Documentation on the progress of Model with Unsupervised Learning and Oracle Server

# Introduction:

Incidents:

In ITIL terminology, an “incident” is defined as an unplanned interruption to an IT service OR reduction in the quality of an IT service.

*The Incident Process shall include:*

*Steps that should be taken to handle the incident*

*Responsibilities*

*Timescales and thresholds for completion of the actions*

*Major incidents are steered via higher impact and urgency = critical severity level*

An incident remains an incident forever: it may grow in impact and/or urgency to become a major incident, but an incident never becomes a problem.

Incident Tickets:

The shift or Line leader for BE and automation maintenance team for FE who has an overview on the business impact, provide 1st level and known resolution support. He or she raise incident to the 1st level support or IT service desk if not able to resolve or no known resolution available. Next remedy ticket is being created by the 1st level support.

What is Symptom Analysis:

Symptom analysis is a natural language processing technique which is able to provide early-stage insights of the problem statements. It captures the key symptoms of the problem from the statement to facilitate the user of gaining an easy overview of the problem.

Symptom Analysis on Incident Tickets:

Incident tickets are raised based on issues occurred during different backend processes.

Symptom analysis will be able to understand which problems are very similar to each other or the problems which contains similar type of words. Furthermore, it can generalize the problems into certain number of symptom areas. Moreover, An incident ticket can belong to multiple number of symptoms based on the words that has been used for describing the problem.

For example,

User not able to view the Lot Test Summary in IFX WIP Main in Camstar

<Test Summary> button grey out.

In the above example, it can be observed that two type of symptoms exist in the above problem statement. One is missing test summary and other one is button grey problem.

Hence, we need to find out which model will have the capability to recognize both of these symptoms.

Data description:

Data: All raw incident ticket details in text format from 2021 to May 2022.

Data count: 9583

Data columns used: Ticketid, Problem Description.

Problem Description example

Type 1:

(1) Problem description on error\*: (Including screenshot of error)

Error during trackout

? 0 - Camstar.LotSplitByItems: SplitLot\_E0100: ZA942324M0J total NDPW (Item Qty) does not match its Lot Qty

(2) Site\*: Mark "x" for user site

[ ] BTH [ ] MKZ PLT [ ] WUX CC [ ] TIJ

[ ] SIN [ ] MKZ SCC [ ] WUX DS [ ] RBG

[ ] MKZ PLA [ ] WUX HPS [ ] CJJ

(3) Lot Number: ZA942324M0J

(4) Equipments affected:STP03

(5) Equipment/PC name:

(6) Camstar Server\*:

https://faqstorage.infineon.com/KnowledgebaseArticle125028.aspx

(7) Integration application: Mark "x" for other application

[ ] AWI [ ] GPN [ ] XMES

[ ] DDM [ ] StMS [ ] XTEST UI

[ ] EAF [ ] Workstream

(8) Printer:

https://faqstorage.infineon.com/KnowledgebaseArticle125029.aspx

(9) Referred to FAQ\*:

(10)Affected area contact number\*: 6062515654

\* Mandatory fields. ISCW8CC938477D.infineon.com, 172.21.97.9, INFINEON\AlisMoha, MKZ, ,

Type 2:

ContactNumber:

0271

Affected Area Contact Number:

0271

Requestor Department:

Test

Lot Number:

NA

Product:

NA

Equipments Affected:

NA

Problem Description:

It was informed that some products will switch divisions and PLs in the new fiscal year. WIP\_DATA\_HOURLY excel file from \\klmsdn02.ap.infineon.com\SRFiler\SIN\Camstar is not reflecting the changes, one of which is the switch of some M63xx devices from PL58 to PL30. Please ensure changes are reflected accordingly.

Why Data Extraction is required:

The above tickets show that the data is coming in a ticket format. The tickets contain many fields which are important to the BOC team and for future reference but not useful for a machine learning model.

Hence, we need to extract out only the problem description which can be fed to an algorithm to learn the description similarity.

Data Extraction Procedures:

When the tickets are imported into a python workspace, data will have the below structure:

'(1) Problem description on error\*: (Including screenshot of error) \r\nEAF.CreateEAJob: Cannot create job. Equipment is not in control state ONLINE/REMOTE\r\n\r\n(2) Site\*: Mark "x" for user site\r\n\r\n[ ] BTH \t\t[ ] WUX CC\t\t[ ] TIJ\r\n[ ] SIN \t\t[ ] WUX DS\t\t[ ] RBG \t\t\r\n[x ] MKZ \t\t[ ] WUX HPS\t\t[ ] CJJ \r\n\r\n(3) Lot Number:\r\n1E129234G01\r\n\r\n(4) Equipments affected:\r\nSIM026\r\n\r\n\r\n(10)Affected area contact number\*:\r\n01135389609\r\n062515160\r\n\*Change Summary to most fitting CATEGORY:\r\n\r\nButton is grey out in FAJob Txn\r\nThe FAJob..does not belong to this Equipment\r\nLot is not allow for [EQ]\r\nCamstar: Track out / in issue\r\nCamstar: Move out / in issue\r\nLot CDO (InstanceId="xxx") not found\r\nMissing Test Summary\r\nStrip map total qty does not match lot qty\r\nLot is not allowed to transact\r\nSBA State (waitingtostart/not completed)\r\nStrip state mismatch with Camstar lot state , INFINEON\\MohdYuRo, MKZ/B8.3.R01.P, ,'

We need to extract only the yellow part which will be training data for our model.

Hence, it is required us to use Regular expression to extract all the important portions from incoming data.

An example of the expression =/"((problem description)|(issue))(.\*)(?=((\(2\) site\W)|(site\W)|(equipment name\W)))"/

Why Data Pre-processing is required:

The extracted data is not always in the format which is not always machine understandable. There are some special characters or may be some new line symbol or tab. Hence, it is required us to remove all those before we proceed with further data processing.

In here we are using python replace function to remove those with single space.

For example:

processed['raw'] = processed.Desc.map(lambda x:x.replace("\r",""))

Next, we need to remove the stopwords which creates noise in the data or which are not important towards the model goal.

We have created a manual list of noisy words alongside the NLTK stopward list.

After that, the words need to be lemmatized to bring the words to their base form. Different format of a single word can be different meaning to a algorithm which we don’t want. So, lemmatization is very important.

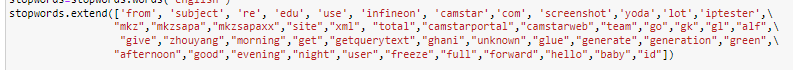
In details steps are mention below:

# Data Pre-Processing:



Additional words which can reduce the performance of a model due to its existence in multiple incidents symptoms can be removed by adding it to the extended stopwords list.

Current extended stopwords list holds the below words:



# NLP Model:

Processed data can now be used in the NLP algorithm for generating the predictions.

During selection of model I have explored quite a lot of models which are as follows:

1. KNN
2. LSTM by manually creating the labels for each symptom
3. Gensim LDA model
4. Sklearn LDA model
5. Sklearn NMF model
6. BertTopic which is a pretrained model
7. ClusterTransformer

Above models didn’t produce good result except the sklearn LDA model, which is also due to constrained of time and good knowledge of the hyperparameters. Please explore more with those models if possible or can try any other models with transformer.

One more reason of going for the unsupervised modeling is manual labeling which requires someone to understand each sentence out of 9000 sentences in details.

So far, I was able to extract 5 well explained topics which is also giving good confidence score for the sentences belonging to those particular topics with the help of sklearn LDA model.

After cleaning operation, the data has the below pattern:

“move strip map quantity match quantity adjust quantity“

Hence, these are the words which model will try to find out in other sentences belonging to the same topic.

* Let me introduce you to the model a little bit:

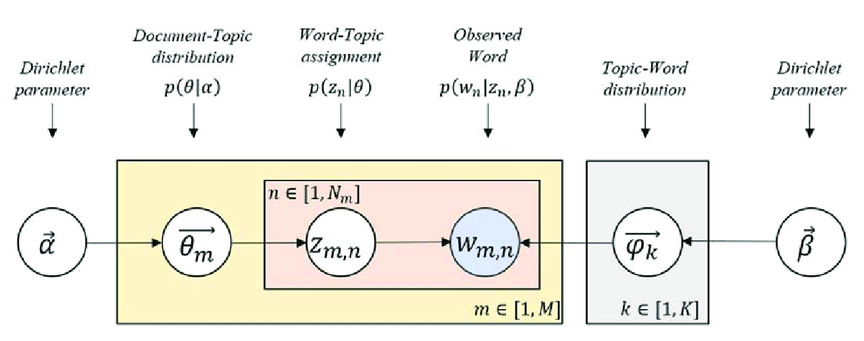
For the given dataset which is very unstructured and manual labeling is very costly in terms of man power and time, can be categorized by the use of Topic Modeling technique of NLP. Topic Modeling has the ability to extract the words which is very close to words belonging to a certain topic. That means it calculates distance between words of a particular sentence and words belonging to two different sentences to minimize the first one and maximum the later one.

Topic Modeling core Idea:

Topic Modeling works on the concept of Latent Dirichlet Allocation.

Wikipedia Definition:

“In [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), the **latent**[**Dirichlet**](https://en.wikipedia.org/wiki/Dirichlet_distribution)**allocation** (**LDA**) is a [generative statistical model](https://en.wikipedia.org/wiki/Generative_model) that allows sets of observations to be explained by [unobserved](https://en.wikipedia.org/wiki/Latent_variable) groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. “



= controls per document topic distribution

= controls per topic word distribution

Apart from LDA, there are other topic modeling methods for example NMF or LSA which are also efficient topic modeling tools.

LDA way of finding probability of a word belonging to a topic:

p(word w with topic t) = p(topic t | document d) \* p(word w | topic t)

First, LDA starts to randomly assign topics to each word and iteratively improves it through a algorithm named “Gibbs Sampling”.

Main Hyperparameters of LDA model:

n\_components

batch size

leanring method

topic\_word\_prior

doc\_topic\_prior

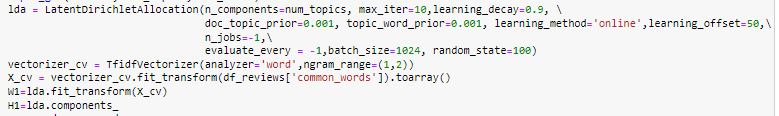
n\_jobs

max\_iter

learning\_offset

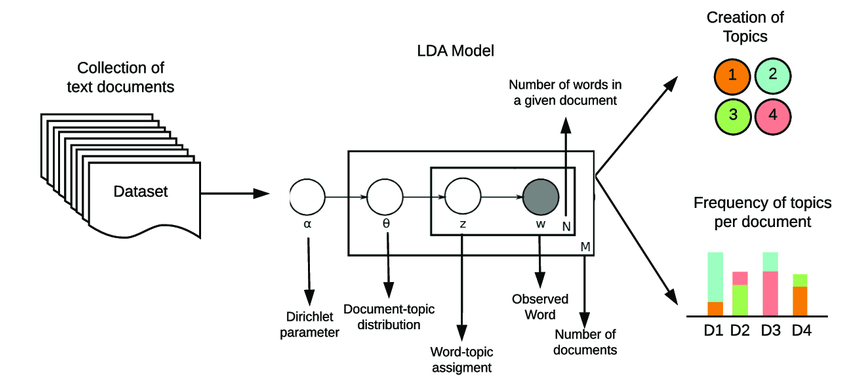
learning\_decay

For training:



For finding out the topic of a new problem description:





# Coherence score or Topic Coherence score:

Topic coherence score is a measure of how good a topic model is in generating coherent topics.

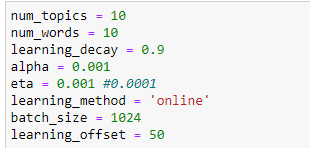
A coherent topic should be semantically interpretable and not an artifact of statistical inference.

# Grid Search CV:

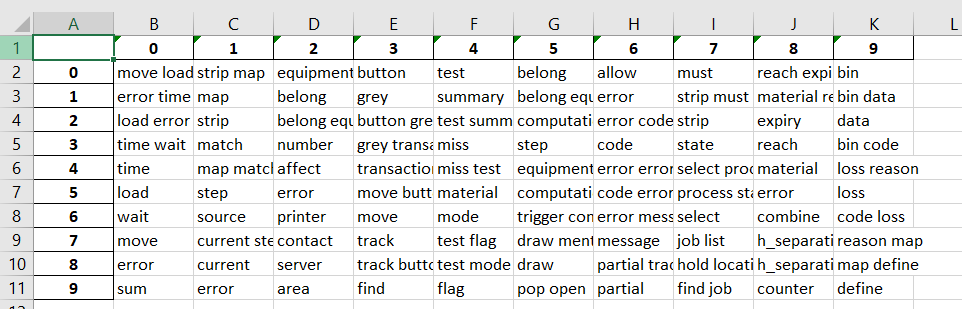
Here, I am trying to find out which hyperparameters give the highest coherence score.



# Sample Output:



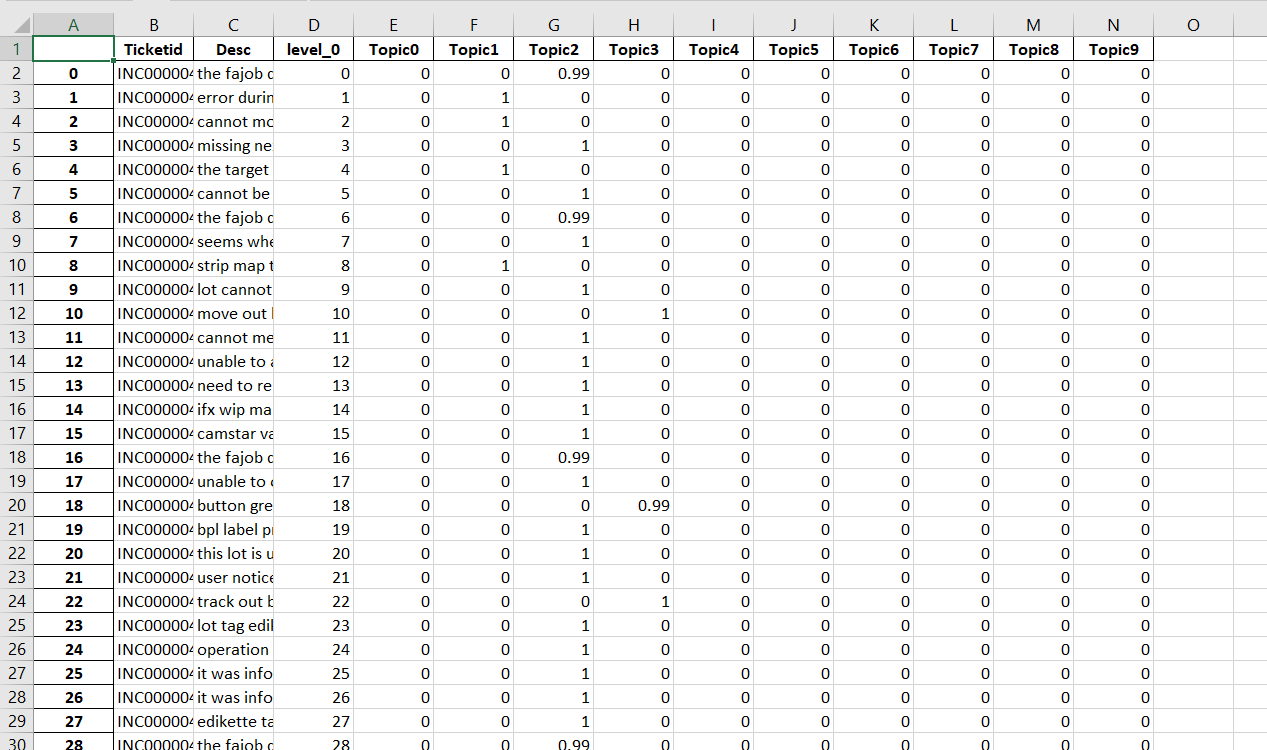
**Table1: Different Topics and words belonging to each topic**



This column shows the words which can represent a particular topic. For example, strip map, match words can relate to the stms error.

The words in a single column can repeat since it’s coming from different unstructured sentence which model is captured by the model. So it is not possible to remove easily.

**Table2: Shows percentage of each ticket belonging to a certain topic**



After prioritizing top 5 words of each group (Table 1), we can see that the topics are not well distributed among the sentences. There are many tickets which only belong to topic 2 which is incorrect according to my verification from the verification of Desc column and the topic table.

Note!!!

Currently, I am looking into fine tuning the hyperparameters and other models too for example LSA or NMF type of. I will include those as soon as I made some progress. For this type of model manual verification is very important for human justification. So, I hope Chin Meng will help in that scenario.

Validation with the JCBE Data:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| num\_topics | num\_words | Learning\_decay | Alpha | Eta | Learning\_method | Batch\_size | Learning\_offset | N\_top\_words | Accuracy |
| 10 | 20 | 0.9 | 0.001 | 0.0001 | Online | 512 | 9 | 20 | 75 |
| 20 | 20 | 0.9 | 0.001 | 0.001 | Online | 512 | 50 | 20 | 86 |
| 20 | 20 | 0.9 | 0.001 | 0.001 | Online | 512 | 100 | 20 | 90 |
| 10 | 20 | 0.9 | 0.001 | 0.001 | Online | 512 | 100 | 20 | 87 |
| 10 | 20 | 0.5 | 0.001 | 0.001 | Online | 512 | 100 | 20 | 67 |
| 20 | 10 | 0.9 | 0.001 | 0.001 | Online | 512 | 100 | 10 | 90 |

Validation by CM with the set of hyperparameters from the first row:

|  |  |
| --- | --- |
| Button Grey | 88% |
| FAJob Issue | 98% |
| Test Summary | 96% |
| Strip Queue | 44% |
| Strip Quality` | 76% |

* Different set of hyperparameters:



num\_topics = 15

num\_words = 12

learning\_decay = 0.9

alpha = 0.001

eta = 0.001

learning\_method = 'online'

batch\_size = 512

learning\_offset = 20

n\_top\_words = 12

epochs = 100

random\_state = 50

ngram\_range =(1,2)

max\_df = 0.9

min\_df = 2

max\_features = 5000

the topic table name is

**topics\_1028\_2**



num\_topics = 16

num\_words = 10

learning\_decay = 0.9

alpha = 0.001

eta = 0.001

learning\_method = 'online'

batch\_size = 512

learning\_offset = 20

n\_top\_words = 10

epochs = 200

random\_state = 100

ngram\_range =(1,3)

max\_df = 0.9

min\_df = 2

max\_features = 5000

the topic table name is **topics\_1028\_3**



num\_topics = 20

num\_words = 10

learning\_decay = 0.9

alpha = 0.01

eta = 0.001

learning\_method = 'online'

batch\_size = 512

learning\_offset = 10

n\_top\_words = 10

epochs = 200

random\_state = 50

ngram\_range =(1,3)

max\_df = 0.9

min\_df = 2

max\_features = 5000

the topic table name is **topics\_1028\_4**



**num\_topics = 20**

**num\_words = 20**

**learning\_decay = 0.9**

**alpha = 0.001**

eta = 0.001

**learning\_method = 'online'**

**batch\_size = 512**

**learning\_offset = 10**

**n\_top\_words = 10**

**epochs = 150**

**random\_state = 50**

**ngram\_range =(1,3)**

**max\_df = 0.9**

**min\_df = 0.01**

**max\_features = 5000**

the topic table name is **topics\_1028\_5**

6.

num\_topics = 20

num\_words = 12

learning\_decay = 0.9

alpha = 0.001

eta = 0.001

learning\_method = 'online'

batch\_size = 512

learning\_offset = 10

n\_top\_words = 12

epochs = 300

random\_state = 100

ngram\_range =(1,3)

max\_df = 0.9

min\_df = 2

max\_features = 5000

the topic table name is **topics\_1028\_7**



num\_topics = 20

num\_words = 10

learning\_decay = 0.9

alpha = 0.001

eta = 0.001

learning\_method = 'online'

batch\_size = 512

learning\_offset = 10

n\_top\_words = 10

epochs = 200

random\_state = 50

ngram\_range =(1,3)

max\_df = 0.75

min\_df = 0.001

max\_features = 10000

the topic table name is **topics\_1028\_6**

Testing With pretrained model BERTopic:

|  |  |  |  |
| --- | --- | --- | --- |
| Top\_n\_words | Nr\_topics | N\_gram\_range | Accuracy |
| 20 | 10 | 1,3 | 53 |
|  |  |  |  |
|  |  |  |  |

Different Approach:

Cluster Transformer:

Month of may - button grey problems

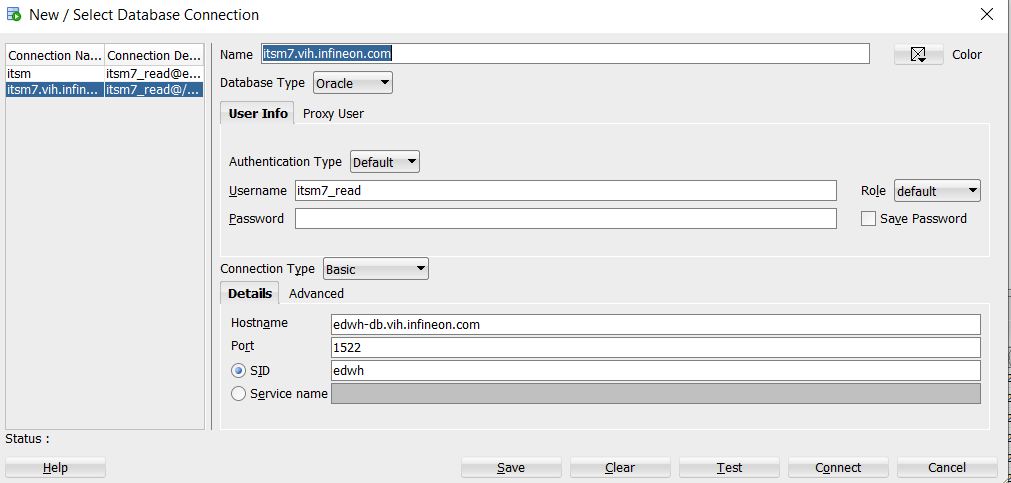
Calender week 18 to 24

Sin, batam, wux – 8.5

Printer topic can be done using self resolution

Common Error which I faced:

Oracle server access:



API name: OracleData.py

Use the below parameters to run the API:



Reference:

1. <https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#:~:text=In%20natural%20language%20processing%2C%20the,of%20the%20data%20are%20similar>.
2. <https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2>
3. <https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/>
4. <https://www.mygreatlearning.com/blog/understanding-latent-dirichlet-allocation/>
5. <https://towardsdatascience.com/topic-modeling-with-bert-779f7db187e6>
6. <https://colab.research.google.com/drive/1FieRA9fLdkQEGDIMYl0I3MCjSUKVF8C-?usp=sharing#scrollTo=SNa-KtKDRnus>
7. <https://github.com/MaartenGr/BERTopic>
8. <https://camstarstgr8.sin.infineon.com/CamstarPortal/default.aspx> ---> camstar malaka

chatbot:

https://confluencewikiprod.intra.infineon.com/display/INTPS/0.+How+to+prepare+for+a+Chatbot

Appendix:

# Documents on Maintaining server and jupyterhub

How to access the python virtual environment: https://confluencewikiprod.intra.infineon.com/pages/viewpage.action?pageId=381410499

[Install Kernel for Use in JupyterHub](https://confluencewikiprod.intra.infineon.com/display/ITARISE/Install+Kernel+for+Use+in+JupyterHub):

<https://confluencewikiprod.intra.infineon.com/display/ITARISE/Install+Kernel+for+Use+in+JupyterHub>

Oracle Connection with Python:

<https://confluencewikiprod.intra.infineon.com/display/ITARISE/Connect+Oracle+inside+JupyterHub+Notebook>

Infineon Cloud Server Support:

<https://confluencewikiprod.intra.infineon.com/display/ICPC/HICP%20Community%20Home>

file transfer to the project folder:

<https://confluencewikiprod.intra.infineon.com/display/ITARISE/Transfer+Files+from+Windows+PC+to+your+Unix+Project+Account>

<https://confluencewikiprod.intra.infineon.com/display/ITARISE/Transfer+Files+from+Windows+PC+to+your+Unix+Personal+Account>

new python package:

https://confluencewikiprod.intra.infineon.com/display/ITARISE/Generate+a+Python+Virtual+Environment+in+HPC

# Proposed Chat Bot Idea:

Problem Description By user

User’s verification if the topic is matching with his problem

Topic and relevant words recommended by the bot

BOC support experts

No

Automatic resolution or redirected to the support team specialized in that field

Yes