SCRIPTS

Package directory hierarchy:

1. **Readme.pdf:** this file
2. **SCRIPTS:** a directory containing the integrated system
3. **NIST-data:** a directory containing all the processed data of the 1A, 1B, and 1S data sets
4. **Default-configs:** default system configuration files
5. **bin:** a directory containing the installation data
6. **install-step1.sh**: the installation script 1/2.
7. **install-step2.sh**: the installation script 2/2.

Please contact us if you want to try the system on our machines before installing it on your side.

1- System requirements

A server with UbuntuServer:16.04-LTS. The minimum server requirement must have the same features as our Azure server: Standard NV12 (12 vcpus, 112 GB memory, 2 M60 GPUs, 3TB SSD). And the recommended server should have the features as our second Azure server: Standard_NC24s_v2 (24 vcpus, 448 GB memory, 4 P100 GPUs, 4TB SSD)

2- Installation

The installation requires root access to the system. The installation script installs all the system prerequisites (java8, CUDA8, CudNN 5.1, docker, nvidia-docker, and the system docker images). This step is slow and may take couple of hours based on the available machine resources.

Notes:

- If any of these components is already installed or you want to install it manually, please comment out its corresponding part in the installation scripts (install-step1.sh and install-step2.sh).
- CudNN 5.1 is included under bin/cuda. If you prefer to manually download it before running the installation script, please replace the bin/cuda/lib64 and bin/cuda/include directories with your downloaded version

A. **Prerequisite installation:**
   [This step will take 20-60 minutes]
   mkdir -p /absolute_path/to/docker_installtion_directory

```
cd path/to/package_directory
sh install-step1.sh /absolute_path/to/docker_installtion_directory
```

After this step you need to exit your current session on your machine and login again. After that proceed to the next step

[This step will take many hours. So it is highly recommended to run it under "screen" or "tmux" utilities to avoid any interruption]

```
cd path/to/package_directory
sh install-step2.sh
```

B. **Add all users to the docker group:**
By default the installation script adds the user who installed the system to the docker group. To add more users:
sudo usermod -aG docker required_username

C. **Prepare the system settings:**
Edit the following flags in this file path/to/package_directory/SCRIPTS/config/scripts.properties:

a. Edit the scripts.corpora.path flag to point to the absolute path of the NIST-data directory. The default value is scripts.corpora.path=/storage3/data/NIST-data. You need to replace with the actual absolute path of NIST-data directory

b. script.config.gpu.number=number_of_attached_gpus
script.config.giga.ram.per.gpu=min_ram_size_of_attached_gpus
To find the attached GPUs info:

```
nvidia-smi
```

3- Execution

To run the preprocessing pipeline:

This mode processes the input data using the ASR, three MT systems, LangID, domainID, Morphological analyzer, Query analyzer, and the MT-postprocessing.

```
cd path/to/package_directory/SCRIPTS
java -jar scripts-release-20190327-0.1.jar path/to/input_config_file.json
path/to/temp_directory
```

To run the indexing pipeline:

This mode indexes all the preprocessed data from the preprocessing step.

```
cd path/to/package_directory/SCRIPTS
java -jar scripts-release-20190327-0.1.jar path/to/input_config_file.json path/to/temp_directory
```

To run the E2E pipeline:

This mode is the query-time mode. It retrieves the relevant documents and their summaries for the input list of queries.

```
cd path/to/package_directory/SCRIPTS
java -jar scripts-release-20190327-0.1.jar path/to/input_config_file.json path/to/output_directory path/to/temp_directory
```

Please refer to the next section for the details of the input configuration files.

Notes:

- All paths must be absolute. Relative paths can make errors
- This version of the pipeline works only with 1A, 1B, and 1S languages
- You can run as many instances of the pipeline as you want until you maximize the available resources on your machine. You will not have any conflict between the running instances. But you need to provide a different temp directory to the pipeline to be used as a work-directory for each instance you want to run. You can not use the same temp directory for more than one instance.
- 1A, 1B, and 1S languages are completely processed. You don't need to call the preprocessing or the indexing pipeline unless there are more data required to be indexed
- The queries must go through the pre-processing pipeline before you will be able to search them in the E2E pipeline. All the query lists of the three languages are already processed. If you have more queries, then need to be analyzed first using the preprocessing mode.

4- Input Configuration files

I. Preprocessing pipeline configuration file format

The data that is required to be processed must be placed in the NIST-data directory in its corresponding language. The input configuration file is a json file that specifies the location of the NIST-data directory and a list of all the datasets required to be

processed. You can add datasets from one or more of any of the three supported languages {tl, sw, so}. But please note that you need at least one GPU per language. So if you are going to process datasets from the three languages, you need at least three GPUs.

Note: the "root_absolute_path" key needs to point to the absolute path of the parent directory of the NIST-data directory

Ex:

```
{
 "corpora": {
  "root_absolute_path": "/path/to/parent_directory_of_NIST-data",
  "relative_path": "NIST-data",
  "queries": [
   {
    "query_name": "1B/IARPA_MATERIAL_BASE-1B/QUERY1/query_list.tsv",
    "language": "tl"
   },
   {
    "query_name": "1B/IARPA_MATERIAL_BASE-1B/QUERY2/query_list.tsv",
    "language": "tl"
   },
   {
    "query_name": "1B/IARPA_MATERIAL_BASE-1B/QUERY3/query_list.tsv",
    "language": "tl"
   }
  ],
  "collections": [
   {
    "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/ANALYSIS1",
    "source_location": "text/src",
    "meta_data_location": "text/metadata/metadata.tsv",
    "type": "text",
    "language": "tl"
   },
   {
    "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/ANALYSIS1",
    "source_location": "audio/src",
    "meta_data_location": "audio/metadata/metadata.tsv",
    "type": "audio",
    "language": "tl"
   },
   {
    "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/ANALYSIS2",
    "source_location": "text/src",
    "meta_data_location": "text/metadata/metadata.tsv",
    "type": "text",
```

```
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/ANALYSIS2",
  "source_location": "audio/src",
  "meta_data_location": "audio/metadata/metadata.tsv",
  "type": "audio",
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/DEV",
  "source_location": "text/src",
  "meta_data_location": "text/metadata/metadata.tsv",
  "type": "text",
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/DEV",
  "source_location": "audio/src",
  "meta_data_location": "audio/metadata/metadata.tsv",
  "type": "audio",
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/EVAL1",
  "source_location": "text/src",
  "meta_data_location": "text/metadata/metadata.tsv",
  "type": "text",
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/EVAL1",
  "source_location": "audio/src",
  "meta_data_location": "audio/metadata/metadata.tsv",
  "type": "audio",
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/EVAL2",
  "source_location": "text/src",
  "meta_data_location": "text/metadata/metadata.tsv",
  "type": "text",
  "language": "tl"
 },
 {
  "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/EVAL2",
  "source_location": "audio/src",
  "meta_data_location": "audio/metadata/metadata.tsv",
```

```
      "type": "audio",
      "language": "tl"
     },
     {
      "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/EVAL3",
      "source_location": "text/src",
      "meta_data_location": "text/metadata/metadata.tsv",
      "type": "text",
      "language": "tl"
     },
     {
      "corpus_name": "1B/IARPA_MATERIAL_BASE-1B/EVAL3",
      "source_location": "audio/src",
      "meta_data_location": "audio/metadata/metadata.tsv",
      "type": "audio",
      "language": "tl"
     }
    ]
   }
}
```


II. Indexing pipeline configuration file format

A fixed configuration file can be found here:
Default-configs/indexing-config.json
This file contains a list of all datasets in the three supported languages. It has an extra section to specify the parameters of indexer:


```
"indexer": {
    "indri_parameters": [
       "path/to/package_directory/Default-configs/indexer-parameters/indexing_para
       ms_porter_stemmer_v2.0.txt",
       "path/to/package_directory/Default-configs/indexer-parameters/indexing_para
       ms_v1.0.txt"
    ]
   }
```


You also need to update the "root_absolute_path" key to point to the "/path/to/parent_directory_of_NIST-data" as shown in the pre-processing section. Any new processed data, need to be added to the current list of the Default-configs/indexing-config.json

III. E2E pipeline configuration file

Configuration files are available for the following query/document pairs for each language:

- Q1/DEV

- Q1/A1A2
- Q2Q3/E1E2E3

One configuration is available for each Tagalog and Swahili and two different configurations are available for Somali (system combination and single system).

The configurations for Swahili are based on the best performing configurations submitted to the August evaluation -- a Borda count-based combination of SMT and NMT document translation systems in which some of the systems were updated and cutoff was changed to better correspond to Beta=40. The fixed cutoff for the DEV set is set to 2, the cutoff on A1A2 is set to 5 and the cutoff on E1E2E3 is set to 50.

- Default-configs/swahili-Q1-DEV-combination.json
- Default-configs/swahili-Q1-A1A2-combination.json
- Default-configs/swahili-Q2Q3-E1E2E3-combination.json

The configurations for Tagalog are based on the best performing configurations submitted to the August evaluation -- a Borda count-based combination of SMT document translation and PSQ systems in which some of the systems were updated and cutoff was changed to better correspond to Beta=40. The fixed cutoff for the DEV set is set to 5, the cutoff on A1A2 is set to 3 and the cutoff on E1E2E3 is set to 50.

- Default-configs/tagalog-Q1-DEV-combination.json
- Default-configs/tagalog-Q1-A1A2-combination.json
- Default-configs/tagalog-Q2Q3-E1E2E3-combination.json

The system combination configurations for Somali are based on the best performing system submitted to the January evaluation with some small system updates. The configuration uses STO combination of two NMT and one SMT systems for document translation and a PSQ system, on which STO cutoff is applied.

- Default-configs/somali-Q1-DEV-combination.json
- Default-configs/somali-Q1-A1A2-combination.json

- Default-configs/somali-Q2Q3-E1E2E3-combination.json

The Somali single best system is using NMT document translation approach. This system was with some previous versions of the components submitted to the January evaluation. The fixed cutoff for the DEV and A1A2 sets is set to 2, and the cutoff on E1E2E3 is set to 50.

- Default-configs/somali-Q1-DEV-single-system.json
- Default-configs/somali-Q1-A1A2-single-system.json
- Default-configs/somali-Q2Q3-E1E2E3-single-system.json

To be able to run the configuration file in the installed environment, the query_list_path in the query_processor needs to contain the full path to the queries in json format created by the query analyzer component in the preprocessing pipeline.
Ex: in the somali-Q2Q3-E1E2E3, the blue part below needs to be replaced with the absolute path of the NIST-data directory.

```
"query_processor": {
            "version": "query-analyzer-umd:v10.3",
            "query_list_path": [
                "/path/to/NIST-data/1S/IARPA_MATERIAL_BASE-1S/query_store/
                query-analyzer-umd-v10.3/QUERY2/",
                "/path/to/NIST-data/1S/IARPA_MATERIAL_BASE-1S/query_store/
                query-analyzer-umd-v10.3/QUERY3/"
            ],
            "target_language": "so"
}
```