

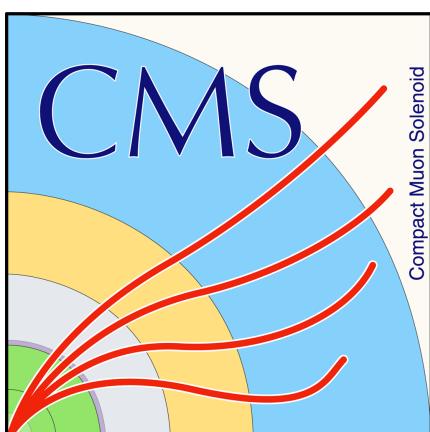
Automatic log analysis with NLP for the CMS workflow handling

CHEP 2019

Daniel Robert Abercrombie¹, Hamed Bakhshiansohi², Sharad Agarwal³,
Jennifer Adelman-McCarthy⁴, Andres Vargas Hernandez⁵, Weinan Si⁶,
Lukas Layer⁷, Jean-Roch Vlimant⁸

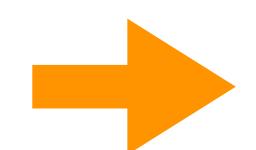
on behalf of the CMS collaboration

Massachusetts Institute of Technology¹, DESY², CERN³, FNAL⁴, Catholic University of America⁵,
University of California Riverside⁶, Università e sezione INFN di Napoli⁷, California Institute of Technology⁸



Automatization of failing workflow handling

- Central MC production in CMS utilizes the **LHC computing grid**
- Thousands of **workflow tasks** each with thousands of jobs on more than 100 sites worldwide
- A **certain rate fails** and has to be handled manually by **Computing Operators**: e.g. resubmit, kill, change memory, change splitting, etc.
- **Common errors** are missing/corrupt input files, high memory usage, etc.



Operational Intelligence: Automatize the failure handling using Machine Learning - CMS Tools & Integration group

Strategy and Dataset

- Information of ~ **33.000 failing workflow tasks** collected since 2017
- Actions** of the operators **stored**
- For each **task** we know on which **sites** how many times an **error code** was thrown

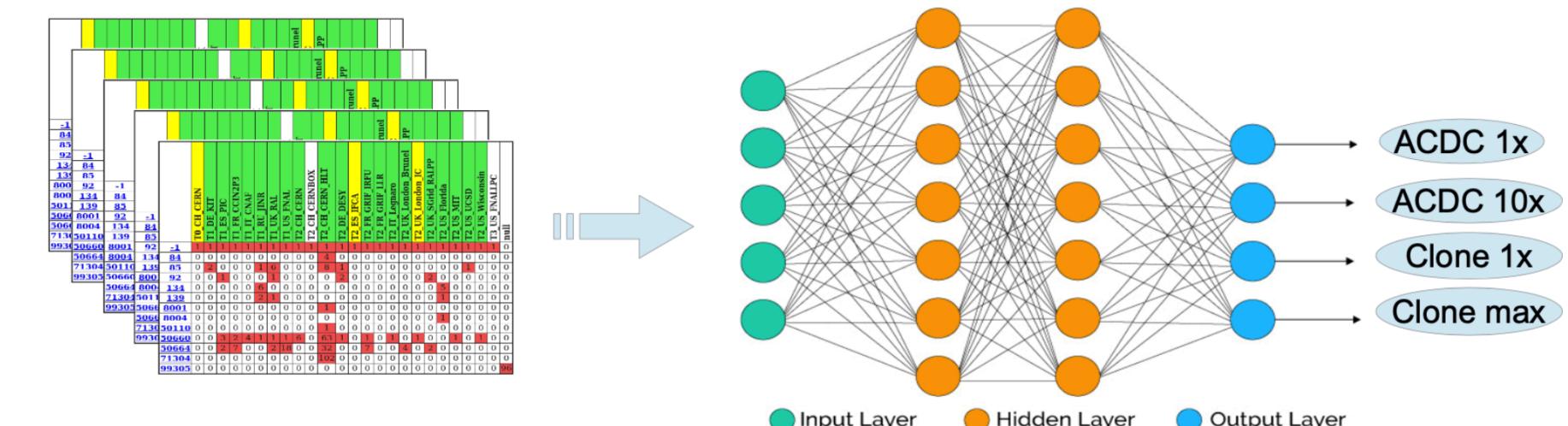
→ **Build a sparse matrix for each workflow**

→ **Goal: Predict the operator's action**

- Challenges: **small data, sparse input, class imbalance**
- Tried: **Feed-Forward NN, CNN, Embedding Models**
- Similar results** for different models

Site: T1_DE_KIT
Error code: 85
Counts: 2

| | T0_CH_CERN | T1_DE_KIT | T1_ES_PIC | T1_FR_CCIN2P3 | T1_IT_CNAF | T1_RU_JINR | T1_RU_RAL | T1_US_FNAL | T2_CH_CERN | T2_CH_CERN_HLT | T2_DE_DESY | T2_ES_IFCA | T2_FR_GRIF_JRFU | T2_FR_GRIF_LLR | T2_IT_Legnaro | T2_UK_London_Brunel | T2_UK_London_IC | T2_UK_SGrid_RALPP | T2_US_MIT | T2_US_UCSD | T3_US_Wisconsin | T3_US_FNAL_PIC | null |
|-------|------------|-----------|-----------|---------------|------------|------------|-----------|------------|------------|----------------|------------|------------|-----------------|----------------|---------------|---------------------|-----------------|-------------------|-----------|------------|-----------------|----------------|------|
| -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | |
| 84 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 85 | 2 | 0 | 0 | 0 | 1 | 6 | 0 | 0 | 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| 92 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| 134 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 |
| 139 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 8001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8004 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 50110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 50660 | 0 | 0 | 3 | 2 | 4 | 1 | 1 | 1 | 6 | 0 | 63 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 50664 | 0 | 0 | 2 | 7 | 0 | 0 | 2 | 18 | 0 | 0 | 32 | 0 | 0 | 7 | 0 | 0 | 4 | 0 | 2 | 0 | 0 | 0 | 0 |
| 71304 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 102 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 99305 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 96 |

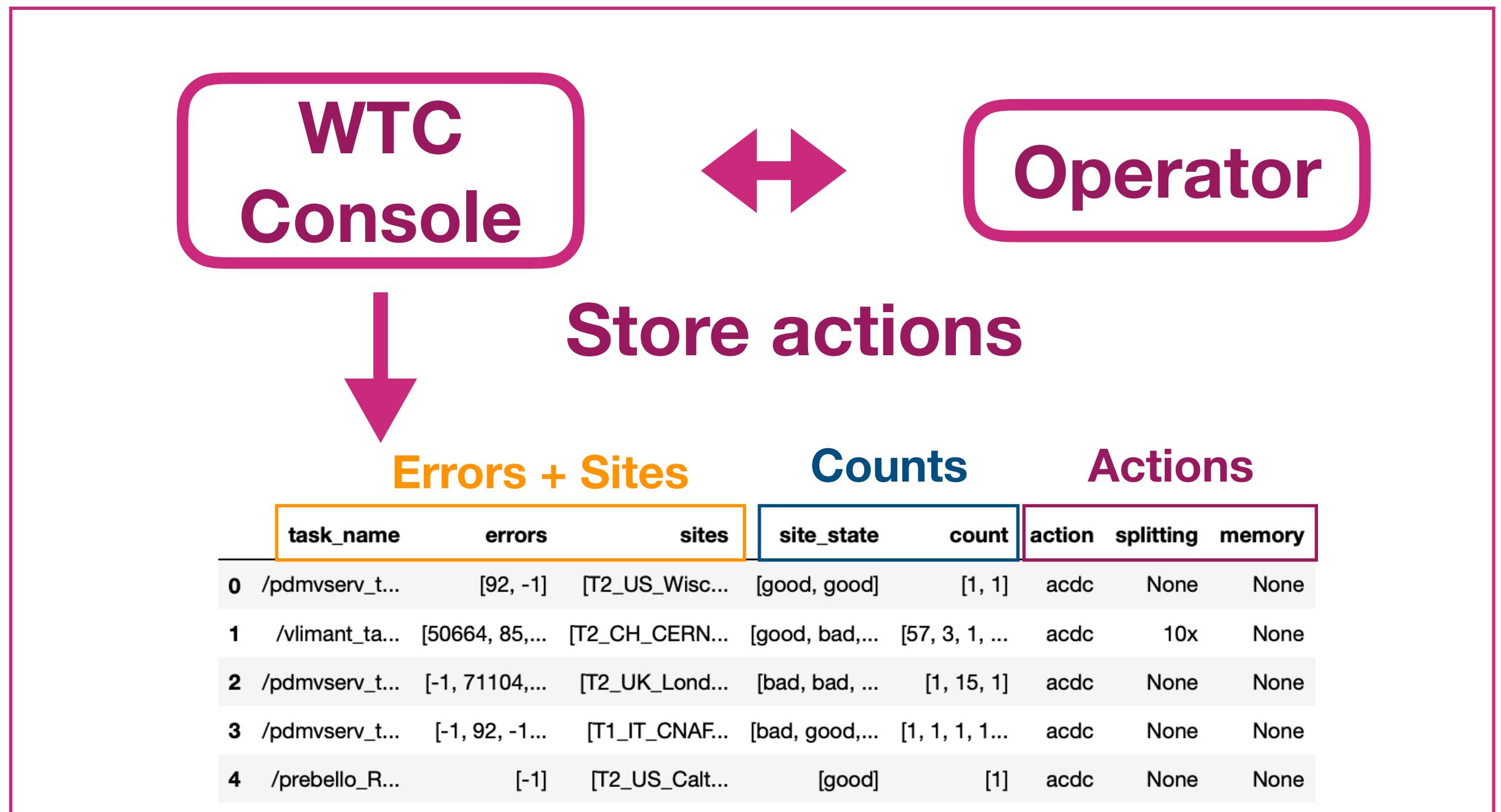


Poster at CHEP 2018:
<https://indico.cern.ch/event/587955/contributions/2937424/>

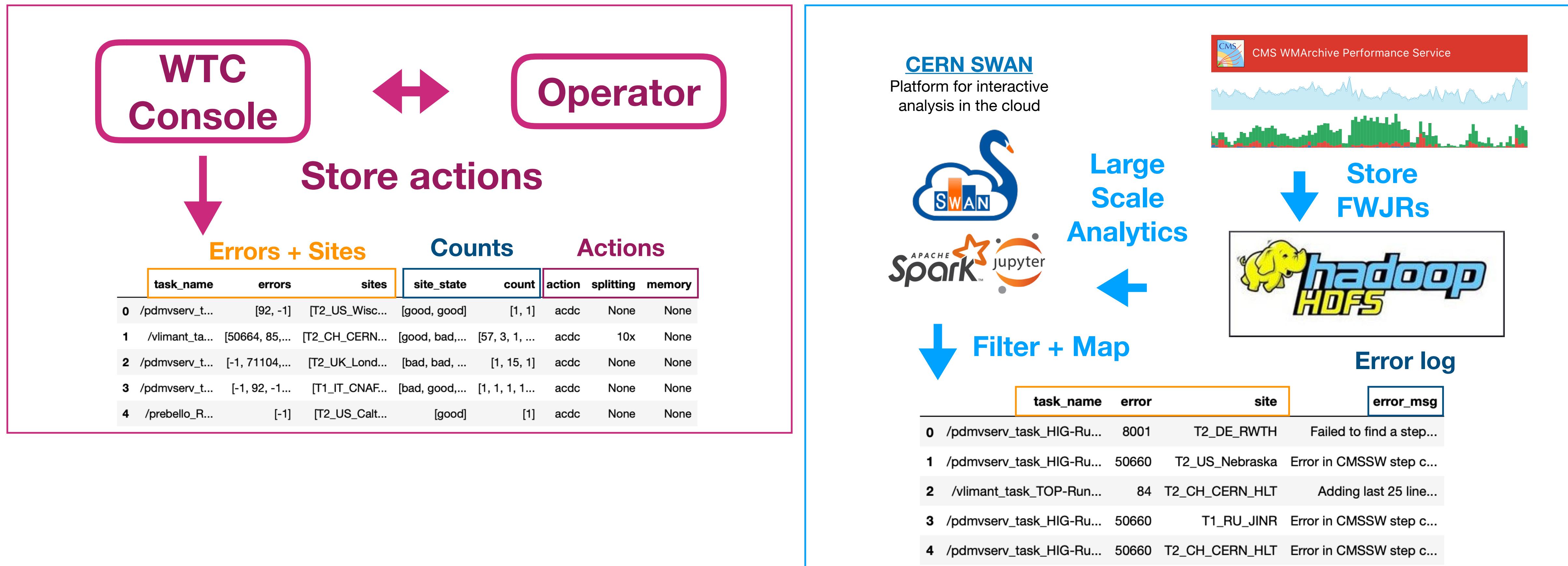
Dev:
<https://github.com/CMSCompOps/AIErrorHandler>

Idea for further improvement:
Addition of Error Logs using Natural Language Processing

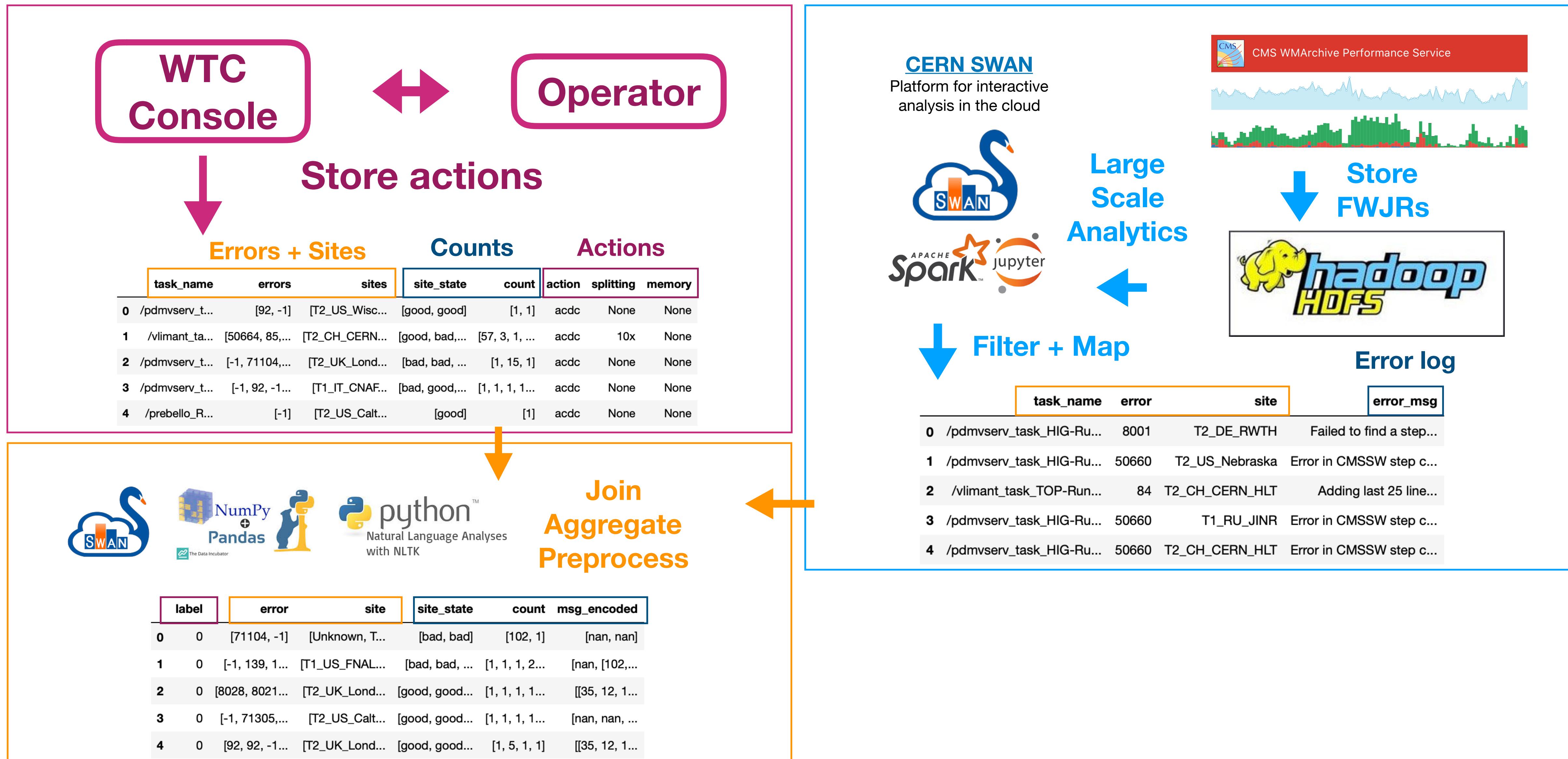
Pipeline



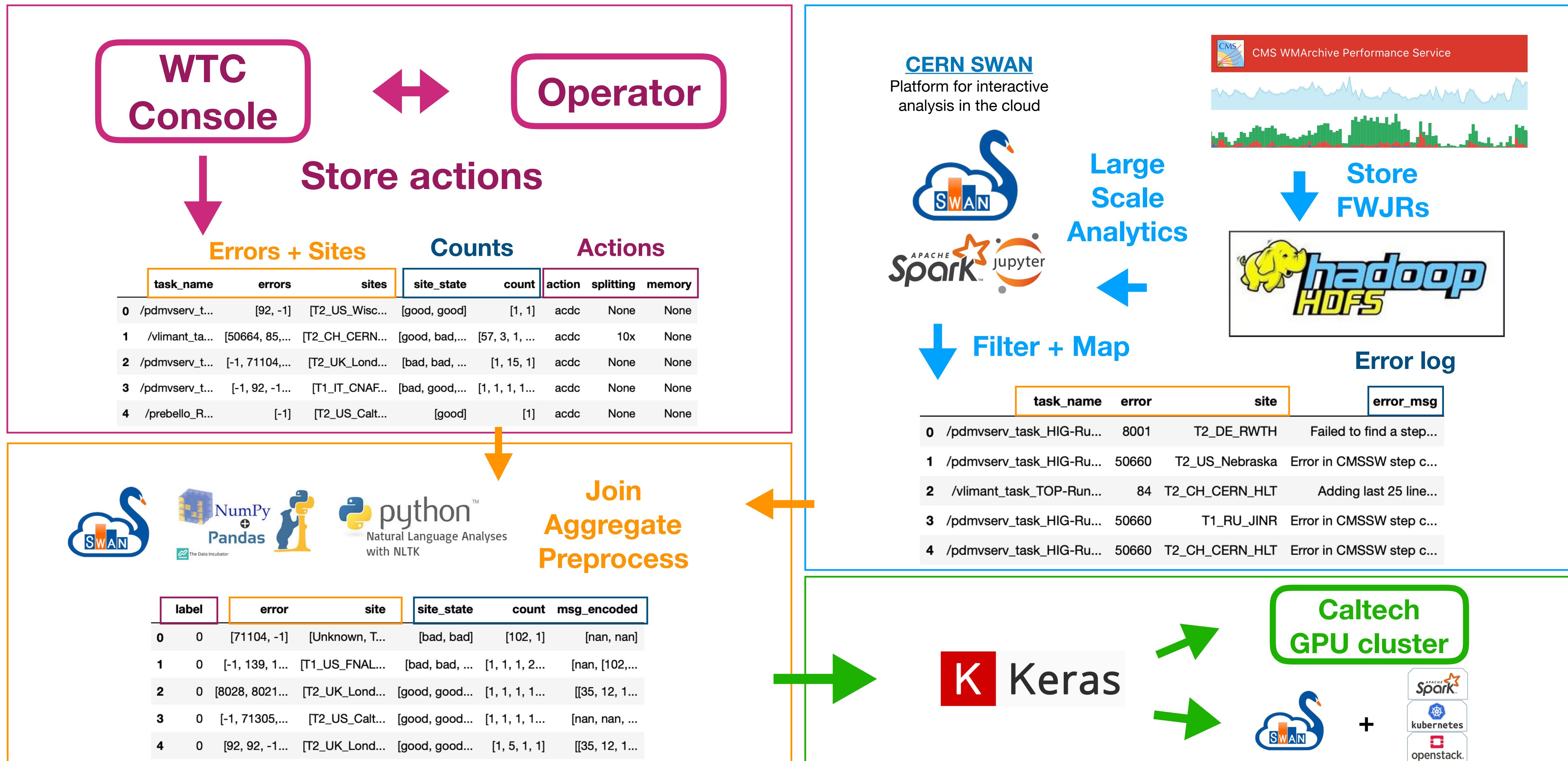
Pipeline



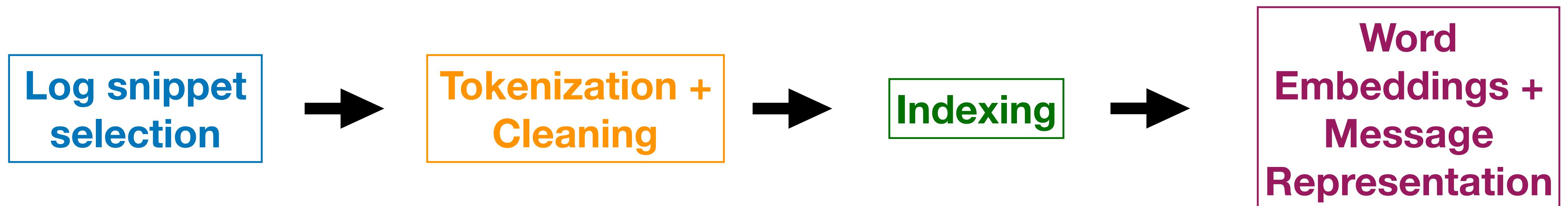
Pipeline



Pipeline



NLP Workflow



```
An exception of category 'FileReadError' occurred
while [0] Calling InputSource::getNextItemType
[1] Reading branch EventAuxiliary [2] Calling X
rdFile::readv() [3] Calling XrdAdaptor::Request
Manager::OpenHandler::open() Exception Message: Xr
dCl::File::Open(name='root://polgrid4.in2p3.fr//st
ore/data/Run2018A/EGamma/RAW/v1/000/315/363/00000/
76CD1546-854B-E811-9B3C-FA163EBA2BF6.root?tried=cc
xrpli003.in2p3.fr,ds21.kipt.kharkov.ua', flags=0x1
0, permissions=0660) => error '[FATAL] Connection
error' (errno=0, code=108) Additional Info:
[a] Active source: ccxrpli003.in2p3.fr:1095 (site
T1_FR_CCIN2P3)
```

```
[u'An', u'exception', u'of', u'category', u'FileRe
adError', u'occurred', u'while', u'0', u'Calling',
u'InputSource', u'getNextItemType', u'1', u'Readin
g', u'branch', u'EventAuxiliary', u'2', u'Calling',
u'XrdFile', u'readv', u'3', u'Calling', u'XrdAda
ptor', u'RequestManager', u'OpenHandler', u'open',
u'Exception', u'Message', u'XrdCl', u'File', u'Ope
n', u"name='root", u'tried=ccxrpli003.in2p3.fr', u
'flags=0x10', u'permissions=0660', u'error', u'FAT
AL', u'Connection', u'error', u'errno=0', u'code=1
08', u'Additional', u'Info', u'Active', u'source',
u'ccxrpli003.in2p3.fr:1095', u'site', u'T1_FR_CCIN
2P3']
```

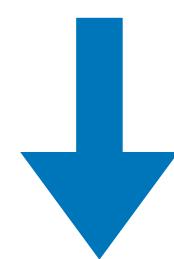
```
[35, 12, 10, 37, 186, 34, 25, 3, 11, 291, 532, 4,
188, 183, 273, 16, 11, 83, 251, 33, 11, 38, 77, 33
7, 17, 24, 23, 68, 18, 48, 41, 15875, 46, 47, 6, 4
39, 847, 6, 107, 2210, 65, 66, 531, 19, 867, 58, 1
68]
```

```
[-0.9386204  0.9671039  1.206119 -0.19411911 -0.23051228 -0.65913016
 0.00338563 -0.5044939 -0.02717912  0.09867108 -0.05116596 -0.5569398
 0.618001  -0.73519474  0.640754 -0.15959941  0.12680046  0.15426521
 0.42407024 -0.14709638  0.47534356 -0.18739735 -0.03758601  1.0197228
 0.06810962 -0.02504043 -0.11898965  0.11782035 -0.34788233  0.14381583
 0.67742926  0.43483204  0.39772424  0.10286205 -0.3774167  0.36935064
-0.1227726 -0.20352381 -0.22164229  0.10978855 -0.50076777  0.39308718
 0.1865182  0.33458158 -0.3371736  0.0493757  0.5454126 -0.94923306
-0.1352783 -0.34716246]
```

Unsupervised word embeddings

- Map **similar words** to **nearby points** in a high-dimensional vector space → Capture **relations** and **reduce dimensionality**
- Train with [Word2vec Skip-Gram](#) algorithm: model to predict context given a word
- Average the word vectors in each message → **input for ML**

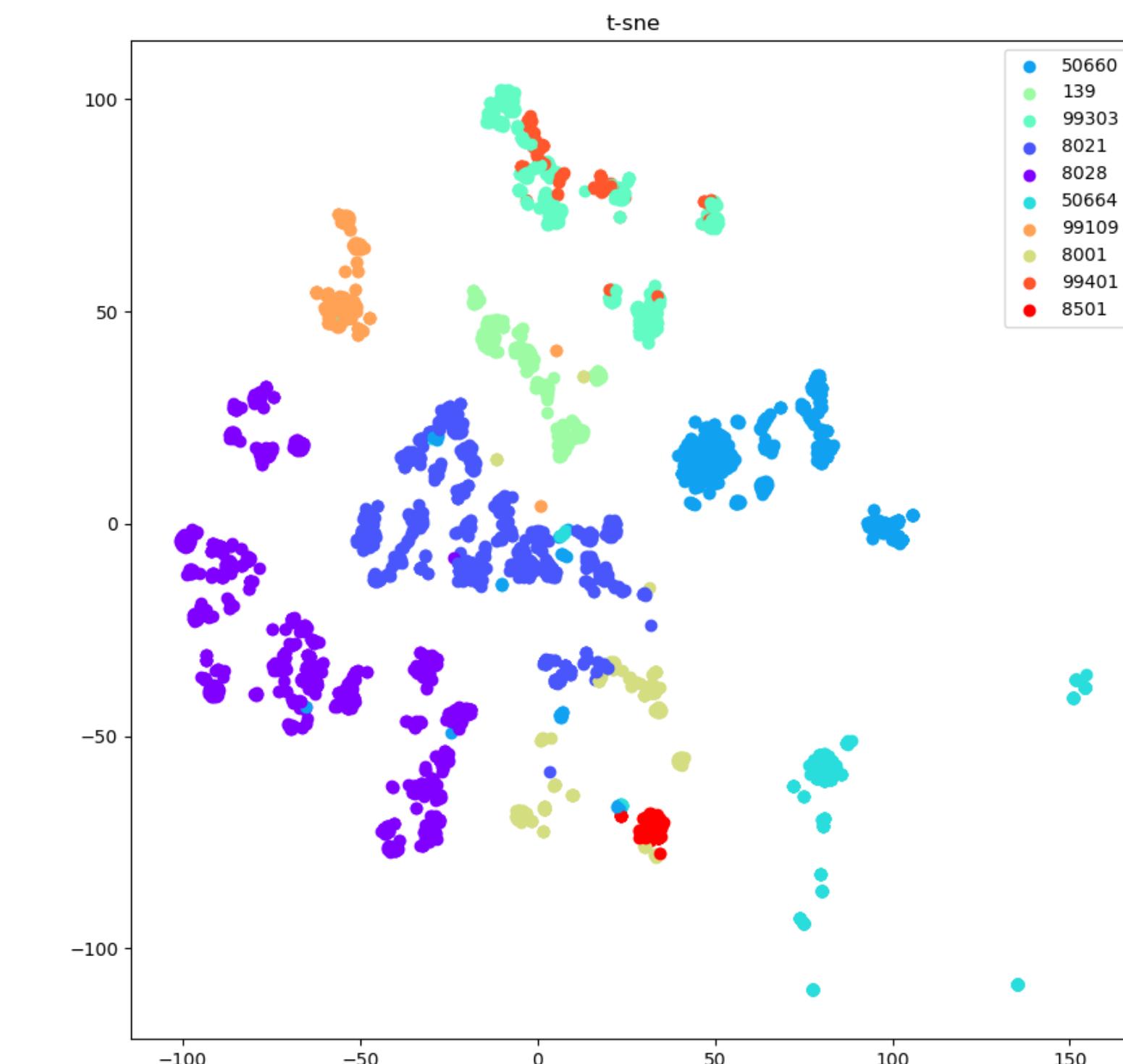
```
[u'An', u'exception', u'of', u'category', u'FileReadError', u'occurred', u'while', u'0', u'Calling', u'InputSource', u'getNextItemType', u'1', u'Reading', u'branch', u'EventAuxiliary', u'2', u'Calling', u'XrdFile', u'ready', u'3', u'Calling', u'XrdAdaptor', u'RequestManager', u'OpenHandler', u'open', u'Exception', u'Message', u'XrdCl', u'File', u'Open', u'name='root", u'tried=ccxrpli003.in2p3.fr', u'flags=0x10', u'permissions=0660', u'error', u'FATAL', u'Connection', u'error', u'errno=0', u'code=108', u'Additional', u'Info', u'Active', u'source', u'ccxrpli003.in2p3.fr:1095', u'site', u'T1_FR_CCIN2P3']
```



Average word vectors

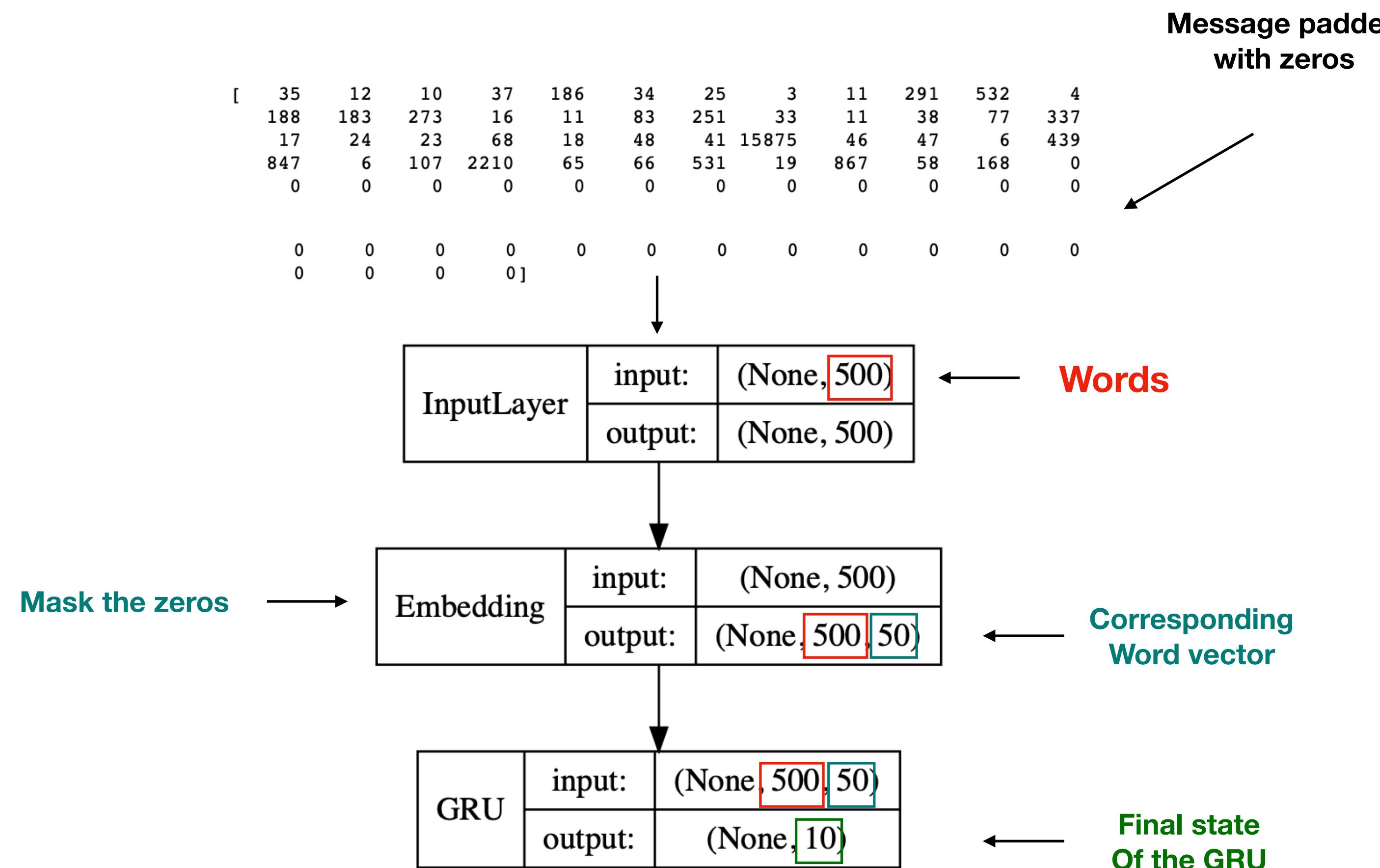
```
[-0.9386204  0.9671039  1.206119  -0.19411911 -0.23051228 -0.65913016
 0.00338563 -0.5044939  -0.02717912  0.09867108 -0.05116596 -0.5569398
 0.618001  -0.73519474  0.640754  -0.15959941  0.12680046  0.15426521
 0.42407024 -0.14709638  0.47534356 -0.18739735 -0.03758601  1.0197228
 0.06810962 -0.02504043 -0.11898965  0.11782035 -0.34788233  0.14381583
 0.67742926  0.43483204  0.39772424  0.10286205 -0.3774167  0.36935064
-0.1227726 -0.20352381 -0.22164229  0.10978855 -0.50076777  0.39308718
 0.1865182  0.33458158 -0.3371736  0.0493757  0.5454126 -0.94923306
-0.1352783 -0.34716246]
```

t-sne for 5000 averaged error message vectors

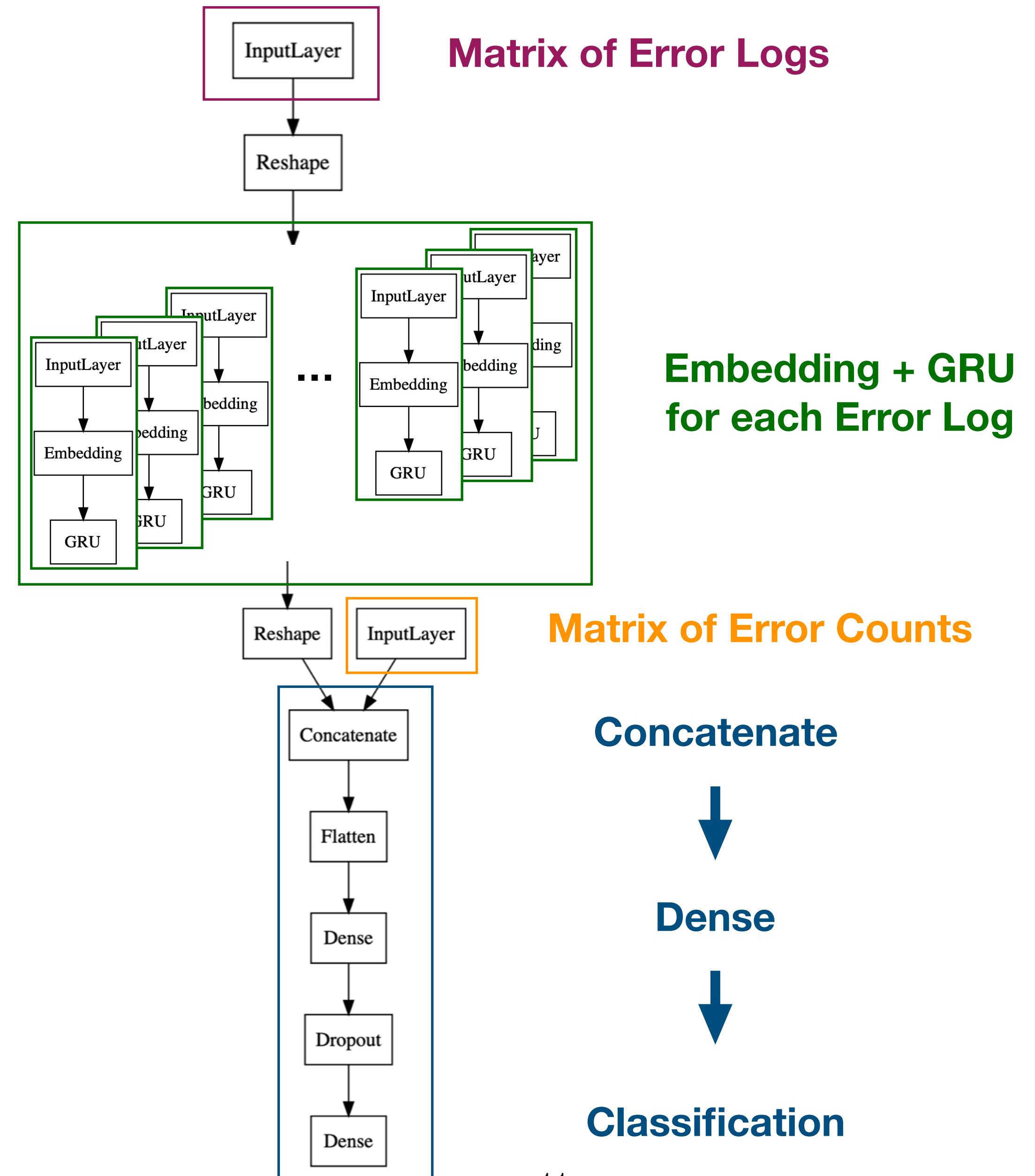


Supervised word embeddings: RNN

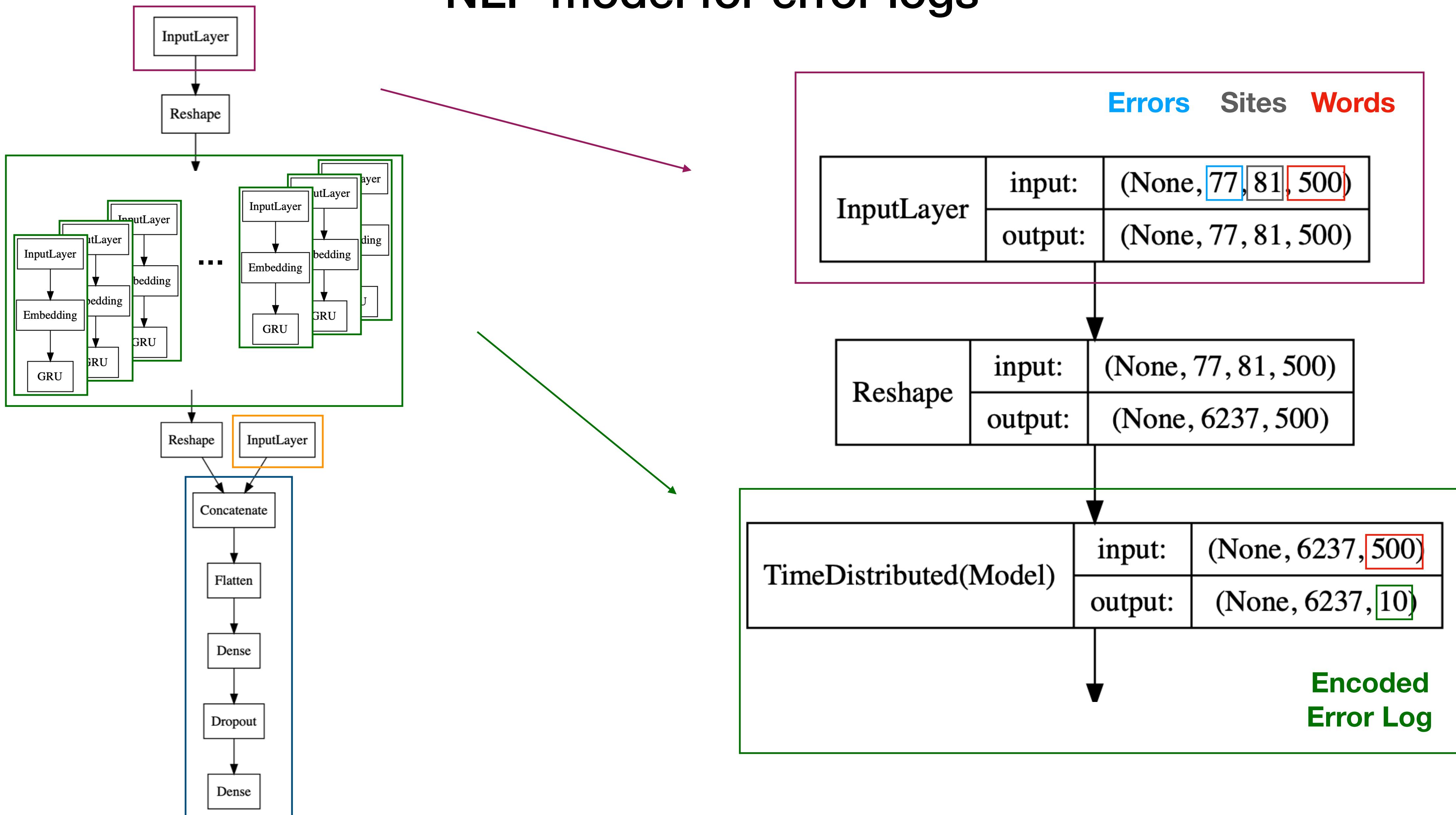
- Averaging does not exploit the **sequential nature of language** → use a RNN (LSTM / GRU)
- GRU **encodes** the embedding vectors of the error log words in a **vector**



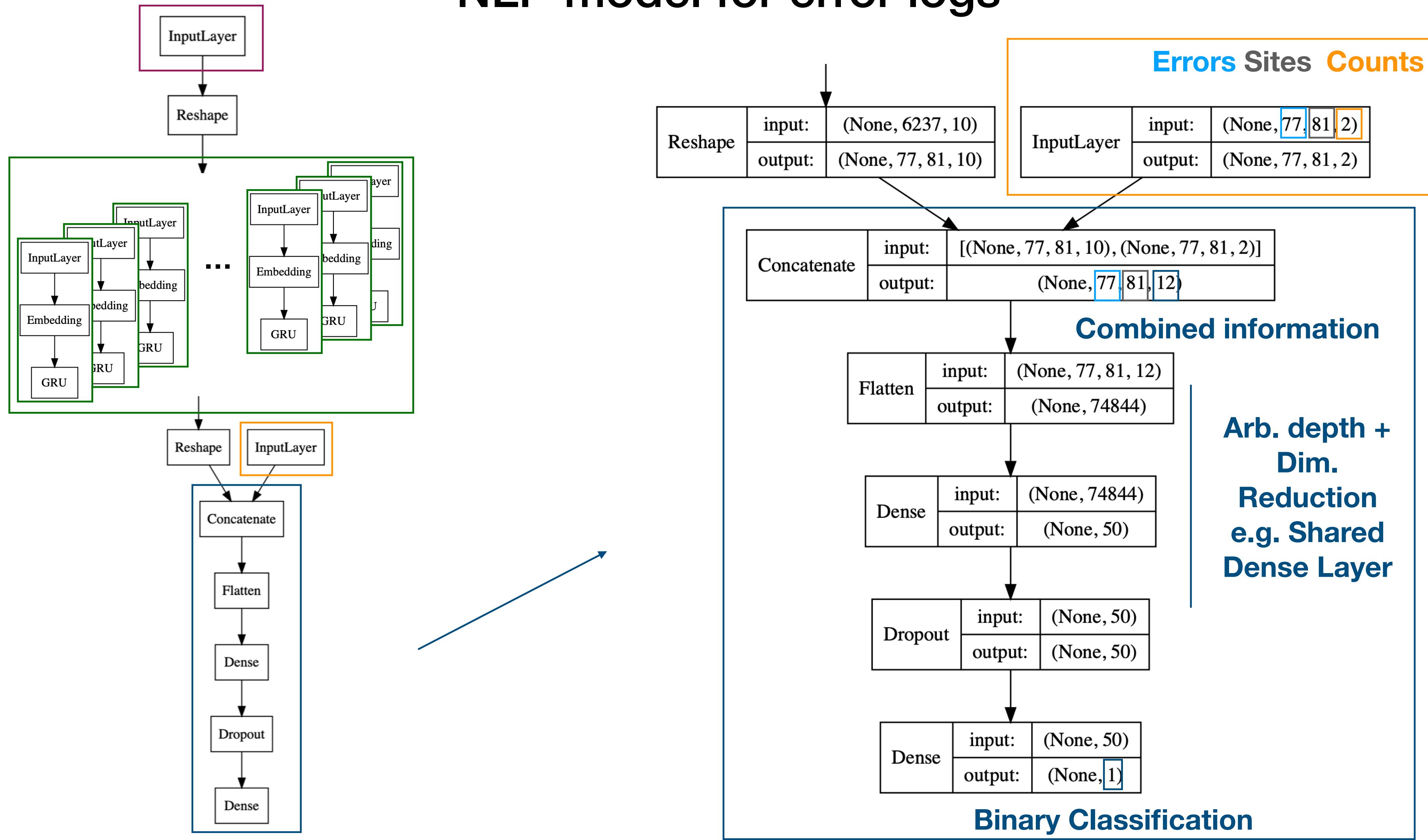
NLP model for error logs



NLP model for error logs



NLP model for error logs



Training and target

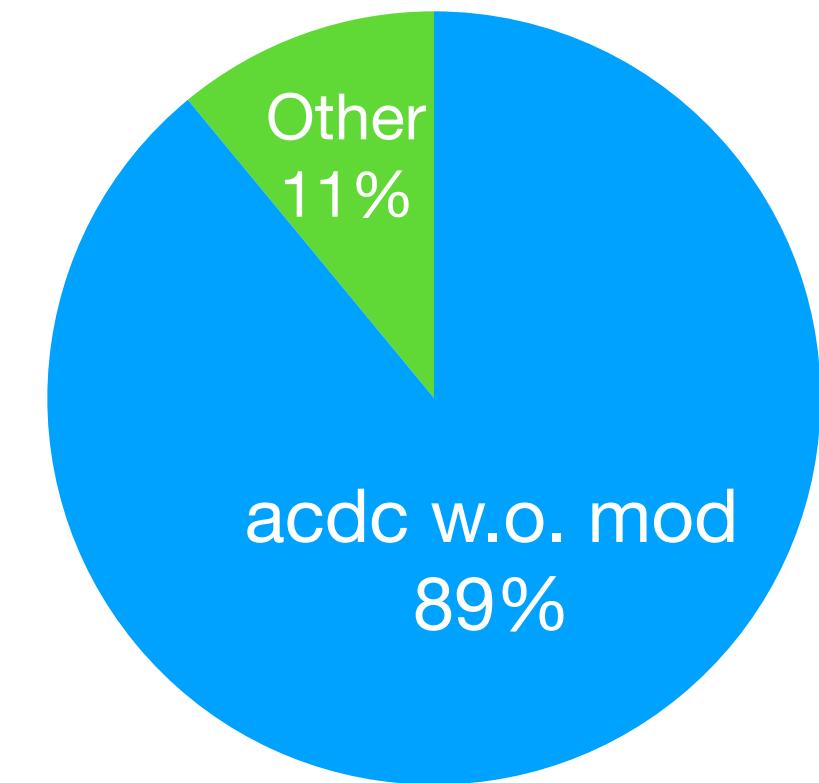
acdc w.o. modification

Retry only failed jobs
No modification of memory or splitting

vs. all other actions

- Simplest target → future: multiclass classification for more actions
- Small data + imbalanced classes → Cross-validation (3-Folds)
- Sparse input: batch size = 4 → **O(1h) / epoch** on a NVidia GeForce GTX 1080

~33.000 samples



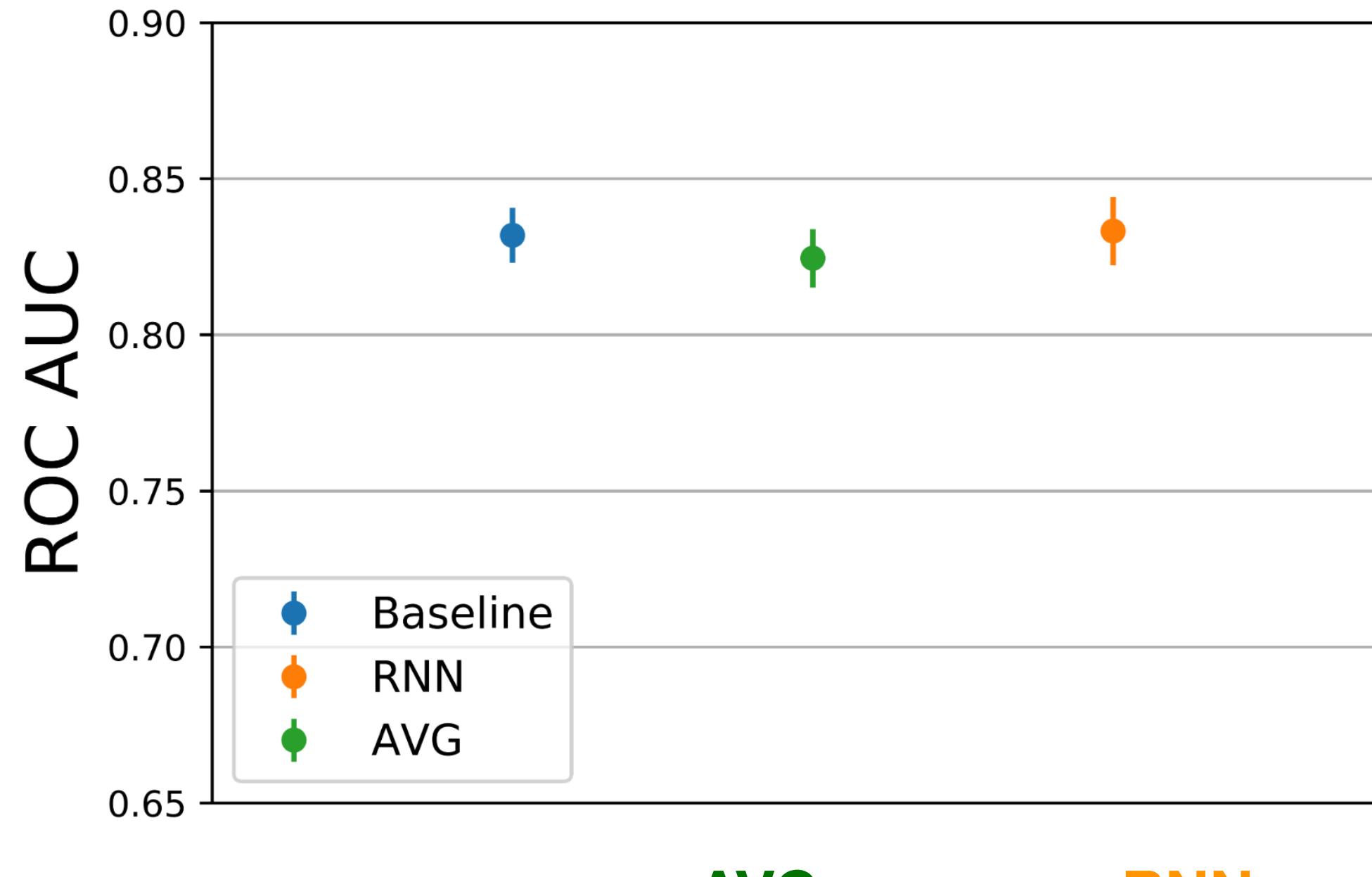
Hyper-parameter Optimization

- Bayesian Optimization with **scikit-optimize**
- 2-8 GPUs on **Caltech GPU cluster** with NVidia GeForce GTX 1080 / Titan X
- Experimental: distributed training with the **NNLO framework** → [Talk at CHEP](https://github.com/vlimant/NNLO)
<https://github.com/vlimant/NNLO>
- Experimental: Training models in parallel with **spark_sklearn** on **SWAN** with the help of CERN IT



Results

ROC AUC for acdc w.o. modification vs. other

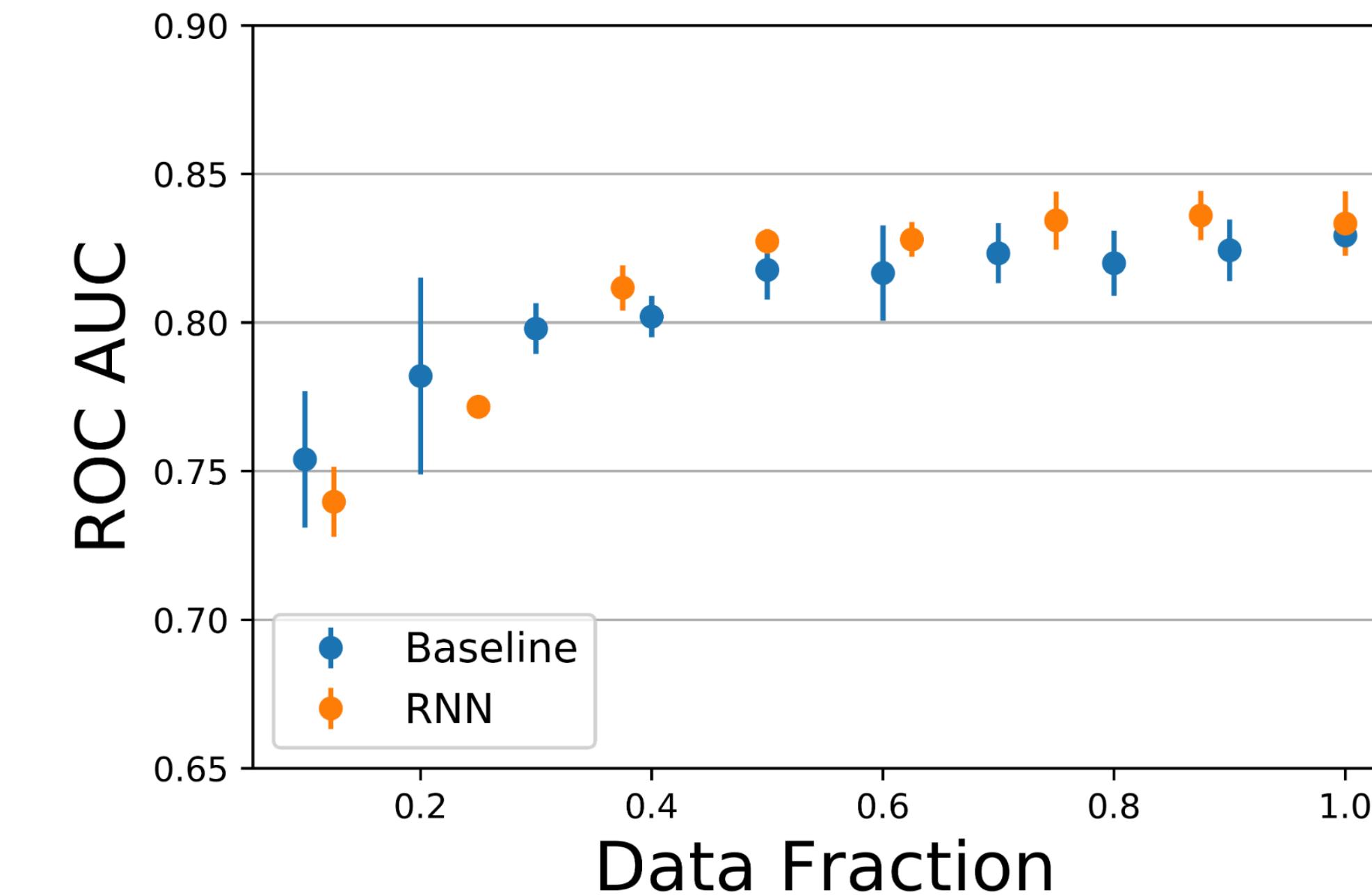


Baseline:
FF with counts only

AVG:
FF for averaged
w2v + counts

RNN:
RNN for embeddings
+ counts

ROC AUC as a function of the fraction of
the total data set used for training



→ Successful training of first complex NLP models

→ Similar results as baseline - performance improving with more data

→ Work ongoing → Full potential not yet exploited

Summary

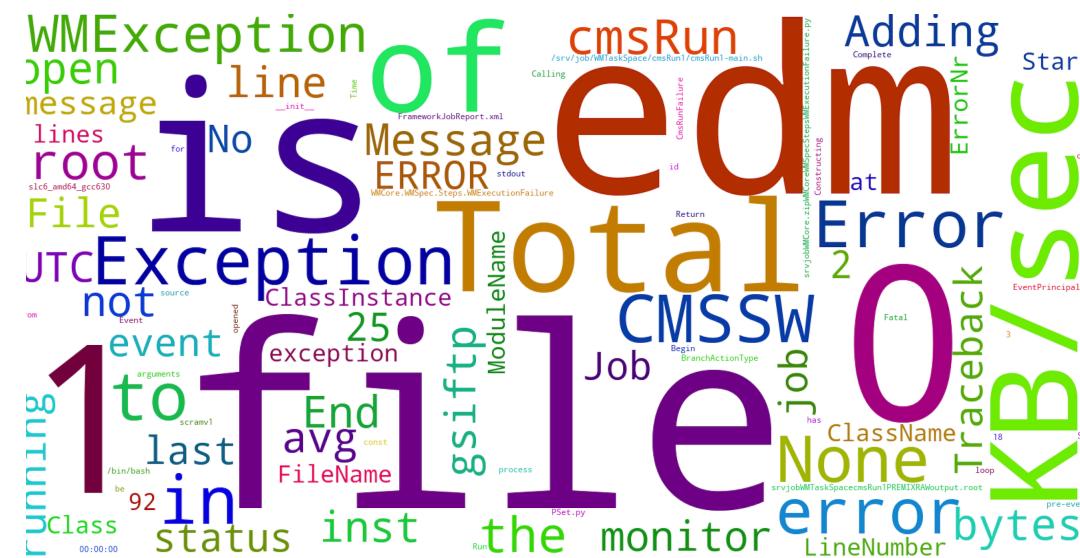
- Implementation of a **pipeline for DAQ and ML** of error logs using big data analysis tools → <https://github.com/llayer/AIErrorLogAnalysis>
- Development of a prototype **NLP model** in Keras

Outlook

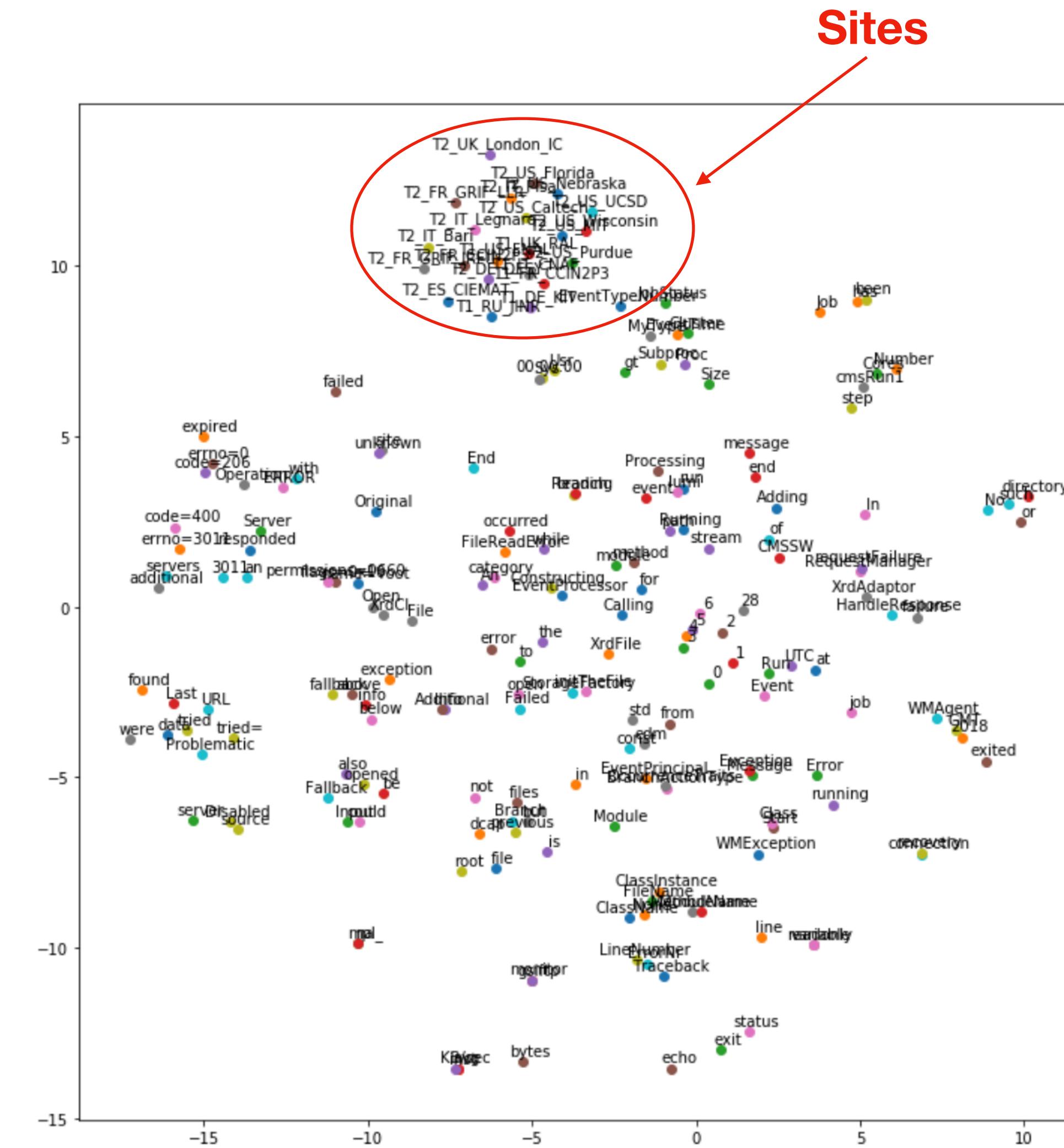
- **Further investigations** of data and model → Reimplement in **pytorch / tensorflow** and exploit the **NNLO framework**
- Get involved with the **NLP community** and learn from their experiences
- **RUCIO** (→ [Talk at CHEP](#)): Development of a **common system** to collect and categorize errors, provide shifters with actions and collect feedback on the suggestion → could also contain **error log snippets**

BACKUP

Word embeddings for log messages



- Vocabulary: ~ 30.000 words after filtering
- Unsupervised training with **word2vec**
- Standard parameters with embedding dimensions **10, 20, 50**
- Visualization: non-linear dimensionality reduction with the **t-sne algorithm** in 2D



Message representation: RNN + Attention

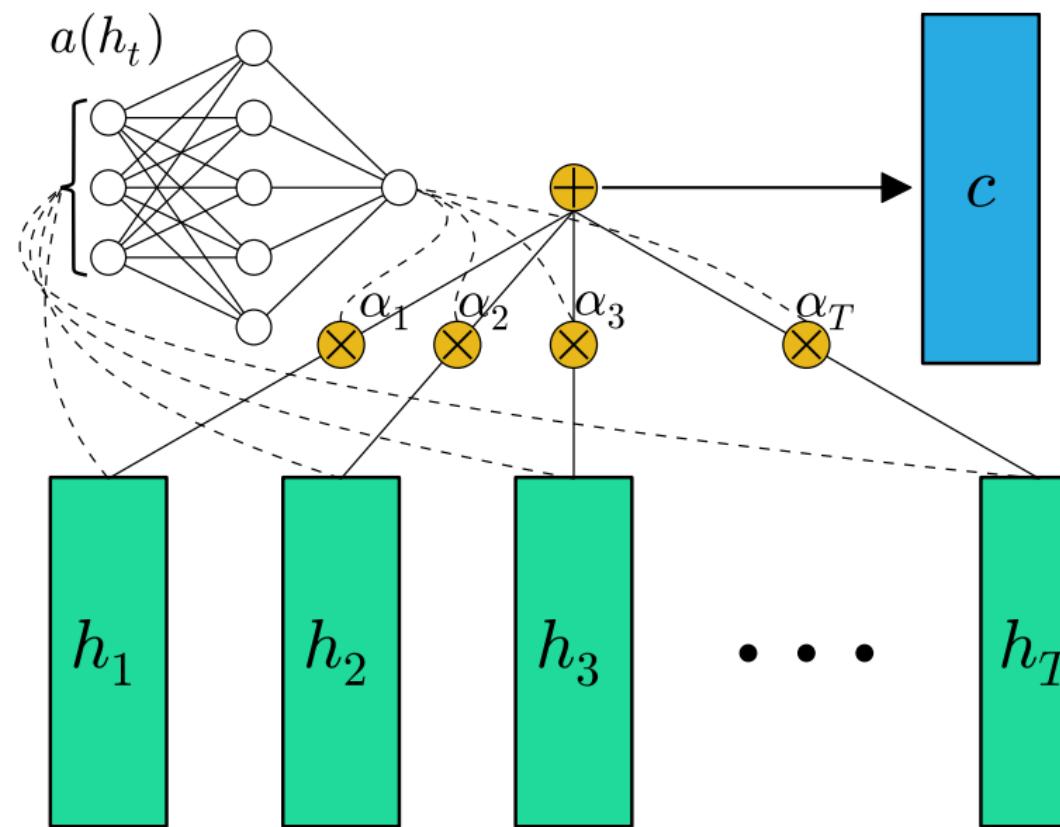
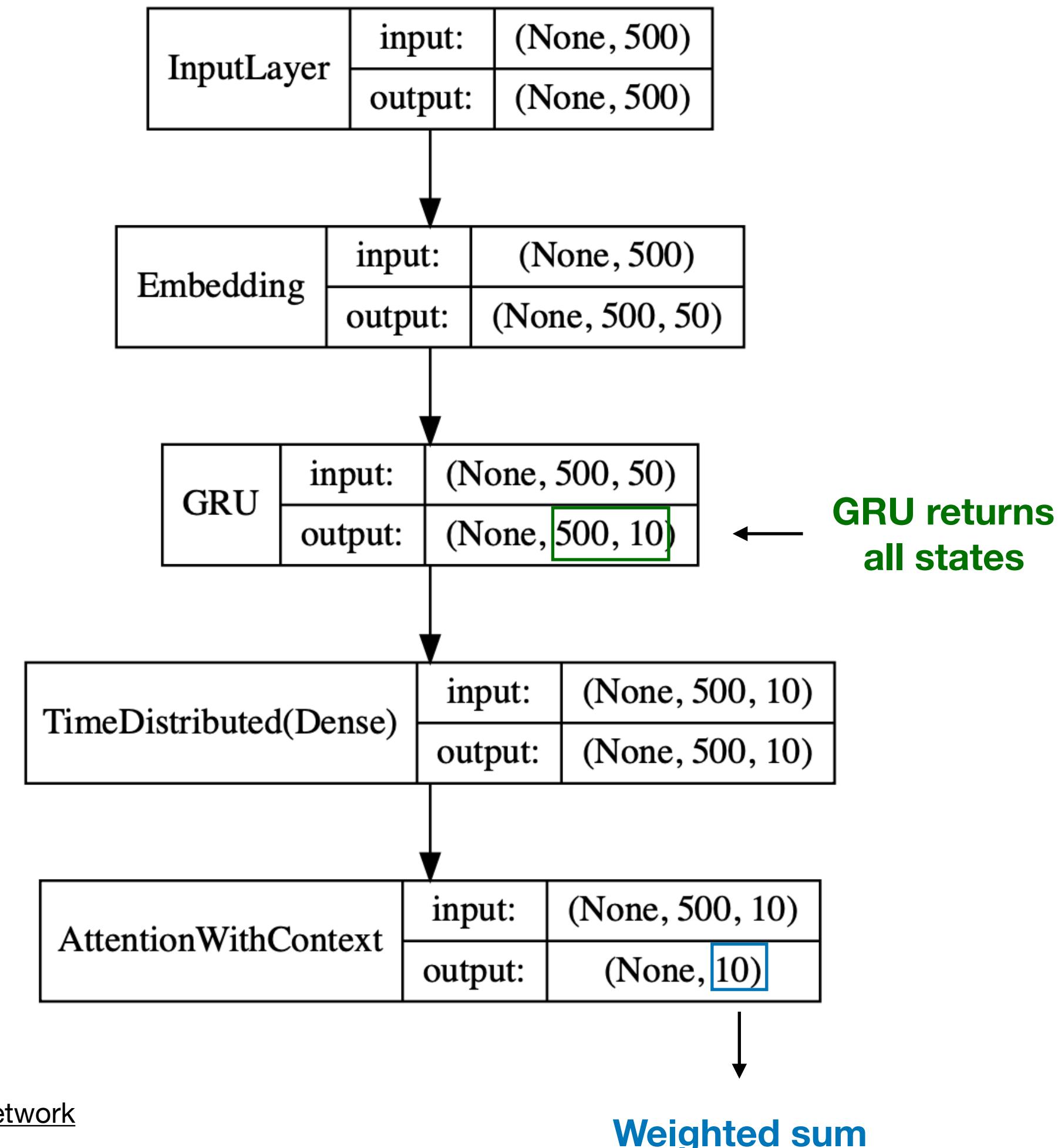
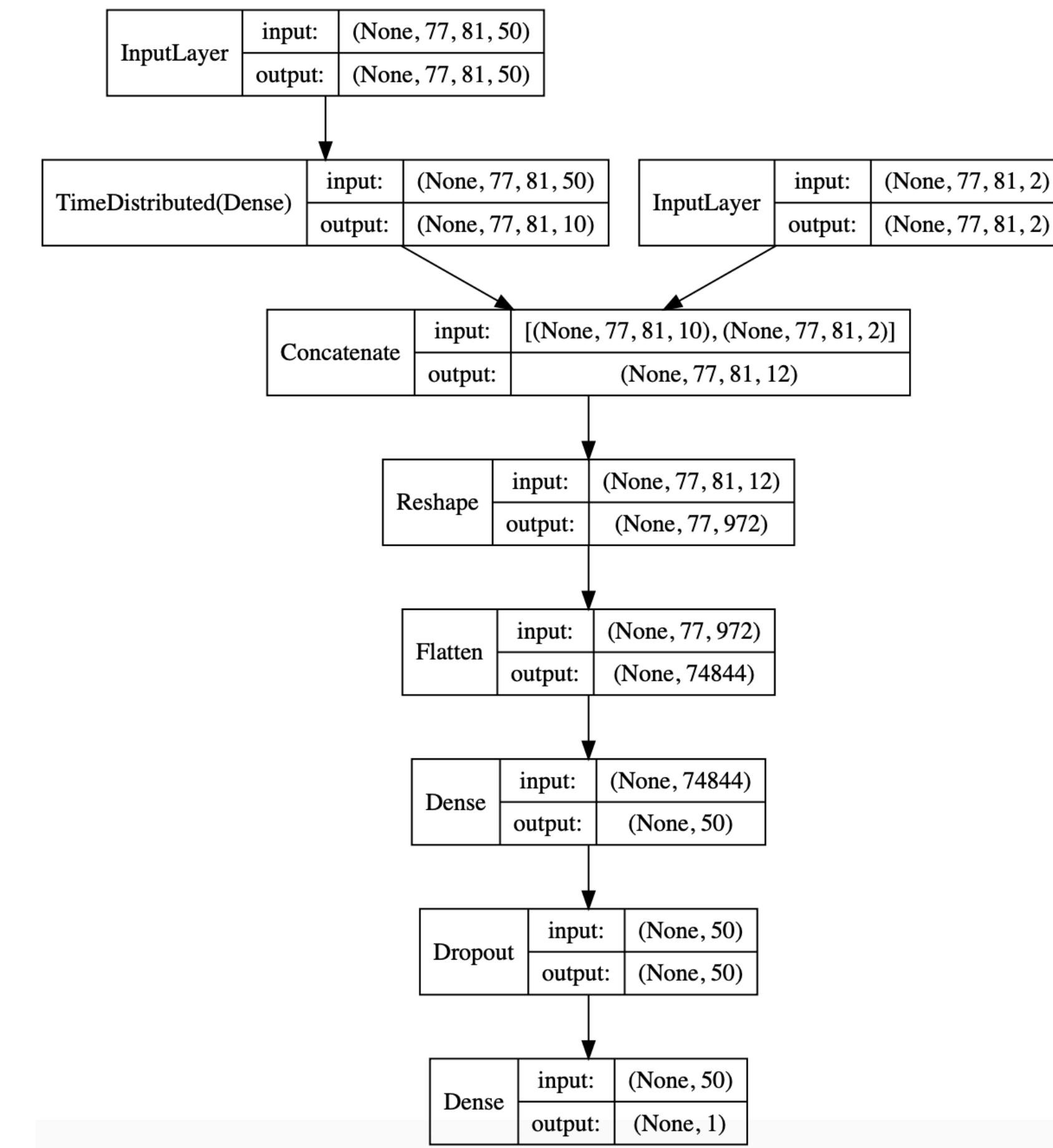


Figure 1: Schematic of our proposed “feed-forward” attention mechanism (cf. (Cho, 2015) Figure 1). Vectors in the hidden state sequence h_t are fed into the learnable function $a(h_t)$ to produce a probability vector α . The vector c is computed as a weighted average of h_t , with weighting given by α .

- Not all words are equally important
- Return **all RNN states** and pass through a **shallow network** to decide **weight** corresponding to each vector
- The **weighted sum** of each vector embodies the **meaning** of those vectors combined



Model for averaged w2v vectors



Best hyperparameters for the RNN model

Batch size = 4

Maximum number of words = 400

```
skot_dim_dimred = [
    Real(      low=1e-5, high=1e-3, prior='log-uniform', name='learning_rate'      ),
    Real(      low=1e-3, high=0.1, prior='log-uniform', name='dropout'      ),
    Integer(   low=5, high=32,                                name='embedding'      ),
    Integer(   low=2, high=20,                                name='rnn_units'      ),
    Integer(   low=10, high = 100,                            name = 'units_site'  ),
    Integer(   low=1,   high=5,                                name='dense_layers'  ),
    Integer(   low=10, high=50,                            name='dense_units'  ),
    Integer(   low=0,   high=1,                                name='encode_sites' ),
]
```

| | learning_rate | dropout | embedding | rnn_units | units_site | dense_layers | dense_units | encode_sites | result | std |
|----|---------------|---------|-----------|-----------|------------|--------------|-------------|--------------|-----------|----------|
| 14 | 0.000010 | 0.001 | 5.0 | 20.0 | 100.0 | 5.0 | 50.0 | 1.0 | -0.833228 | 0.010969 |
| 11 | 0.000037 | 0.100 | 20.0 | 14.0 | 62.0 | 2.0 | 25.0 | 1.0 | -0.831024 | 0.010876 |
| 2 | 0.000018 | 0.100 | 22.0 | 15.0 | 54.0 | 4.0 | 21.0 | 1.0 | -0.828022 | 0.010423 |
| 6 | 0.000019 | 0.100 | 21.0 | 15.0 | 54.0 | 3.0 | 21.0 | 1.0 | -0.826678 | 0.011404 |
| 13 | 0.000010 | 0.001 | 5.0 | 20.0 | 100.0 | 1.0 | 50.0 | 1.0 | -0.824910 | 0.012229 |

Actions

| action | splitting | Site list | memory | rate |
|------------------------------|-----------|--------------|---------|--------|
| ACDC | None | Modified | Not set | 87.53% |
| clone | None | Not modified | Not set | 4.61% |
| ACDC | None | Modified | > 20GB | 3.28% |
| ACDC | None | Modified | > 10GB | 1.12% |
| ACDC | None | Modified | < 10GB | 0.75% |
| ACDC | 10x | Modified | Not set | 0.74% |
| Other actions (27 more rows) | | | | < 2% |