FTS log analysis and clustering

Micol Olocco

University of Physics, Torino

Supervised by Federica Legger

OPERATIONAL INTELLIGENCE

Operational Intelligence **(OI)** is a cross-experiment project aiming to reduce the cost of computing operations for WLCG experiments:

- by increasing the level of automation in operation tasks

  → **NLP algorithms development**

- by leveraging common tools and infrastructure

  → **k8s deployement**

- by collaborating and sharing expertise, approaches and solutions to common problems among experiments

OPERATIONAL INTELLIGENCE

Operational Intelligence (**OI**) is a cross-experiment project aiming to reduce the cost of computing operations for WLCG experiments:

- by increasing the level of automation in operation tasks

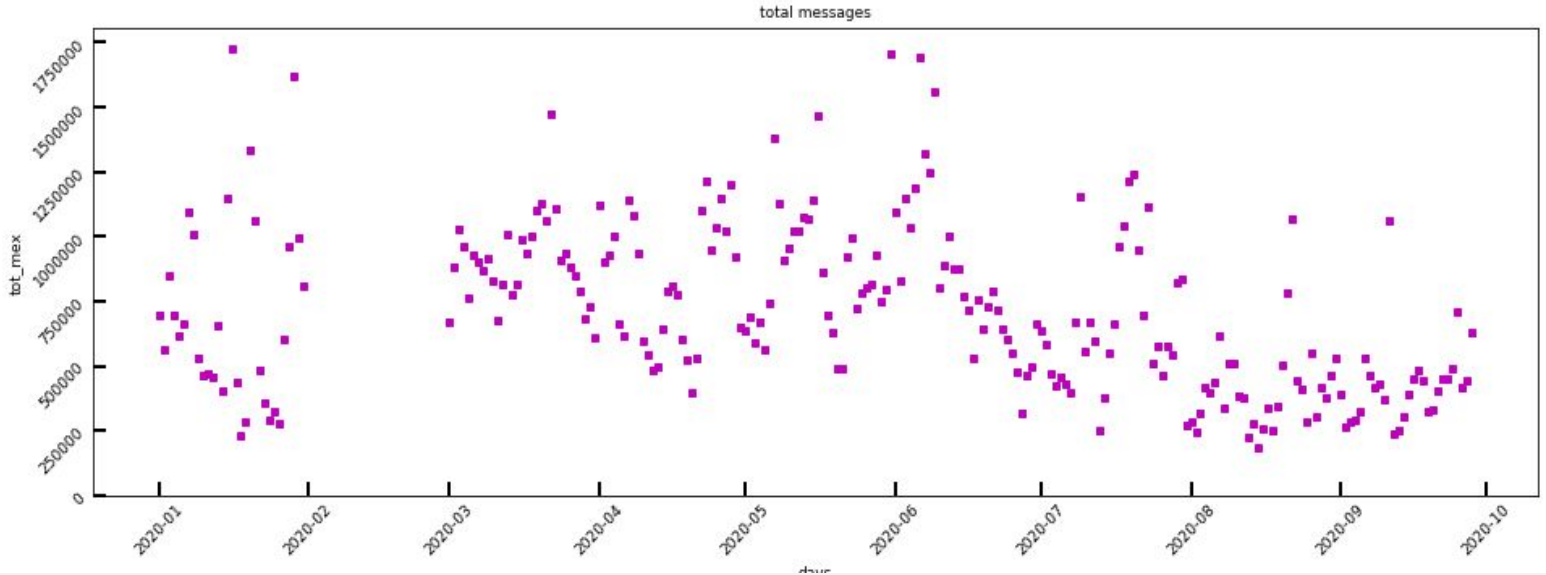
  → **NLP algorithms development**

- by leveraging common tools and infrastructure

  → **k8s deployement**

- by collaborating and sharing expertise, approaches and solutions to common problems among experiments

FTS error messages over time



~10^6 messages per day

FTS Log-analysis workflow

Getting *FTS* error messages from *HDFS*

↓

Clustering the logs with the *clusterlogs* module

↓

Injecting data with the information about clusters to *MonIT* via *StompAMQ* module

↓

Visualizing results in a Grafana dashboard

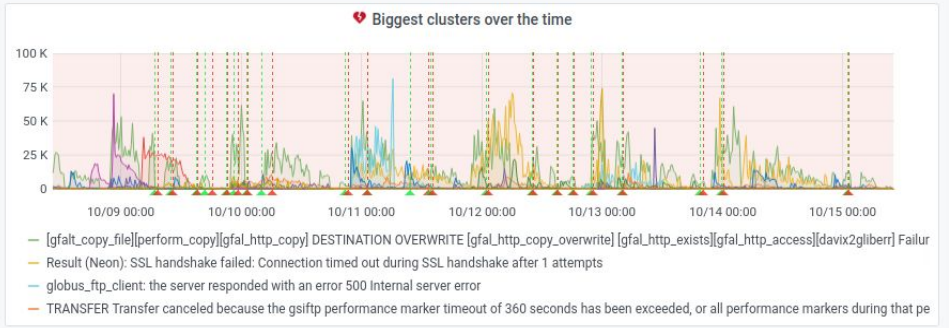
custerlogs module original version by Maria Grigorieva:
https://github.com/maria-grigorieva/ClusterLog/tree/development/clusterlogs

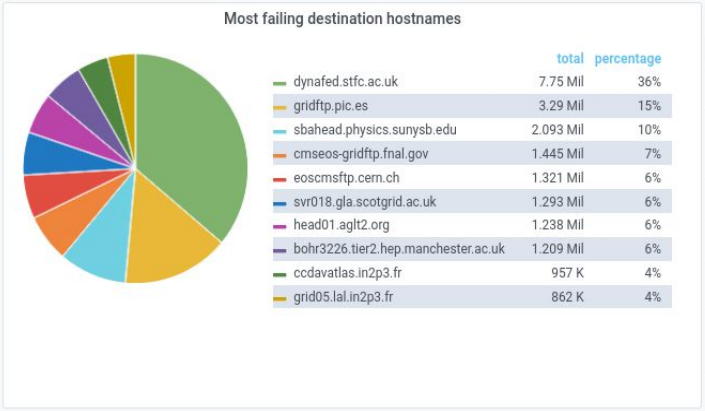
https://monit-grafana.cern.ch/d/Zx_bXneWz/fts-log-clustering?orgId=11

GRAFANA dashboard

Production / FTS log clustering

Last 7 days UTC





10 biggest clusters by the top 3 destination hostnames

data.cluster_pattern	data.dst_hostname	Count
[gfalt_copy_file][perform_copy][gfal_http_copy] DESTIN...	dynafed.stfc.ac.uk	7690773
Result (Neon): SSL handshake failed: Connection time...	ccdavatlas.in2p3.fr	920554
Result (Neon): SSL handshake failed: Connection time...	xrootd.physik.uni-bonn.de	477895
Result (Neon): SSL handshake failed: Connection time...	webdav.mwt2.org	259642
globus_ftp_client: the server responded with an error 5...	grid05.lal.in2p3.fr	825116
globus_ftp_client: the server responded with an error 5...	se01.dur.scotgrid.ac.uk	280055
globus_ftp_client: the server responded with an error 5...	atlas.dcache.nikhef.nl	21576
TRANSFER Transfer canceled because the gsiftp perfo...	srm.triumf.ca	214470
TRANSFER Transfer canceled because the gsiftp perfo...	srm.ndgf.org	155105
TRANSFER Transfer canceled because the gsiftp perfo...	dcache.ijs.si	70077
DESTINATION SRM_PUT_TURL error on the turl reques...	svr018.gla.scotgrid.ac.uk	1051709

Messages from all VOs (ATLAS, CMS, LHCb)

OPERATIONAL INTELLIGENCE

Operational Intelligence **(OI)** is a cross-experiment project aiming to reduce the cost of computing operations for WLCG experiments:

- by increasing the level of automation in operation tasks

  → **NLP algorithms development**

- by leveraging common tools and infrastructure

  → **k8s deployement**

k8s deployement:

- Entire codebase wrapped into docker image (including access to HDFS) and executed as a cronjob within k8s pod
- Deployement done within CMS monitoring k8s cluster

01 Log analysis: MY MISSION

Nickodemas's previous work is our state of the art. My mission is to study the process of analysis in details to find out parameters to play with.

My first observation: **Nickodemas's code uses similarity to make the clusters**
My first task: **use ML algorithms to cluster.**

Questions I'd like to answer:

How different is the clustering obtained by unsupervised learning? Is it "better"?
How can we evaluate the clustering quality?

In any case: **comparing the clusters obtained with differents models/techniques will be (or not) a validation of the current implementation.**

01 Log-analysis reminds: How

PROCESSING PHASE:

Two main steps to automatize the **individuation** and **categorization** of error message patterns:

- Transformation of the textual information into numeric (**VECTORIZATION**)
- Grouping of the numerical representations into meaningful error categories (**CLUSTERING**)

PRE-PROCESSING PHASE:

- **DATA PREPARATION:** Cleaning of the messages from “particular” (meaningless) information before vectorizing:
  -substrings with digits removed (ex. file path)
  -lowercase
- **TOKENIZATION**

01 CLUSTERING

Two possible approaches to cluster (after pre-processing):

- **NO ML**: similarity of error patterns defined as Levenshtein distance (number of edits needed to transform one word into another: insertion, deletion, replacement).

- **WORD EMBEDDING (ML):** mapping of a word into a numeric vector space using a dictionary. Similarity of two words given, for example, by the cosine of the relative angle.

Word2Vec
+
Sent2Vec

word2vec: algorithm to perform the mapping word->vector (by Tomas Mikolov and others, Google)

sent2vec: sentence representation computed as mathematical average of the word vector representations of all the words in each sentence

01 ML or not ML? That is the question

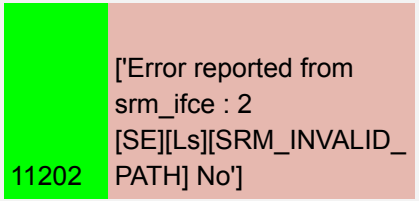
An example of clusters comparison (~12 clusters). The clustered day is 15th October 2020.
In link https://cernbox.cern.ch/index.php/s/qB28ySSdiF96GoQ you will find clusters obtained with:

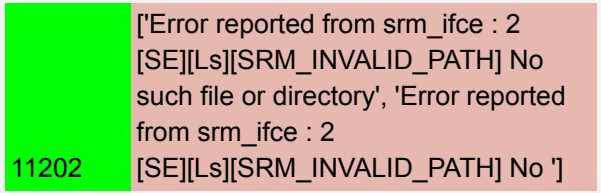
A. NO ML (number of clusters = 460)
B. ML after training from 1 January 2020 to 28 September 2020 (number of clusters = 279)
C. ML after training over a single day (number of clusters = 233)

Questions and observations:

a) Do ML clusters converge to NO-ML ones?
b) Even if same cluster size (between A. and B.) ML patterns look more complete (not broken off)
c) ...your questions and observations...

NO ML

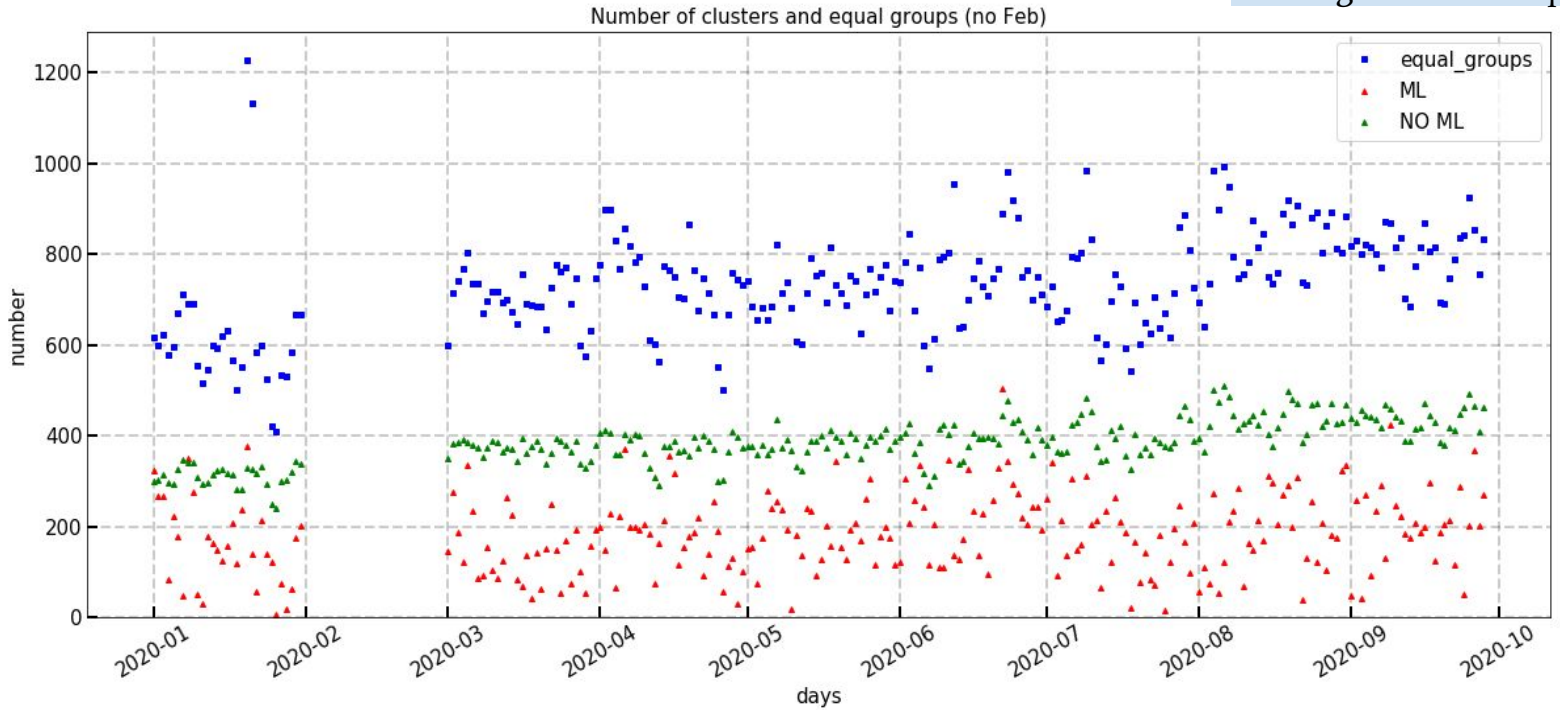




ML

Stability study

equal groups=groups of cleaned messages with same pattern



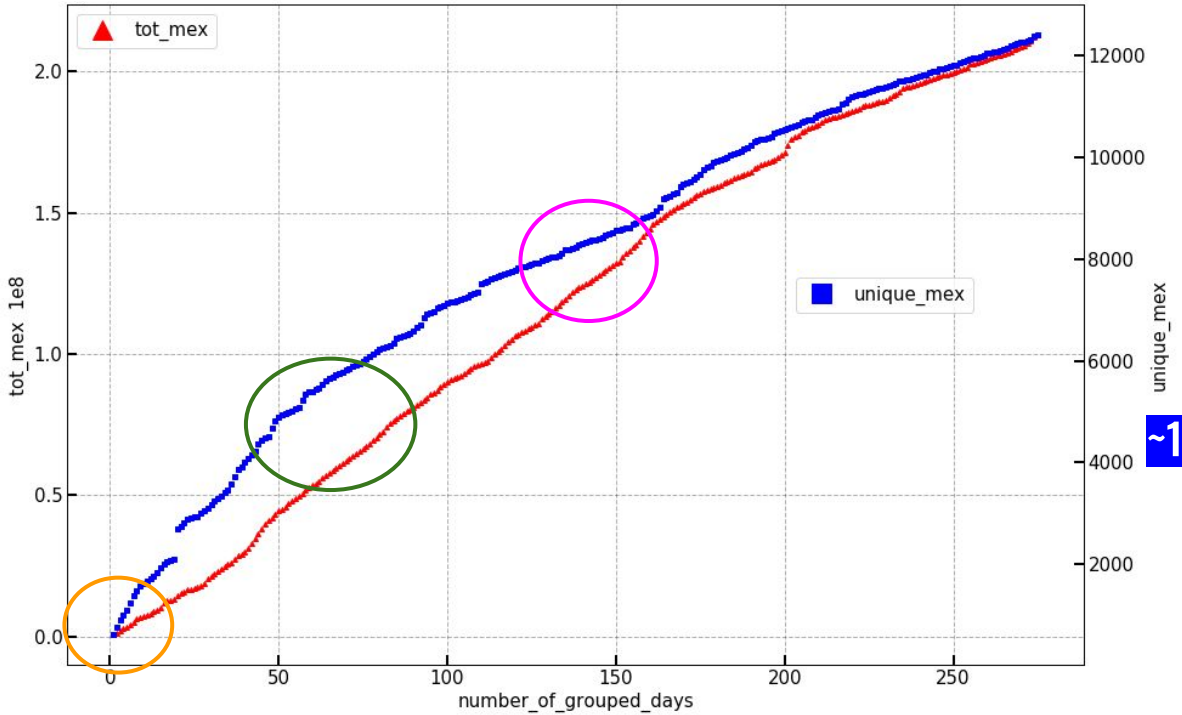
Observations:

1) less clusters by ML
2) number of ML clusters fluctuates more

Problems while running on February → missing

Stability study

number of total messages (left axis) and equal groups (right axis) over time



~10^8

~10^3

The variety of error patterns follows a different trend to the growth of the number of total messages (slope):

a) faster
b) equal
c) slower

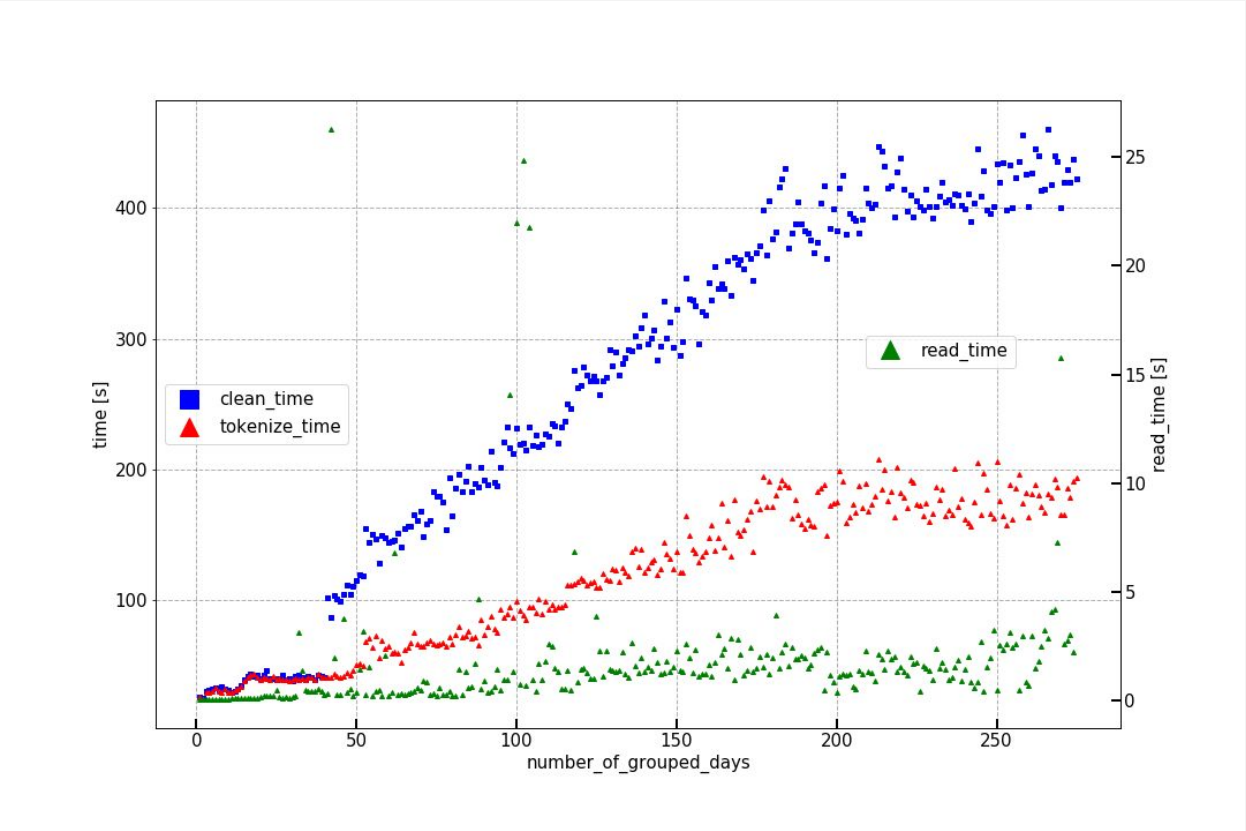
Why don't unique_mex saturate?

What is better for clustering new days?

A “super” trained model periodically updated?

A faster model but uncomplete?

Let’s start from studying the required time for:
-reading
-cleaning
-tokenizing



01 FURTHER PLANS

min_count=1 : threshold value for words. Only word with frequency greater than min_count are included into the model.
size=300 : number of dimensions in which we wish to represent our word. Size of the word vector.
workers=4 : used for parallelization

- Tune word2vec hyperparameters (size, iter, workers, window...)
- Test different clustering algorithms: DBSCAN, HDBSCAN, hierarchical
- Compare the top ten clusters obtained with different models: do they change?
- Is ML approach more advantageous? (time, quality of clusters, computing power..)

Is ML model able to recognize small changes in error patterns? (if trained, it should be clever!)

If you are interested in visualizing word2vec mechanism, here a very nice demo

Thanks for your attention!