map for the six topics, as below figure displays, it’s obvious that the six topics have no overlapping between each other and distribute evenly, which support the effectiveness and reasonability of the partition of our topics.



*Fig. 10 Inter-topic Distance Map (via Multidimensional Scaling)*

## 5.4 Information extraction

Because of the unique structure of our raw data, the information we want to extract like refinery name, capacity, unit, duration etc. are already obvious in the news. Another information we want to extract is reason cannot be extracted directly. The reason can appear in title or notes. Refer below to see detail.



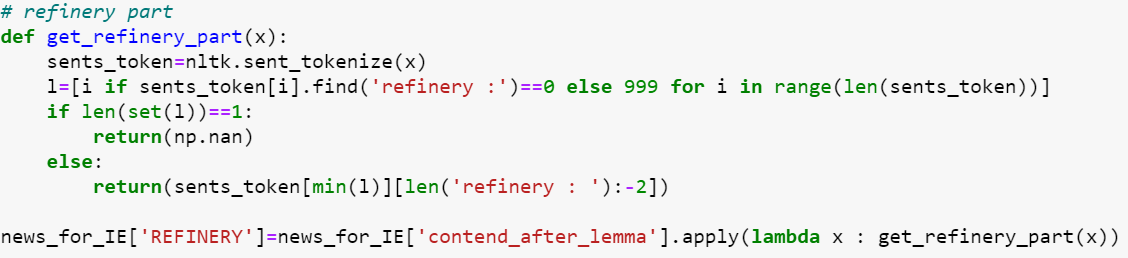
*Fig. 11 Example of information extraction entities[[1]](#footnote-2)*

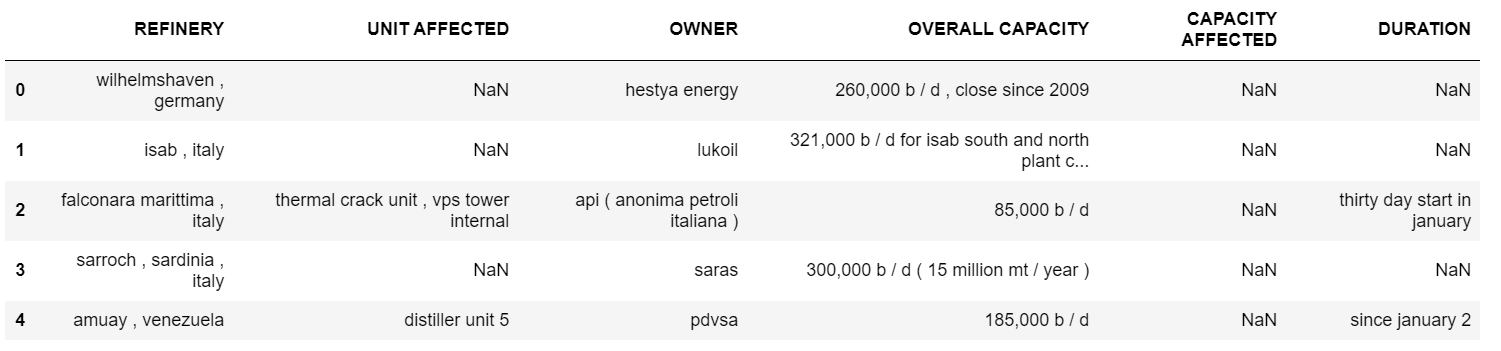
For the information already obvious in the news, we did some tricks in the data pre-processing part. First, we defined the item list to capture those key words, then we add a ‘ . ’ at the end of each key word row. For example, “Refinery: Milazzo, Sicily, Italy” will become “Refinery: Milazzo, Sicily, Italy . ”. The reason behind is that when we use nltk to tokenize the contend, we want each of them become a sentence.



*Fig. 12 Tricks for the purpose of tokenization*

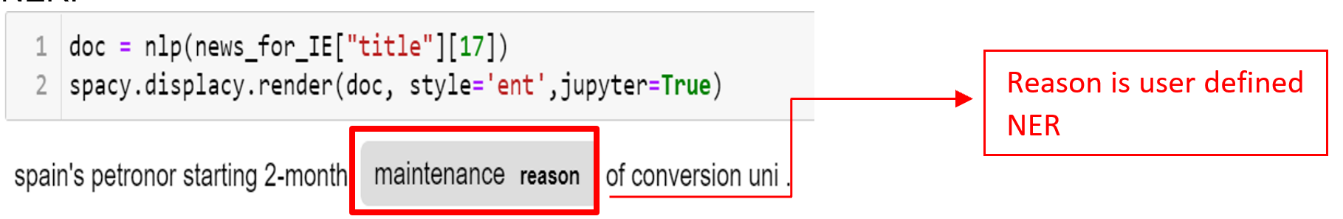
Then we use spaCy to do lemmatization, and use tokenization, string operation and slicing to extract that information. Refer below to see an example of code and results.





*Fig. 13 Demonstration of the code and results of standardized information extraction*

The results are good, this solution can extract almost all of the target information the news have. To extract reason, after training the model, the model can recognize our user defined NER.



*Fig. 14 Demonstration of the code and results of unstandardized information extraction*

In reality, our work on fine tune the pre-trained model is not enough, because fine tune a pre-trained model to recognize user defined NER needs a lot of annotated examples for the model to learn. The annotated examples need a lot of manually effort to annotate the user defined NER and their start index, end index in a sentence. Moreover, the reasons are diversified, they appear in many kinds of structure and context. Therefore, the model needs even more examples to learn.

**6. Discussions and Gap Analysis**

According to the performance, the algorithm of document retrieval works well for single keyword, and the word form will not influence the retrieval since we have stemmed both the documents and keywords. However, one limitation of our algorithm is when we set a phrase as keyword, it will not work. Since all documents had been tokenized and stemmed as the list of single words, it is impossible to find a phrase in the bag-of-word resource.

For Document Classification part, SVM classifier failed to pick up the rare keywords, for example word “rehabilitation” and “blast”. This should be able to be resolved if we acquire more news to the training dataset. Besides, we can consider word2Vec model and Skip Gram approach. In Mikolov et al. (2013b), the authors mentioned Skip Gram works well with small amount of data and is found to represent rare words well. Skip-gram is one of the unsupervised learning techniques used to find the most related words for a given word. In the stage of model evaluation, we discovered a small amount of news texts that were labelled wrongly when we tried to review those misclassified news texts. This human error can be corrected and avoided in the future when company has rigorous data quality control process to look after this area.

The topic modeling works well for recognizing and partitioning the news into six different topics. From the view of Inter-topic Distance Map, there is no overlapping between topics. Also, from the view of human evaluation, which is based on what we know about the dataset, the topic results are highly consistent with our pre-judgment on the topics for the news corpus. However, we do not have an accuracy and clear standard to decide exactly how may topics is the best choice, as the perplexity of 6 topics is not the lowest compared to other number of topics.

As mentioned in information extraction method, we have trained one NER model based on models in spaCy library to autodetect new entities we want from the news. The result shows it can recognize our target entities successfully after we use a small set of examples to train the model. Besides, the trained model can be used in wider or more professional domains since we can define our own entities according to the specific data and objectives. However, there is no external source for the training data just because of the specificity. Thus, we can only manually generate and sort it to the acceptable form of training data, which is a list of tuples that each of it should contain the text and a dictionary holding the start and end indices, as well as the label, of the named entity in the text. Because of time limitation, it is very difficult to gain enough amount of training data, so the model now can only successfully recognize part of entities we set in the training data.

**7. Future Work and Conclusion**

In terms of the objective of this project, we have already gotten a good result. There is not any other point in topic modelling and information retrieve. Actually, our classification models behave good too, to some extent, human error on labelling cannot be avoid. By error analysis, we know what kind of news our models have difficulty in predicting, so we can increase the sample weight or increase that kind of training data in our model. By the way, there are also many other text classification methods, and the representation is different too. For example, instead of using TF or TF-IDF represent our data, word embedding is another good way to represent text data. Our SVM model failed to recognize rare words like “rehabilitation” and “blast”, in reality, the meaning of “rehabilitation” is close to restart which is an important key word in our target news, and “blast” is close to fire or incidence. Traditional machine learning model cannot capture the meaning those words, so word embedding is useful in this case since after word embedding, the similarity between “rehabilitation” and restart will be high. Therefore, we can consider use word embedding as representation in the future. Besides, since our news are not long, Fasttext is a good method too, Fasttext is a CNN model created by Facebook, this model has low model complexity and high efficiency.

In terms of information extraction, there are a lot of work can do in the future. In our project, we failed to prepare enough annotated training data, so in the future, the first thing is that we can try if there is any other good solution for us to generate enough training data; another thing is nowadays the most powerful pre-trained models come from BERT create by Google, we can consider make use of it to do information extraction.

**8. Project Experiences**

Dong Fang:

In this project there are 4 tasks engaged, including Document Retrieval, Document Classification, Topic Modeling, and Information Extraction. When completing these tasks, we not only went through the relevant topic learning material obtained from class and understood the knowledge more deeply, but also browsed some external resources to learn more methods and models in order to achieve our objective. As for me myself, I mainly focus on the analyzing part for Document Retrieval and Topic Modeling. Besides, I have engaged in searching external resources and learning spaCy to resolve the problems encountered in Information Extraction tasks. Also, I have arranged the progress and timeline for the whole project process. As a team, we have experienced a highly efficient and comfortable cooperation. With reasonable job allocation, we have all played our best as we are good at different aspects. And what I like in this project is that our achievement is meaningful for business world, and I believe the solution or models we came up with could really help the company understand the industry better and reduce the cost of getting information they need.

You Pingping:

This is my first text analytics project. It is a fun journey to work with my team to churn out some useful insights from the text dataset. During the project, I was responsible for the text classification task. With the machine learning techniques, I learned from another course and the concepts and text unique dataset structure learnt in this course, I was lucky to build my very first text classifier with quite high measurement score. Surprisingly, I enjoyed the process of data preprocessing. It is fascinating to see the text transformed into numbers and fed into the machine models. For my least favorite part in this project, I think it should go to the data labelling task. As I do not have industry knowledge to decide which texts should be labeled as positive, I had to check with dataset owner, Jiale, several times when I started my labelling task. Eventually I managed to label 260 new texts.

Lu Di ：

The text of our Project comes from energy news that is updated daily. Energy trading companies need to judge the fluctuation of energy prices based on this news to specify the correct trading strategy. This type of news can be divided into two categories. The first type is news that will reduce crude oil production, such as an explosion or fire in an oil refinery, and the other is news that will not affect crude oil production. I am responsible for label part of these news manually and do the pre-process in the dataset in pandas to extract useful information from raw text and add into analyzed data frame. Although I have not learned machine learning before, but luckily my teammate has learned machine learning. So, they build the classification model and I do the pre-possess to extract more specific data from raw dataset to the model to improve its accuracy. I learned a lot in this project, such as data prepossess in pandas and machine learning commonly used packages such as sklearn, familiar with nature package nltk. I think our text classification model does can help energy company to classify news automatically and create real value.

Tang Yue:

This project gives me an opportunity to practice what I learned in class and explore how to apply text analysis technology to resolve problems in real business world. The objective of our project is to extract important information (potential topic) from news, retrieve target news and use machine learning model to automatically distinguish influential news from non-influential news. I mainly participate in news classification part. Before building classification, initially we did simple data processing and used naïve bayes and SVM model to classify news, but the result is not good. So, we use more text processing technologies to process data. In this process, I studied the text processing technologies including their respective processing tools, functions, and application. For example, in classification model we use spaCy library to complete tokenization, removing stop words, stemming and lemmatization etc. And after elaborated text processing, our classification models perform good, which lets me know the importance and necessity of data processing. Besides, this project also guides me to use mathematic method to process and analyze data no matter the data is text or numeric. For example, we use TF-IDF to prepare data before feeding in models. Moreover, in this project, I studied how to use spaCy, NLTK and Scikit-learn to process data and build corresponding models to achieve our analysis goals.

Zhang Jiale:

This project is one of my internship projects. When we start our project, the first thing I do is share with my team members the domain knowledge since I am the only person understand what the company is looking from the refinery news and why they are looking for that information. Then based on the pre-processing work of Xiaolan, I updated some codes in order to ensure the corpus can be successfully tokenize and generate the CSV file for later use. In terms of the four tasks, I mainly contribute to 2 of them, one is topic modelling, another is information extraction. I build a base LDA model then Xiaolan further refined the model and Dong Fang analyzed the final results. I also tried use regular expression to do information extraction, but find it is not a good solution. With the help of Dong Fang, she shared with me many external sources to do information extraction, I learned how to fine tune a pre-defined model to train user defined NER using spaCy. For some information, I use tokenization and self-define functions to extract the information. And for the NER ‘reason’, I manually annotated training data then trained the model to automated detect ‘reason’. Although I did not engage too much on the model part of text classification, I contribute more on the labelling process. I make the guidance to my team members how to label the news, and I finished labeling all the news then compare their labelling results with mine to ensure our labels are correct. In terms of the learning experience, the best things I learned from this project are the good package spaCy and use a pre-trained model to train self-defined NER. Moreover, our team members are awesome, we have a good cooperation with each other. I like this project because it let me feel we are generating value for a real business and what we have learned are powerful.

Yin Xiaolan:

We need the knowledge within and go beyond the learning in the classroom to implement this project. During the process we practiced what we have learned in lecture and lab and tried some new packages of doing text analysis. When we are discussing about the project, I already had the idea of doing document retrieval. So, I did part of the data representation, change the single txt file to a txt corpus, then I did the design and experiment part of the document retrieval and help Jiale further implement the model building part of topic modelling, and then added the evaluation of the models, topic distribution, label ratio, top n keywords of each topic, also the word cloud. In addition, I designed the methodology, data representation, document retrieval, conclusion, and reflection part of the ppt. From the teamwork I see that the communication is very important. We can share our ideas and align what we want to do. I like the project since we can practice different text analysis method and solve real business problems.

**9. Reference**

Shrivarsheni. (2020, June 12). How to Train spaCy to Autodetect New Entities (NER) [Complete Guide]. *Machine Learning Plus* . Retrieved from https://www.machinelearningplus.com/nlp/training-custom-ner-model-in-spacy/

Mikolov, Tomas & Sutskever, Ilya & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems. 26.

1. Note: some information is hide by the blue box because of data protection purpose. [↑](#footnote-ref-2)