Master Thesis update

Voice input based story generation

17.11.2020 by:

Rohit Keshav Bewoor (11011831)

Big Data and Business Analytics 2018-20 batch
SRH Hochschule Heidelberg

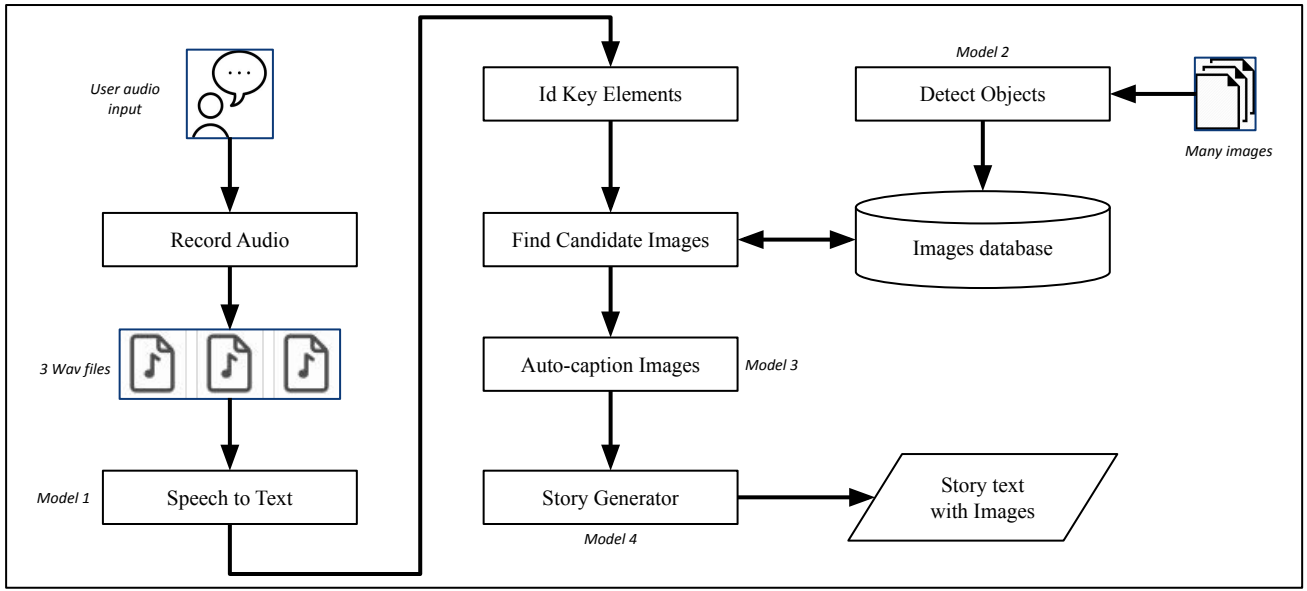
SRH HOCHSCHULE HEIDELBERG
Intelligence in Learning

Overall Implementation Approach

STAATLICH ANERKANNTE HOCHSCHULE

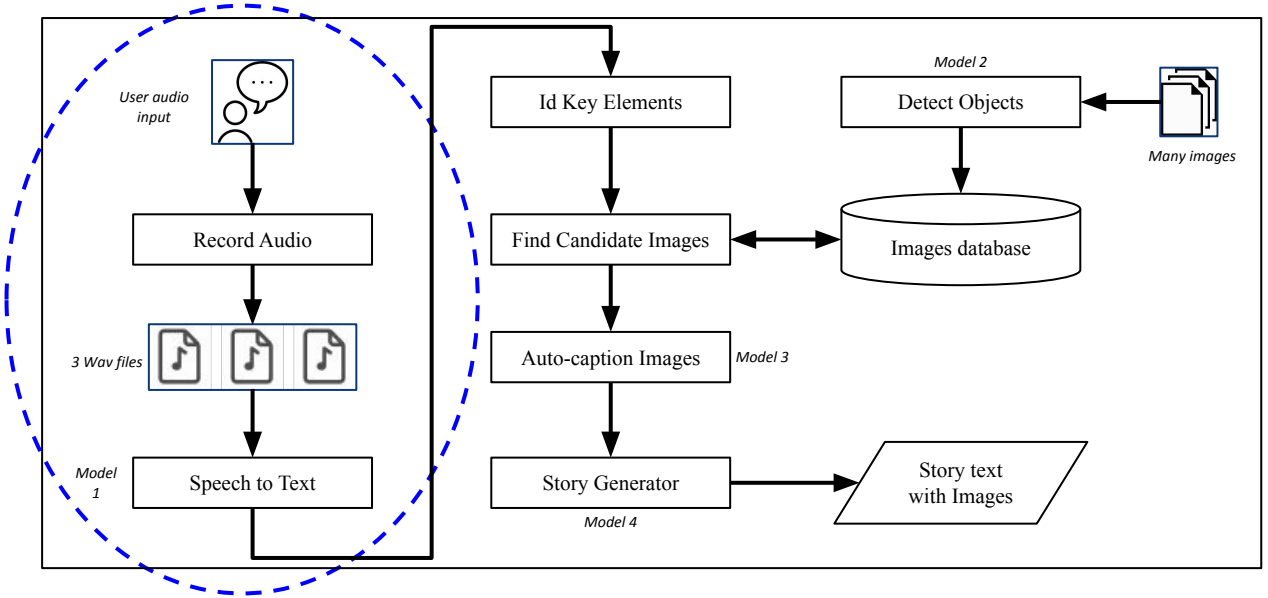
- **Goal**: Accept a voice input from user and use it to create a story with images selected from a database.
  Use different neural networks for specific tasks and pass data between them.

- Model 1: Speech-to-Text (STT)
- Model 2: Object Detector
- Model 3: Image Auto-caption
- Model 4: Story Generator
- All implementation in Python 3
- Used a GUI where possible



STAATLICH
ANERKANNTE
HOCHSCHULE

Stage 1: Capture audio & Speech-to-Text



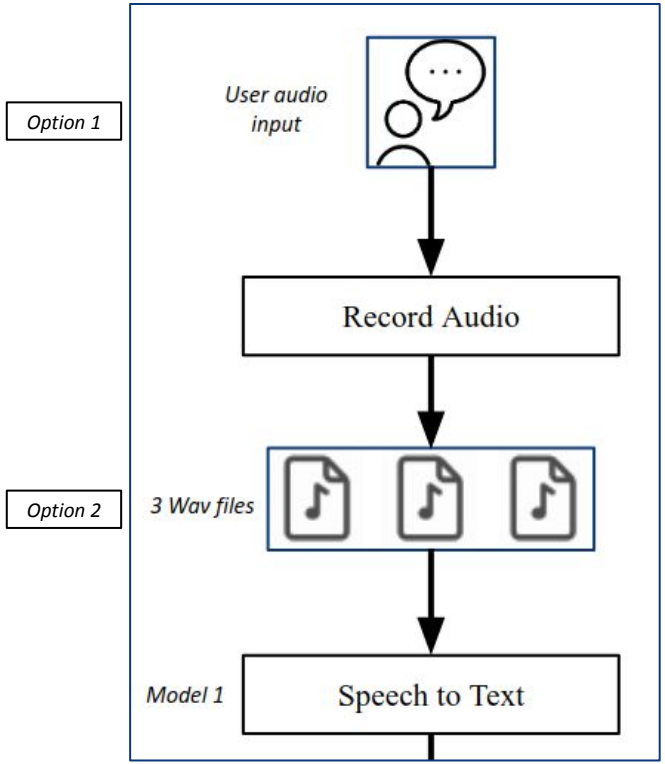
Mic Input (optional stage) - Approach

STAATLICH ANERKANNTE HOCHSCHULE

- **Goal**: Record user speaking three sentences and save them as Wav files.

- User can start process via two methods:
  - > ***Option 1***: Record three audio files, then start Speech-to-Text processing
  - > ***Option 2***: No recording allowed, directly process pre-recorded audio files

- GUI implemented to Play / Record / Do STT inference. Python modules used:
  - > Tkinter for GUI
  - > Pyaudio to play/ record Wav files



How to specify with / without Recording run

STAATLICH ANERKANNTE HOCHSCHULE

- Command line parameter “micYorN” decides run type:
  - > “Y” or “y” for WITH Recording required
  - > “N” or “n” for NO Recording. Directly process pre-recorded Wav files

```
python3 comb_functional_mic_stt_idKey_queryNeo_imgSelect_imgCap_7D-WIP.py -micYorN "Y" -switchere
"/home/rohit/PyWDUbuntu/thesis/audio/wavs/fromMic/" -file4STT "/home/rohit/PyWDUbuntu/thesis/combined_execution/SttTranscribe/mic_input_wavfiles_1.txt"
-opfileallposinfo "/home/rohit/PyWDUbuntu/thesis/combined_execution/IdElements/all_words_pos_info_1.txt" -logfileloc
"./LOG_comb_functional_mic_stt_idKey_queryNeo_imgSelect_imgCap_7D-WIP.LOG"
```

```
python3 comb_functional_mic_stt_idKey_queryNeo_imgSelect_imgCap_7D-WIP.py -micYorN "N" -switchere
"/home/rohit/PyWDUbuntu/thesis/audio/wavs/fromMic/" -file4STT "/home/rohit/PyWDUbuntu/thesis/combined_execution/SttTranscribe/mic_input_wavfiles_1.txt"
-opfileallposinfo "/home/rohit/PyWDUbuntu/thesis/combined_execution/IdElements/all_words_pos_info_1.txt" -logfileloc
"./LOG_comb_functional_mic_stt_idKey_queryNeo_imgSelect_imgCap_7D-WIP.LOG"
```

Speech to Text - Goal and Approach

STAATLICH
ANERKANNTE
HOCHSCHULE

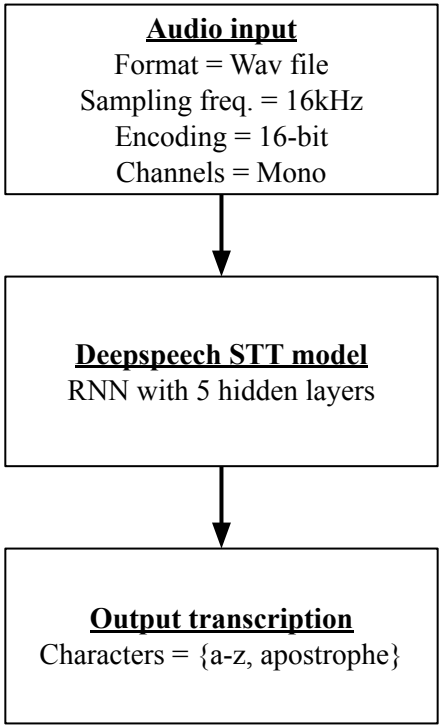
- **Goal**: Process voice input and output a string for the transcription
  - > Currently, system accepts one pre-recorded wav file per input sentence.
  - > Exactly three wav files expected for processing
- Choosing approach:

Microsoft Azure 	**Mozilla Deepspeech** 
Good accuracy but paid service	Open source implementation by Mozilla foundation
Had used earlier so familiar	Model available that is pre-trained on huge corpus of English sentences

More about Deepspeech

STAATLICH ANERKANNTE HOCHSCHULE

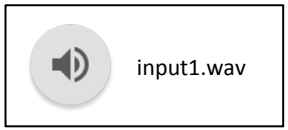
- Used version 0.7.3 (released June 2020)
- Based on 2014 paper “Deep Speech: Scaling up end-to-end speech recognition” by Baidu research team: https://arxiv.org/abs/1412.5567
- ***Implications***: Cannot display words with punctuations, end of sentence period.
- More information...
  - > Project links for this release:
    https://deepspeech.readthedocs.io/en/v0.7.3/
    https://deepspeech.readthedocs.io/en/v0.7.3/DeepSpeech.html
  - > Pypi project link:
    https://pypi.org/project/deepspeech/0.7.3/

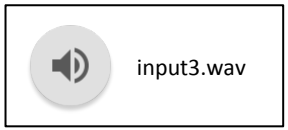


Deepspeech - testing the waters !

STAATLICH ANERKANNTE HOCHSCHULE

- Tested on my own voice with three audio files

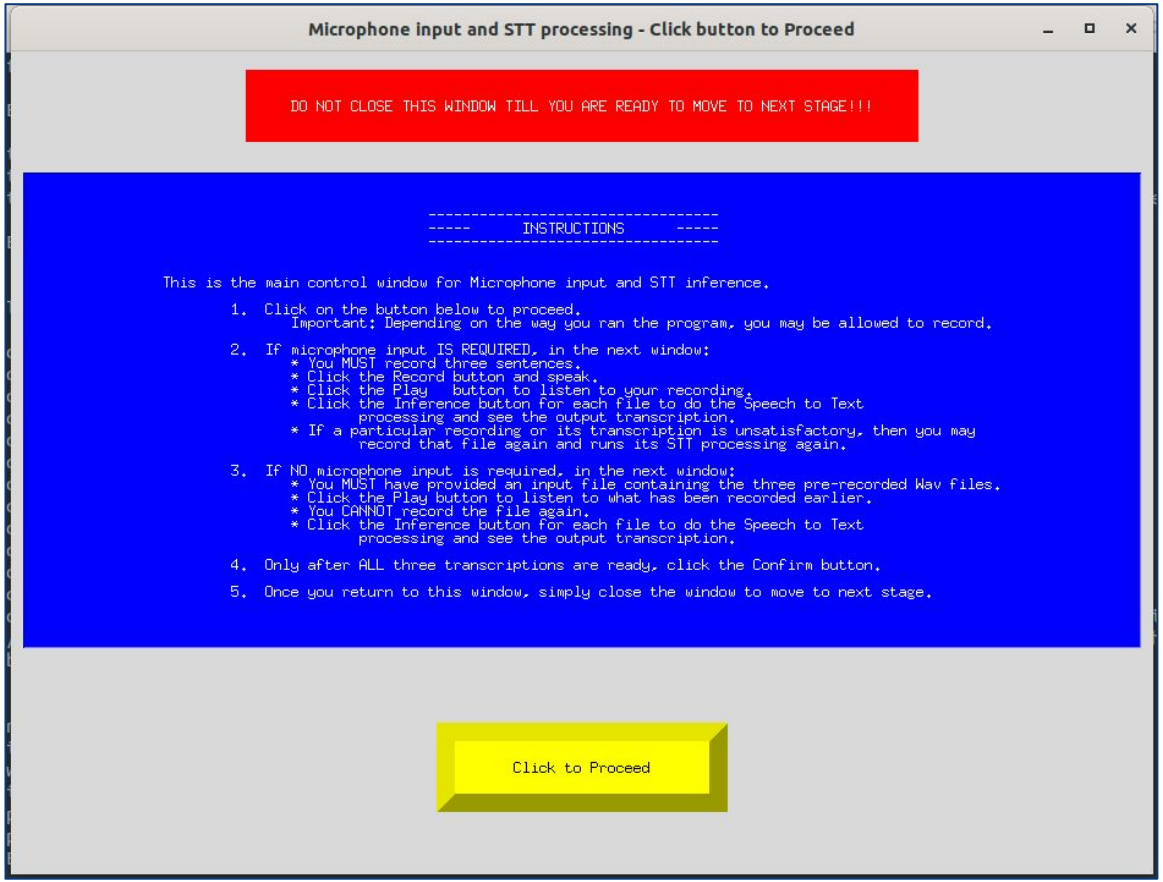




S.N.	What I said in recording	Model inference	Comments
1	*Make me a story about persons sitting at a table. They are playing cards.*	*me me a tory about persons sitting at table the blanchards*	• Cannot distinguish one sentence from other. • Output words in lowercase, no punctuations • Not always accurate- b***ut generally acceptable output***
2	*I want a story about a car on the road. A child plays with a toy.*	*i want a story about a car on the road a child plays with a toy*	
3	*Generate a story about persons walking on the street. A truck is on the road.*	*generate a story about persons walking on the street a truck is on the road*	

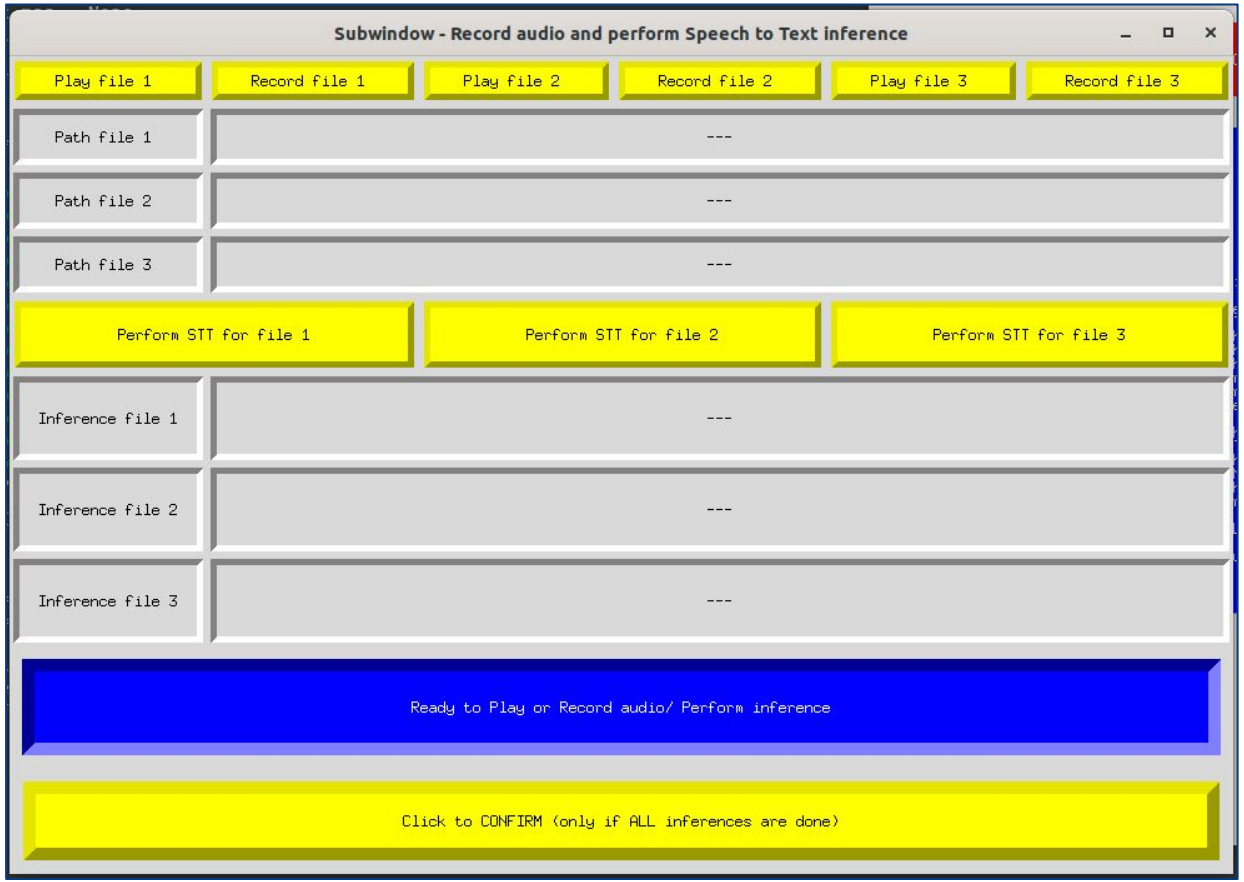
- ***Implications: User must speak only one sentence per input audio file***

Main instructions window



- Main graphical user interface window
- User to click yellow button to proceed

Recording required - Initial window



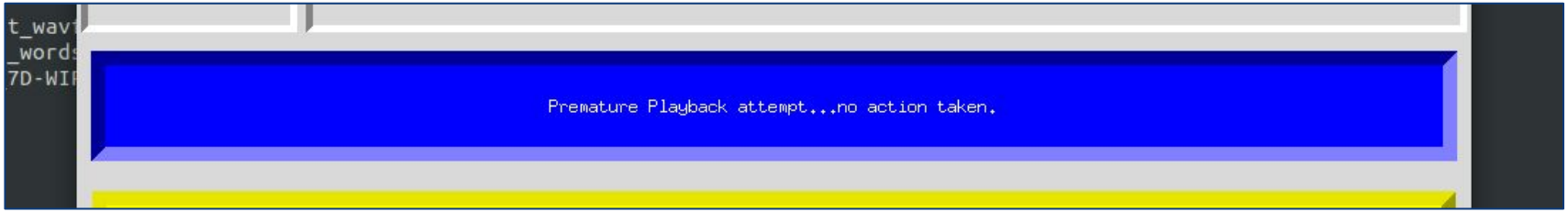
- Buttons to Play and Record
- Wav file paths
- Buttons for STT inference
- Inference outputs
- Message box as aid for user
- Final confirmation button

STAATLICH ANERKANNTE HOCHSCHULE

User Messages (blue box)

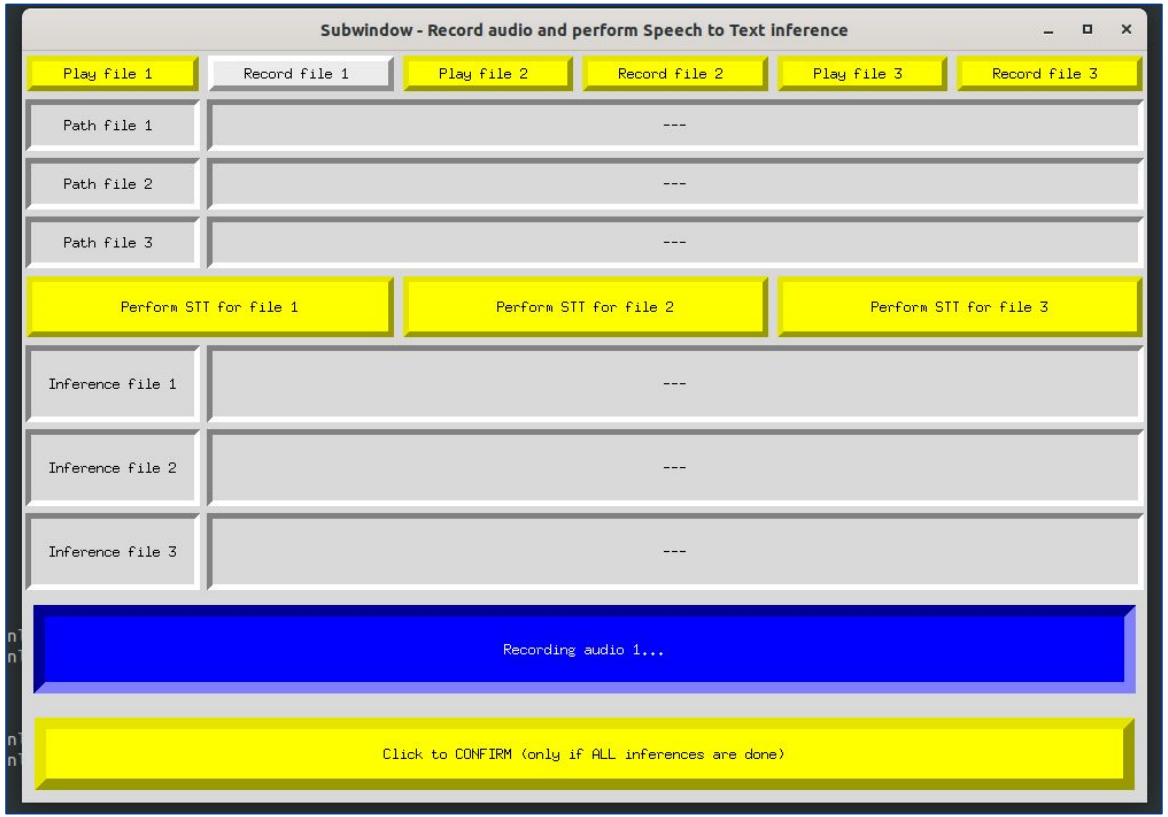
STAATLICH ANERKANNTE HOCHSCHULE

- Messages displayed depending on the situation:
  - > “Ready to Play or Record audio/ Perform inference” - initial message if running with recording
  - > “Ready to Play audio/ Perform inference - No Recording allowed” - initial message if running without recording
  - > “Playing audio” - audio playback in progress
  - > “Playback successful” - audio file played successfully
  - > “Premature Playback attempt...no action taken” - attempted playing a file not yet created
  - > “Recording audio” - audio recording in progress
  - > “Recording successful” - audio file recorded successfully
  - > “Performing inference” - STT processing in progress
  - > “Inference successful” - after STT processing is done
  - > “Premature Inference attempt...no action taken” - STT processing attempted before Wav file is created
  - > “Premature Confirmation attempt...no action taken” - some inferences are pending



User records one sentence

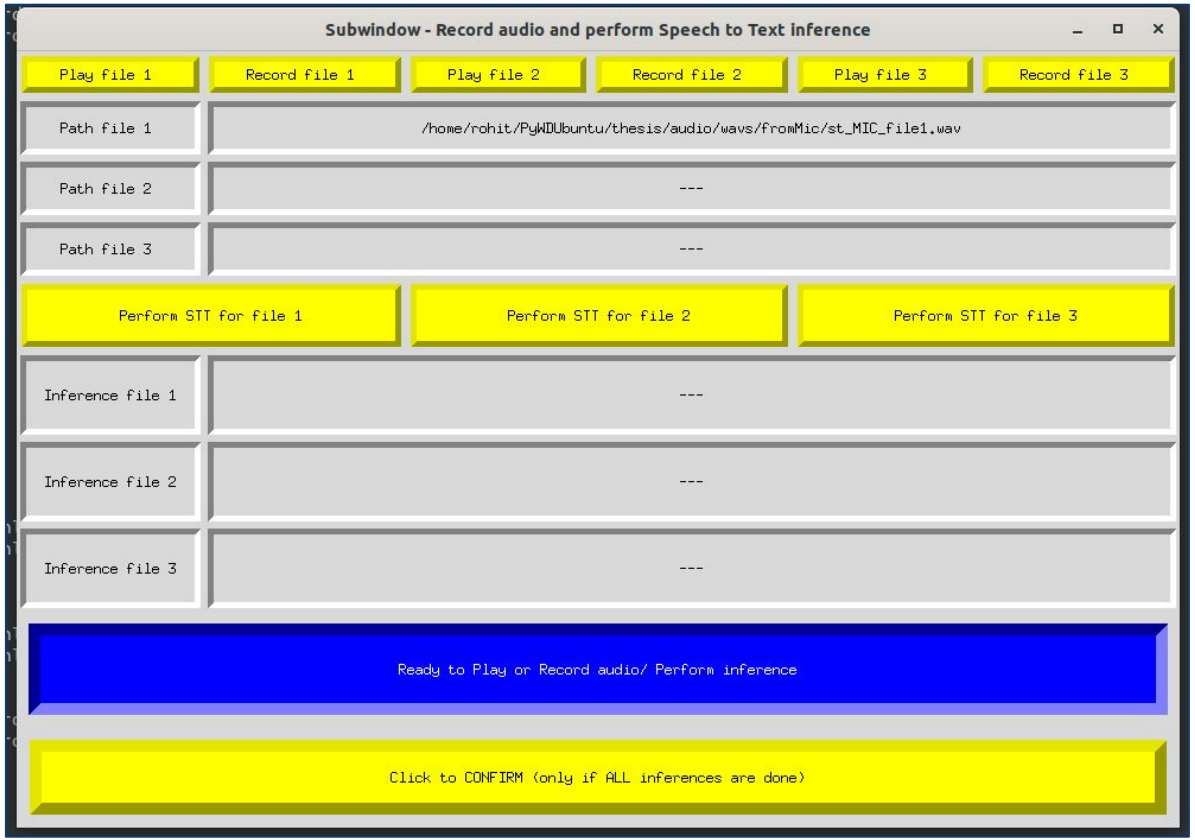
STAATLICH ANERKANNTE HOCHSCHULE



- User clicked one of the Record buttons
- Speaks sentence while the message box displays “Recording audio”
- Recording occurs for fixed time (approx. 10 seconds)
- Then…..(next slide)

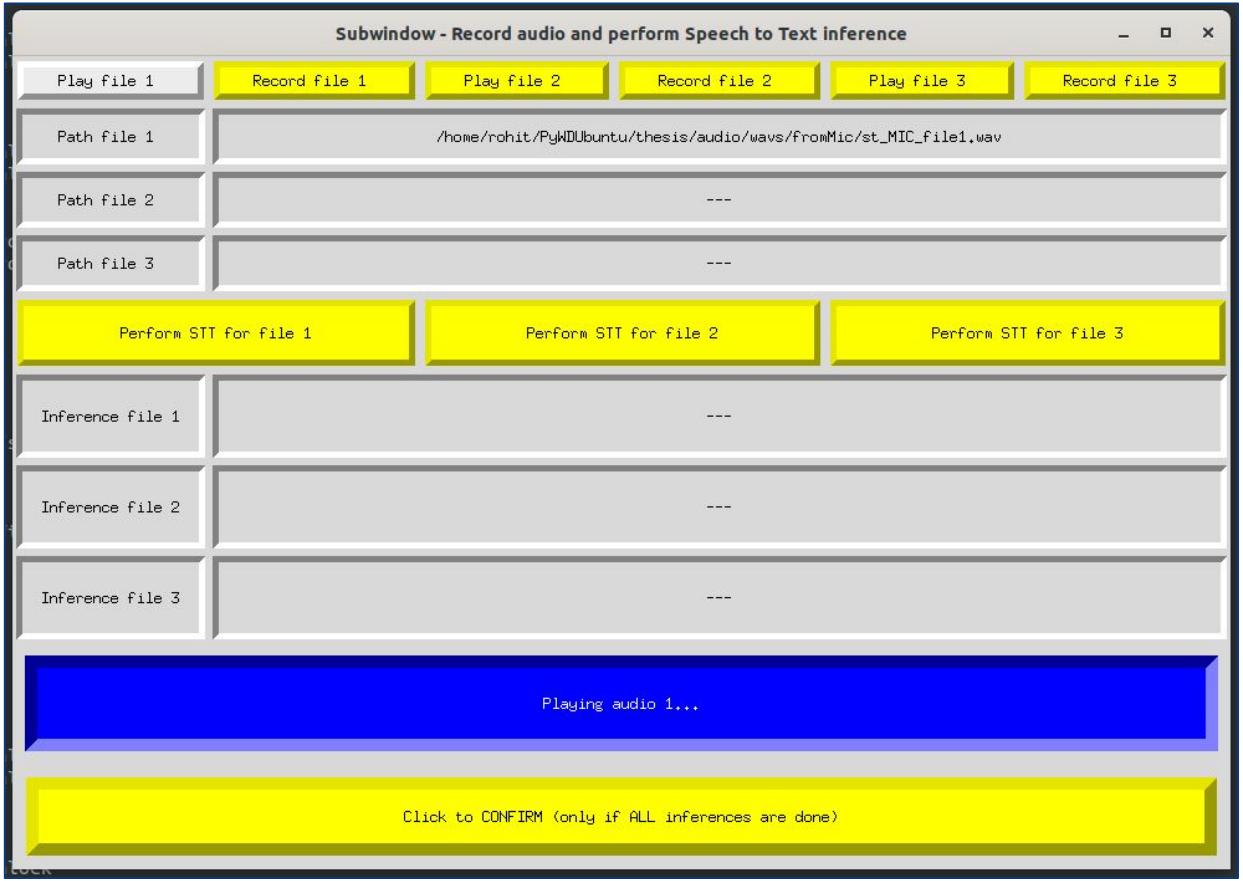
User records one sentence

STAATLICH ANERKANNTE HOCHSCHULE



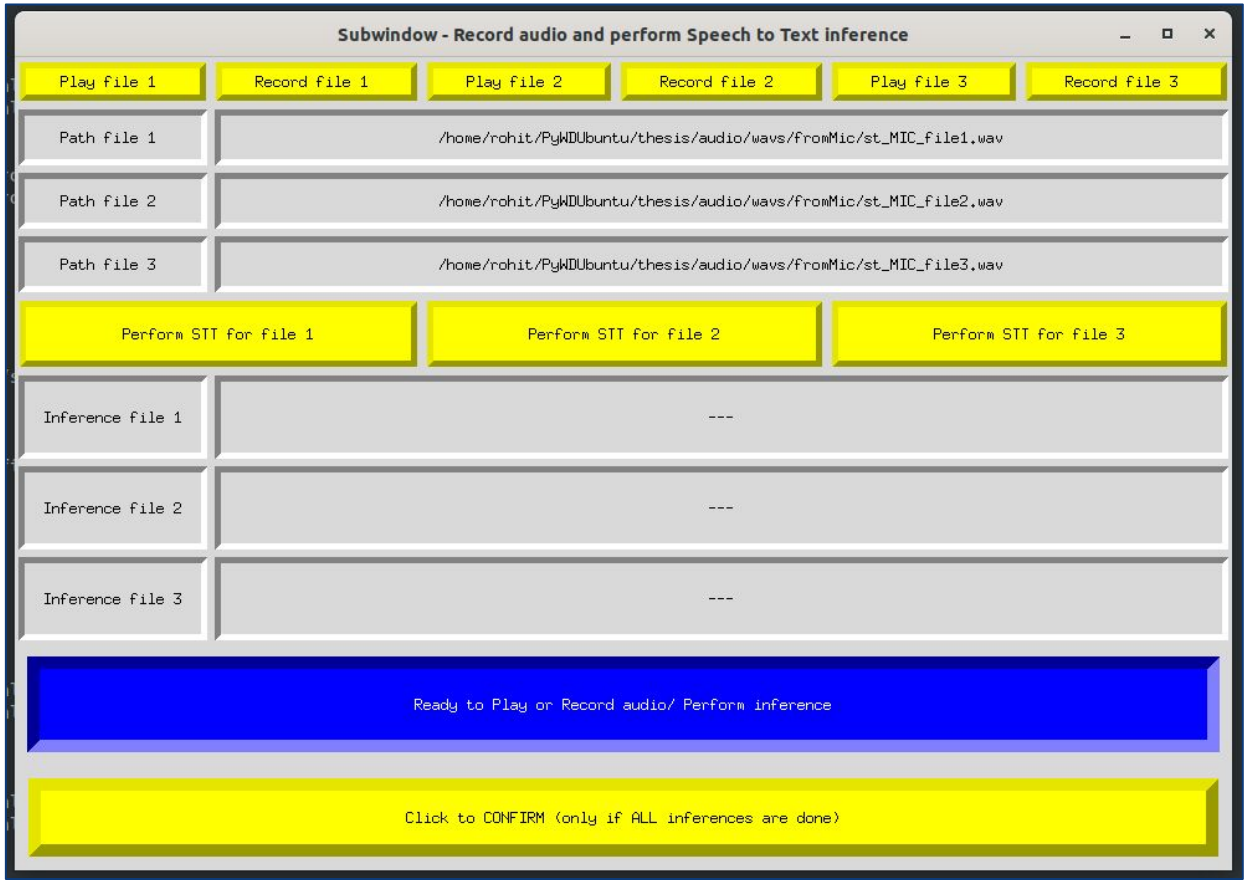
- On completion….
- File path updated
- Message box indicates Ready for next action

User plays one sentence



- User clicked Play button for an already created file
- Message indicates playback occurring
- On completion….
- Message indicates Ready for next action

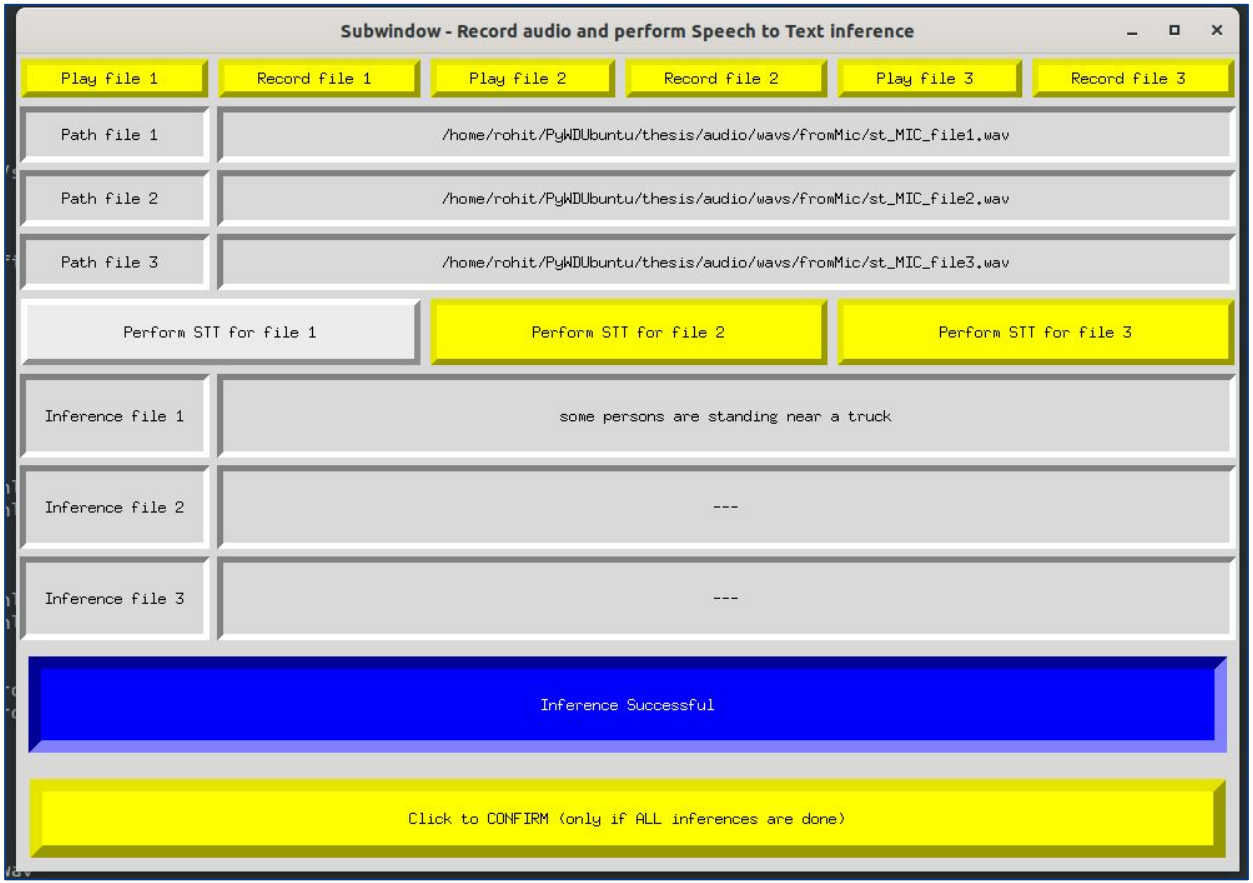
User recorded all three sentences



- User recorded all three sentences
- Happy with the playback
- Next action expected: ***Perform STT inference***

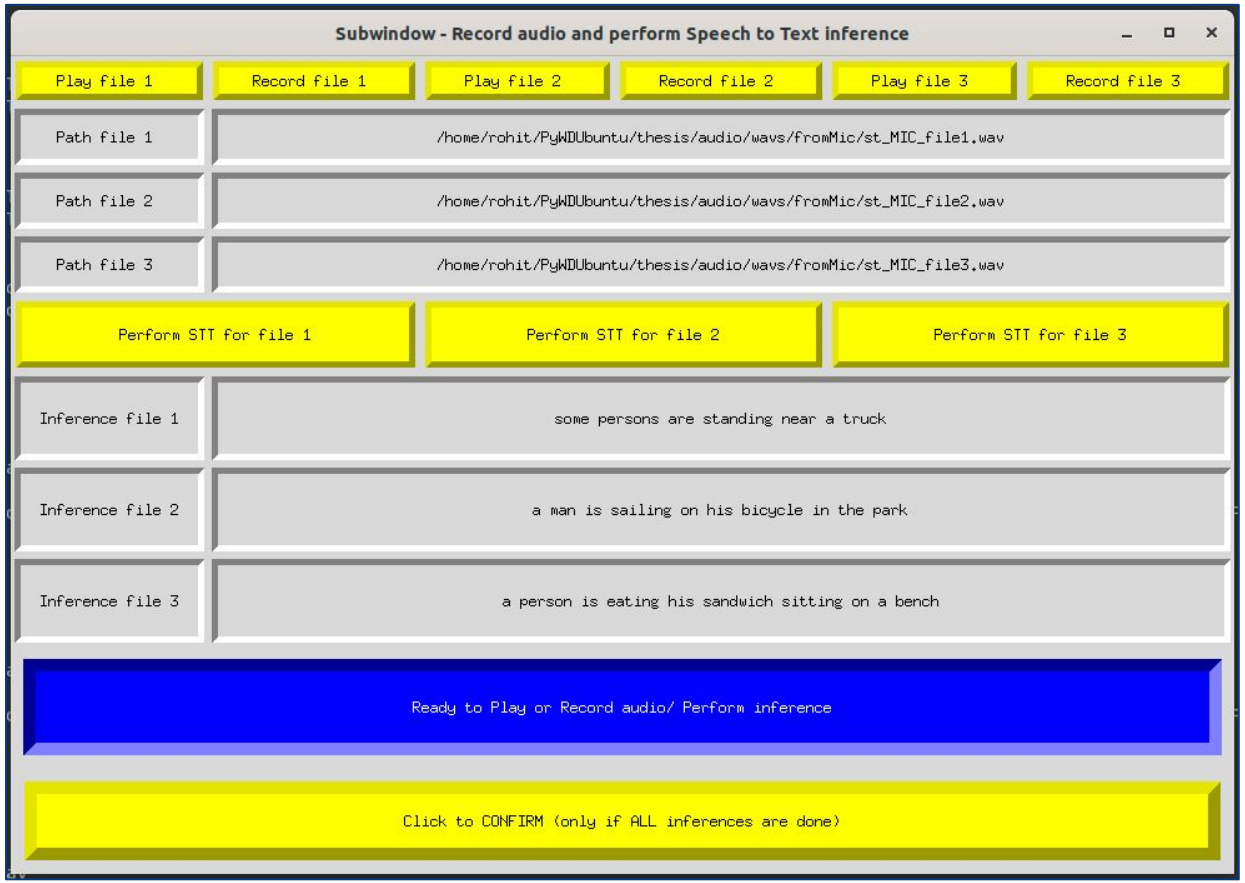
User performs STT inference

STAATLICH ANERKANNTE HOCHSCHULE



- User clicked button to perform STT inference for a file
- Output inference displayed after completion
- Success message briefly displayed

All recordings and inference done



- All three audio files recorded and inference done
- Next action expected:
  User clicks the Confirm button to proceed to next stage

Console output at end of stage

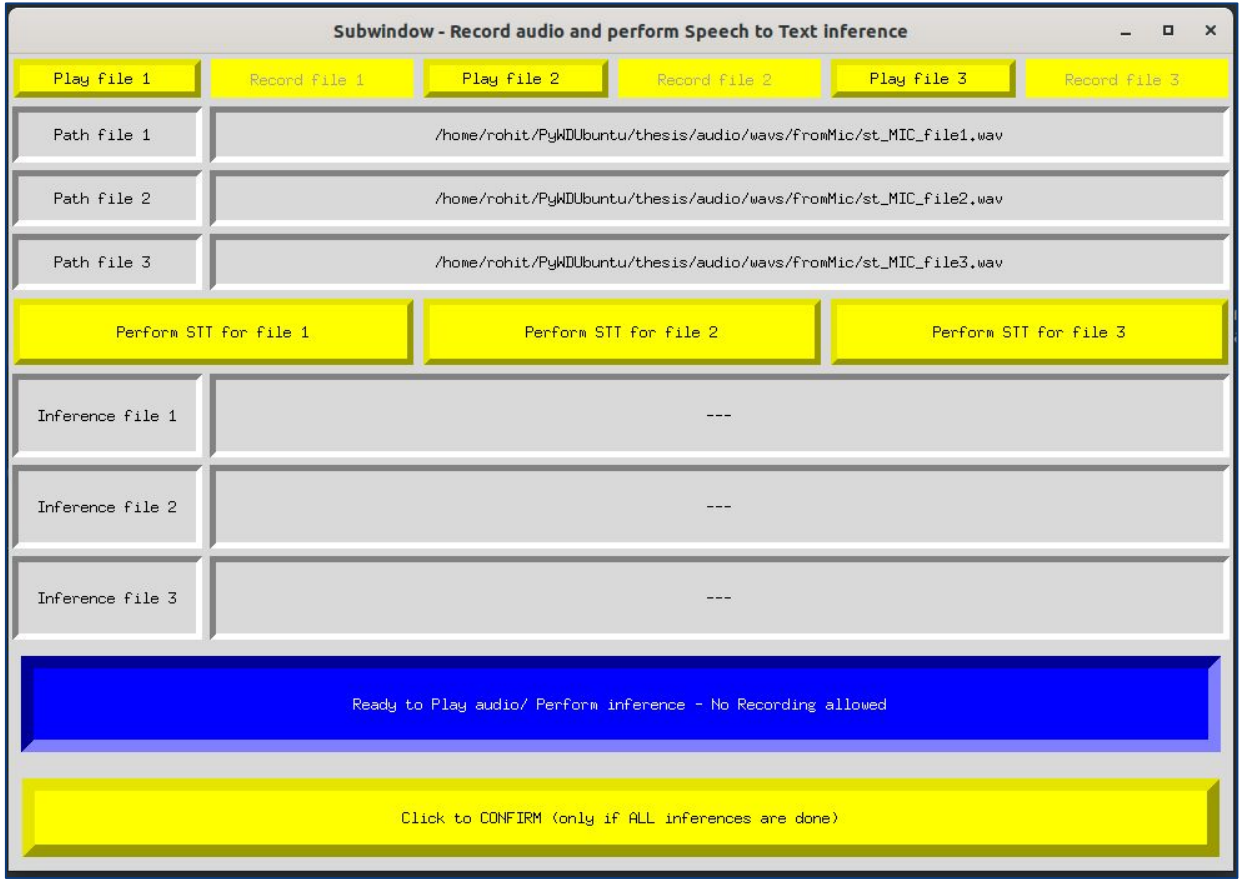
STAATLICH
ANERKANNTE
HOCHSCHULE

- Console output showing data structure (list called mic_stt_module_results)
- This is passed to Identify Key Elements stage

```
LOG_LEVEL INFO ::
After MIC INPUT + STT logic execution:
mic_stt_logic_RC = 0
mic_stt_logic_msg = None
mic_stt_module_results = ['some persons are standing near a truck', 'a man is sailing on his bicycle in the park', 'a person is eating his sandwich sitting on a bench']
```

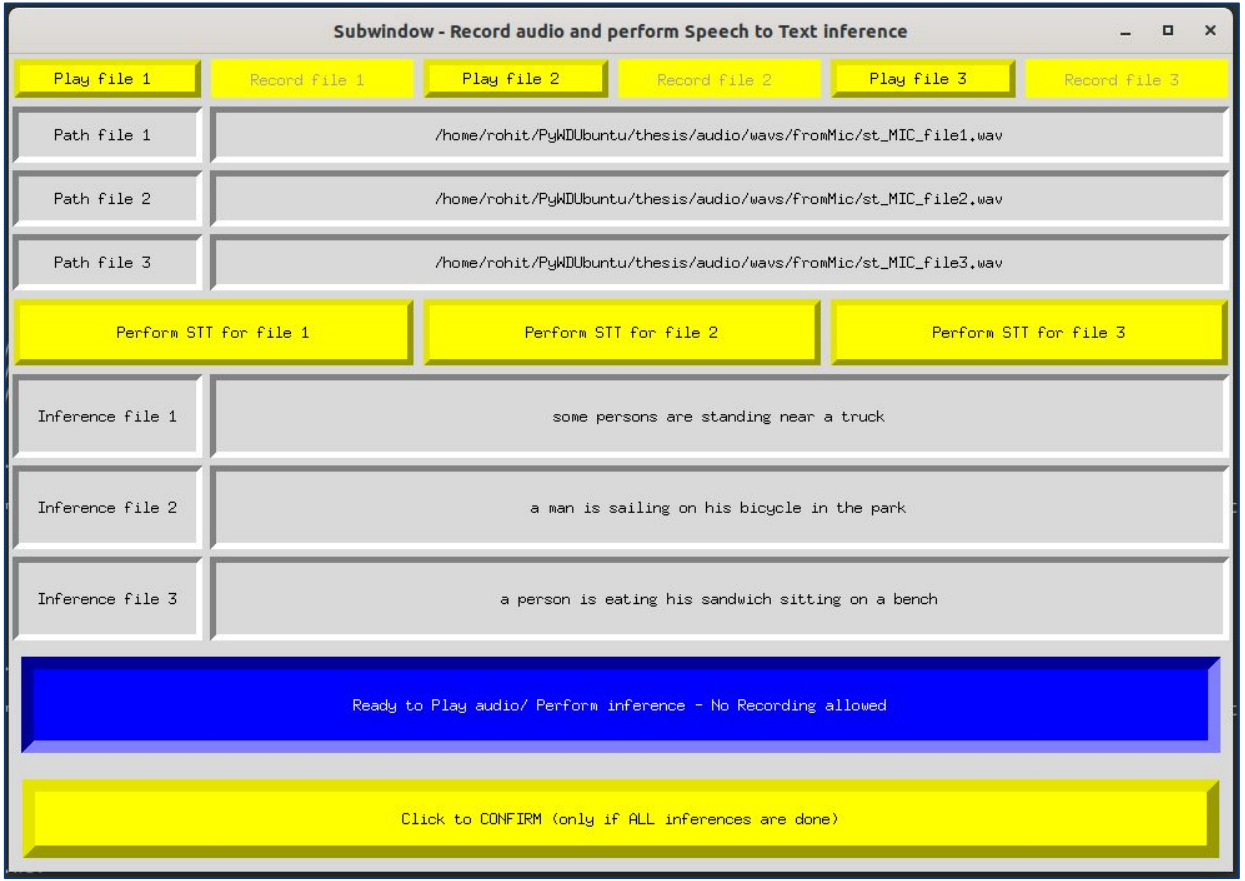
```
LOG_LEVEL INFO ::
Commencing STT inference with Deepspeech version 0.7.3.
on wav file = /home/rohit/PyWDUbuntu/thesis/audio/wavs/fromMic/st_MIC_file3.wav
        Command built as :
deepspeech --model /home/rohit/deepspeech/pretrained/v073/deepspeech-0.7.3-models.pbmm --scorer /home/rohit/deepspeech/pretrained/v073/deepspeech-0.7.3-models.scorer --audio /home/rohit/PyWDUbuntu/thesis/audio/w
avs/fromMic/st_MIC_file3.wav
LOG_LEVEL INFO ::
Inference output :
a person is eating his sandwich sitting on a bench
LOG_LEVEL INFO ::
File for wavlocfile created containing locations of Wav files

LOG_LEVEL INFO ::
After MIC INPUT + STT logic execution:
mic_stt_logic_RC = 0
mic_stt_logic_msg = None
mic_stt_module_results = ['some persons are standing near a truck', 'a man is sailing on his bicycle in the park', 'a person is eating his sandwich sitting on a bench']
```

No recording required - Initial window



- Same layout *except*:
  - > Record buttons disabled
  - > Wav files already exist so paths display them at the outset

No recording run - ready to Confirm



- User ran inference for all Wav files
- Happy with output
- Finally clicks Confirm button (as before) to proceed to next stage

Console output at end of stage

STAATLICH
ANERKANNTE
HOCHSCHULE

- Same as when running with Mic input required

```
LOG_LEVEL INFO ::
After MIC INPUT + STT logic execution:
mic_stt_logic_RC = 0
mic_stt_logic_msg = None
mic_stt_module_results = ['some persons are standing near a truck', 'a man is sailing on his bicycle in the park', 'a person is eating his sandwich sitting on a bench']
```


```
LOG_LEVEL INFO ::
After MIC INPUT + STT logic execution:
mic_stt_logic_RC = 0
mic_stt_logic_msg = None
mic_stt_module_results = ['some persons are standing near a truck', 'a man is sailing on his bicycle in the park', 'a person is eating his sandwich sitting on a bench']
```

Inference - word replacements possibly done

STAATLICH ANERKANNTE HOCHSCHULE

- The database of images has specific names for the objects detected - aka **Labels**. E.g. “handbag”, “tvmonitor”, etc.
  - > In next stage, words from inference output matched against the specific labels.

- Implication:
  - > An inference output of “*the **hand bag** has many items in it*” **will not match** “hand bag” with the **label “handbag”** and logic fails!
  - > ***Special words are replaced to allow downstream processing to succeed.***

Changes made to inference output in this case

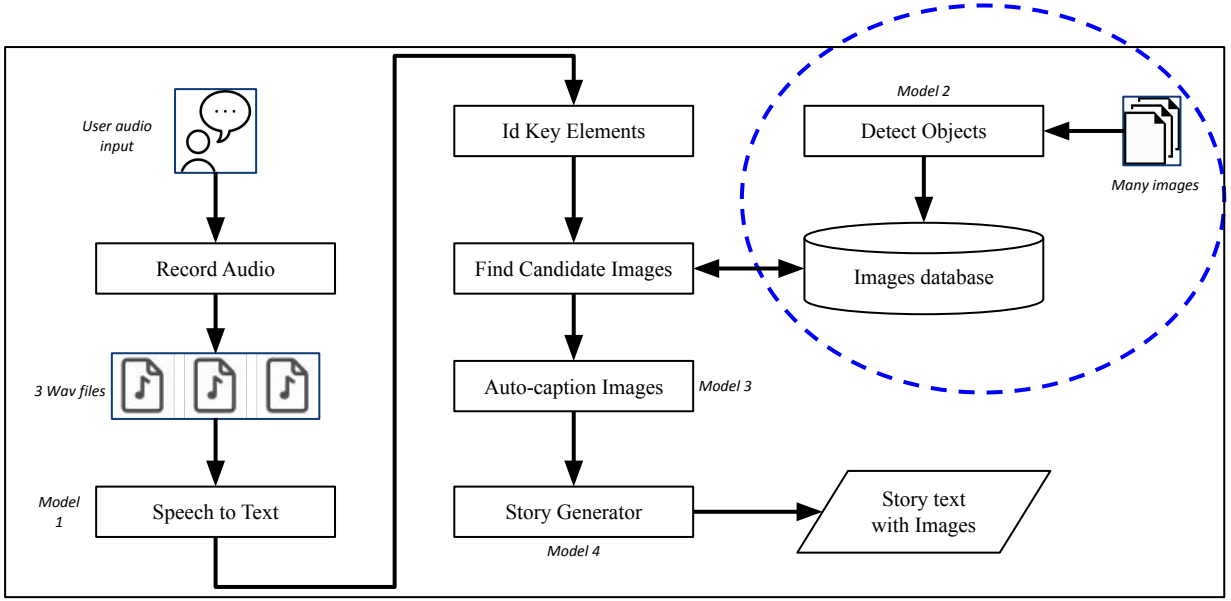
```
LOG_LEVEL INFO ::
Commencing STT inference with Deepspeech version 0.7.
on wav file = /home/rohit/PyWDUbuntu/thesis/audio/wav
        Command built as :
deepspeech --model /home/rohit/deepspeech/pretrained/
avs/fromMic/st_MIC_file2.wav
LOG_LEVEL INFO ::
Word replacement: CHANGES made
Orig inference =
a person watches the news on the television monitor
Changed inference =
a person watches the news on the tvmonitor
```

No change to inference output in thisi case

```
LOG_LEVEL INFO ::
Commencing STT inference with Deepspeech version 0.7.
on wav file = /home/rohit/PyWDUbuntu/thesis/audio/wav
        Command built as :
deepspeech --model /home/rohit/deepspeech/pretrained/
avs/fromMic/st_MIC_file1.wav
LOG_LEVEL INFO ::
Word replacement: NO change
```

STAATLICH
ANERKANNTE
HOCHSCHULE

One-off: Create image database using Object Detector



Goal and Approach

STAATLICH
ANERKANNTE
HOCHSCHULE

- **Goal**: Create a database storing information about the objects detected in the various images.
- Ability to query a large database (several tens of thousands of images)
- One-off execution to create database in advance.
- During story generation process, only query to select suitable images with objects identified from user input.
- Neural networks based on CNN based models popular for such tasks
  - > Single stage detectors (e.g. Yolo, etc.) much faster than 2-stage detectors (RCNN family, etc.)
- ***Object Detector used: You Only Look Once version 3 (YOLOv3)***
  - > A single stage detector
  - > Discarded output inference image due to space constraints.
  - > Pre-trained on Coco-2017 data
- ***Graph database used to capture which images have which objects***
  - > Neo4j graph

YOLO use case in thesis work

STAATLICH
ANERKANNTE
HOCHSCHULE

Goal: Use the YOLOv3 model to perform inference and store information in Neo4j graph database

Use Case:
Present a set of new images to a pre-trained YOLOv3 model.
For each image, capture the detected **object class** and the **confidence score**.
Store information in a neo4j graph database:

- Relationship format:
  (i:Image{ name: "Image123.jpg", data: "dataset source" }) - [r:HAS { score: confidence score }] -> (o:Object { name: "object class" })
- Image node's "data" property value identifies which dataset the image belongs to. E.g. coco_train_2017
- HAS relationship's "score" property value is the confidence score of object being present in the image

E.g. Image123.jpg HAS the objects:

- car (score 58.98%),
- person (score 98.34%)
- person (score 93.23%)

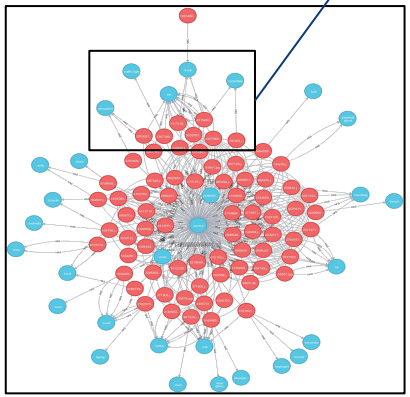
Note: Used a threshold score of 0.45 while populating the database. But during image querying later, will use higher threshold.

- if Object -> HAS[score > 0.45] -> Object :: only then insert into Neo4j db.

Neo4j database - How does the data look?

STAATLICH ANERKANNTE HOCHSCHULE

Neo4j db after inserts



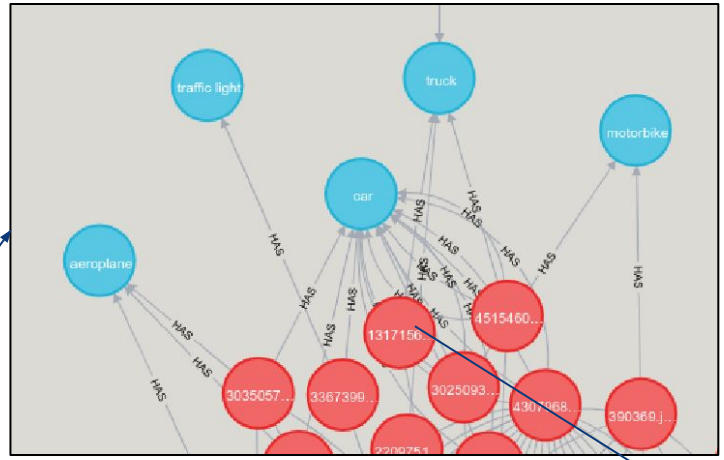


Image (red node) - HAS (line) -> Object (green node)

Objects found in images: Traffic light, truck, motorbike, car, etc.

Many to many relationship could exist.

Detection output: 1317156_det.jpg

Data used to build Neo4j query database

STAATLICH ANERKANNTE HOCHSCHULE

- Common Objects in Context (COCO): http://cocodataset.org/ , http://cocodataset.org/#download
  - > COCO is a large-scale object detection, segmentation, and captioning dataset.
  - > 80 object categories

- Flickr dataset: http://shannon.cs.illinois.edu/DenotationGraph/ , https://www.kaggle.com/hsankesara/flickr-image-dataset

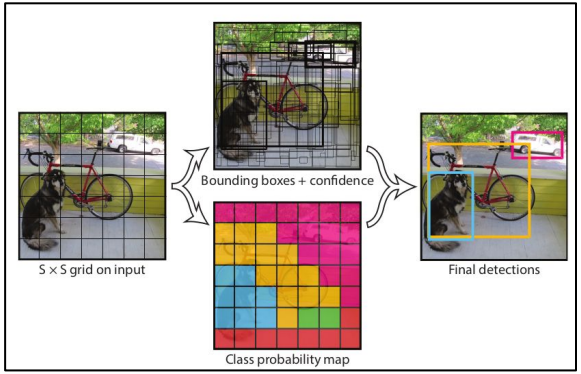
COCO: Common Objects in Context Flickr: Common Objects in Context		
Dataset	**Dataset**	**No. of Images**
COCO	Test	41k
Flickr-30	-	30k

Neo4j database after population

How YOLO algorithm works

STAATLICH ANERKANNTE HOCHSCHULE

- Divides image into a grid of size S x S where S is an integer
- Each pixel evaluated as possible center point of an object
- All detections are evaluated in one pass - very fast algorithm
- Model is trained to identify C classes of objects
- B is the number of Bounding Boxes detected all over the image (without threshold consideration). Five values are output for each bounding box:
  - Two values for center coordinates
  - Two values for dimensions (height and width)
  - Confidence score
- Can handle multiple bounding boxes and aspect ratios (anchor box concept)
  - Anchor boxes are predefined boxes provided by the user to Darknet which gives the network an idea about the relative position and dimensions of the objects to be detected.
  - These are calculated using the training set Objects.

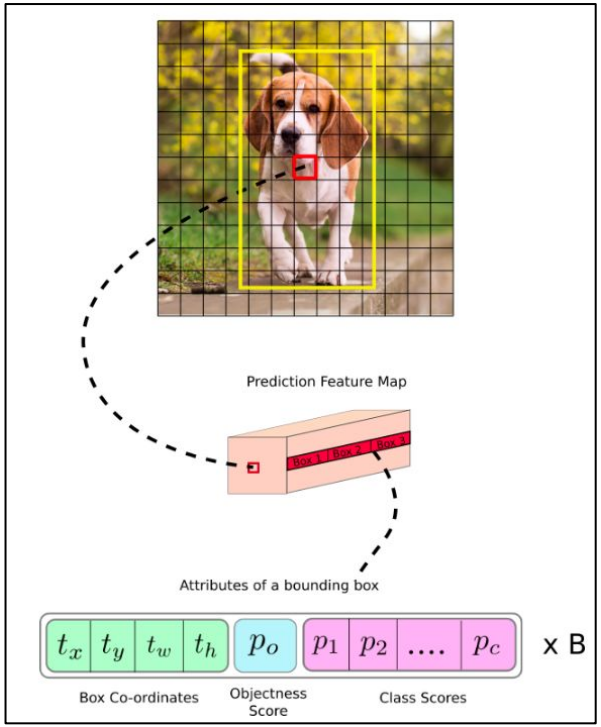


Source: YOLO v1 paper: https://arxiv.org/abs/1506.02640

How YOLO algorithm works

STAATLICH ANERKANNTE HOCHSCHULE

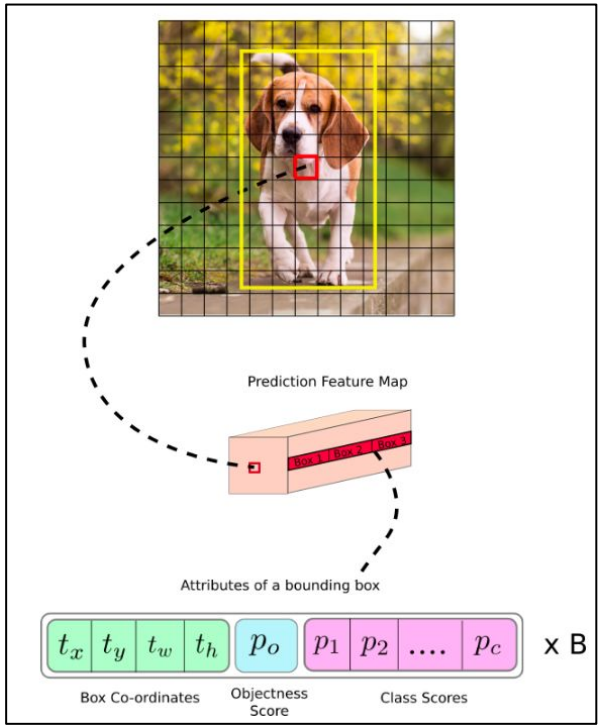
- Usually Non-max suppression used to remove redundant detections
- Total detections per image = (S x S) * (B * (5 + C))
  - Each bounding box has 5 + C attributes
- For example, suppose that:
  - image is divided into 13 x 13 grid (i.e. S = 13)
  - we want to detect 80 classes for COCO (i.e. C = 80)
  - 3 boxes predicted (B = 3)
  - #Detections = (13 x 13) * (3 * (5 + 80)) = 13 x 13 x 255
    = 43,095
- Threshold value used for Confidence Score to evaluate acceptance of object detection



Source: https://medium.com/analytics-vidhya/yolo-v3-theory-explained-33100f6d193

How YOLO algorithm works

STAATLICH ANERKANNTE HOCHSCHULE



Source: https://medium.com/analytics-vidhya/yolo-v3-theory-explained-33100f6d193

- Image divided into a grid of S x S grid-boxes
  - Predictions made at 3 scales where a 416 x 416 pixel input image is divided into 13x13, 26x26 and 52x52
- Each grid can predict maximum B objects if it thinks object center lies in the pixels covered by the grid
- Model outputs regressed values for box information: t_x, t_y, t_w, t_h
  - Mathematical formulae applied later to get actual bounding box center coordinates and dimensions: b_x, b_y, b_w, b_h

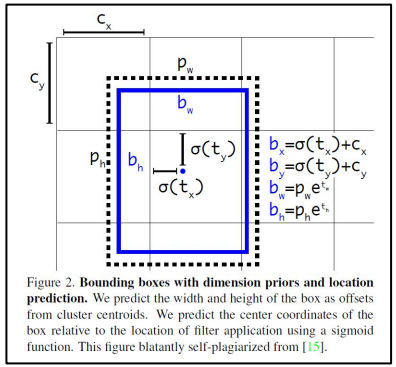


Figure 2. **Bounding boxes with dimension priors and location prediction.** We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function. This figure blatantly self-plagiarized from [15].

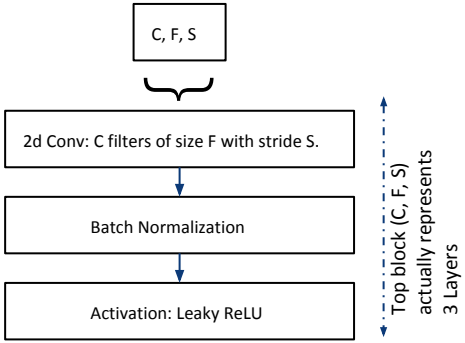
Source: Yolo v3 paper: https://arxiv.org/abs/1804.02767

Yolo Model architecture - Basic Elements

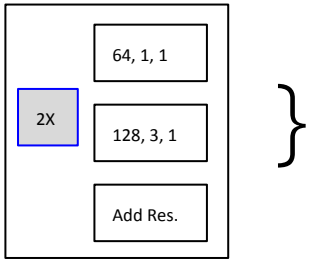
STAATLICH ANERKANNTE HOCHSCHULE

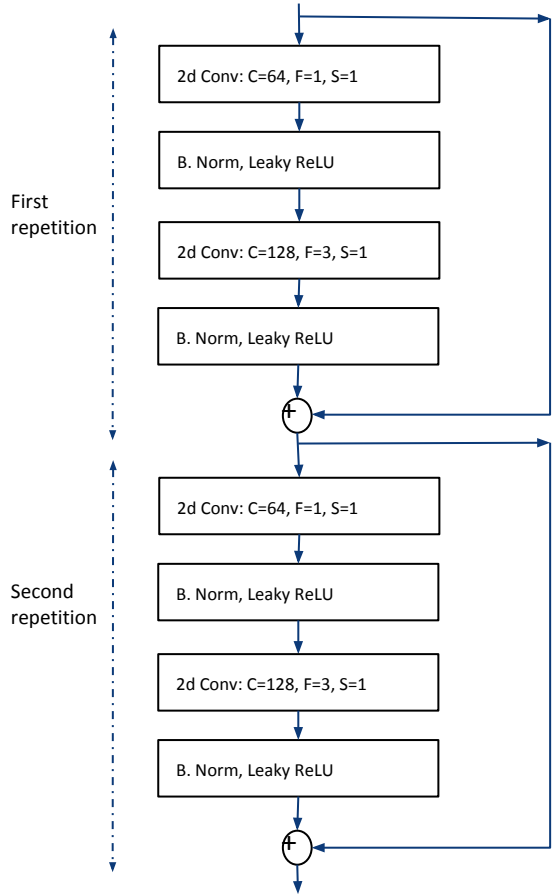
- Series of Convolution, then Batch Normalization, then Leaky ReLU. This is repeated many times.
- Convolution type:
  - Stride = 1, then use “same”
  - Stride = 2, then use “valid”

- Convolution parameters:
  - C: #Filters
  - F: Filter size
  - S: Stride

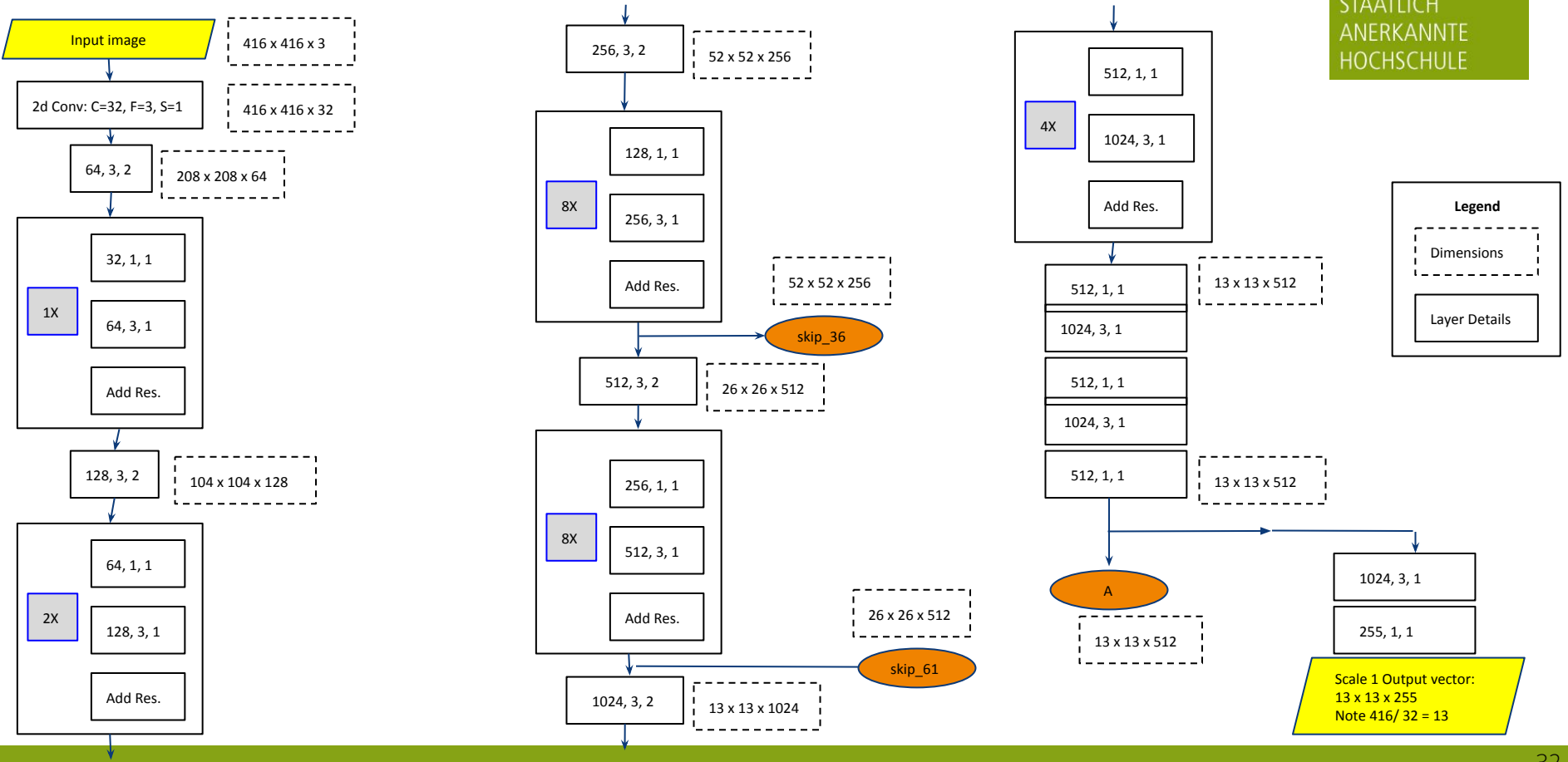


- The 2X means this set of operational blocks is repeated two times.



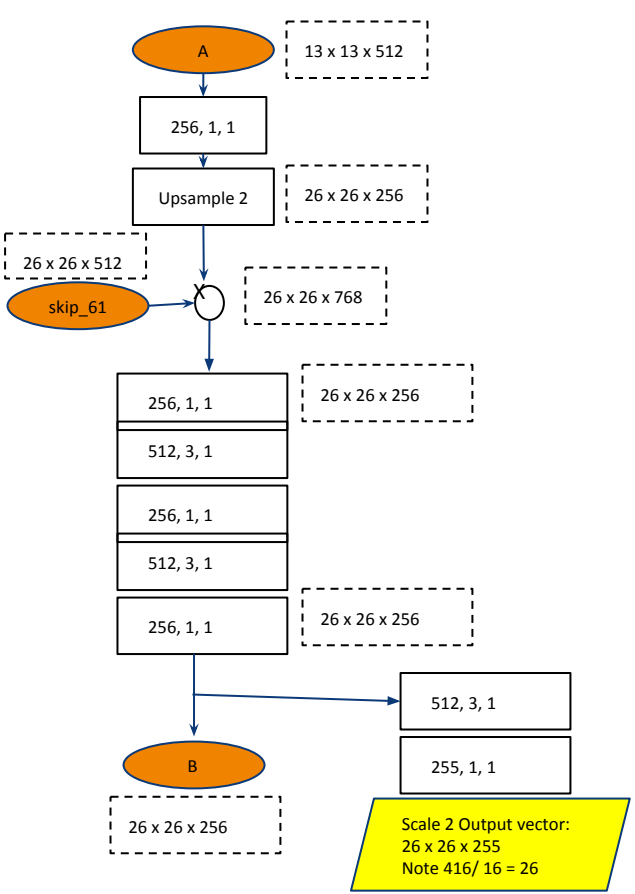


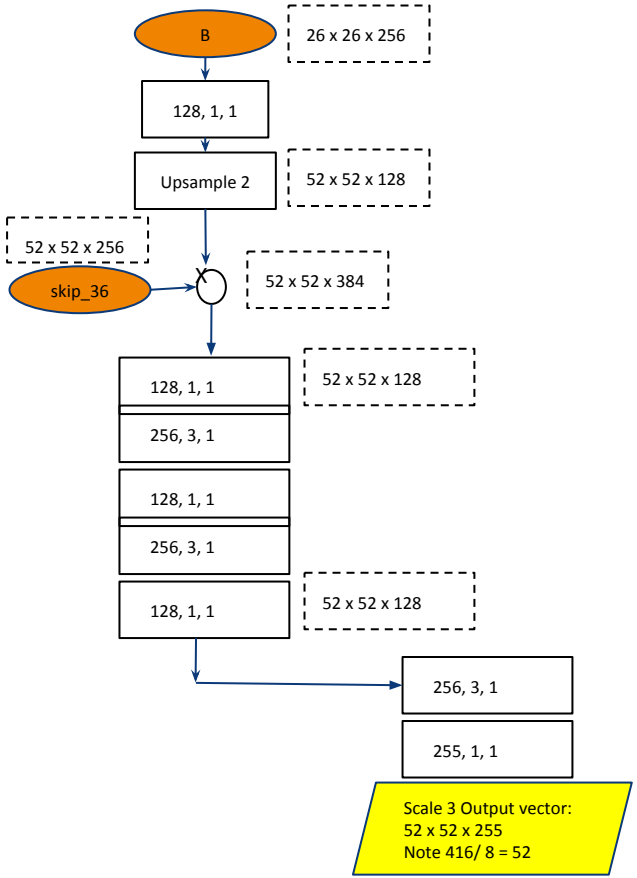
Yolo Model - Detailed architecture

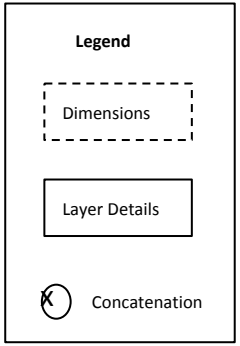


STAATLICH ANERKANNTE HOCHSCHULE

Yolo Model - Detailed architecture

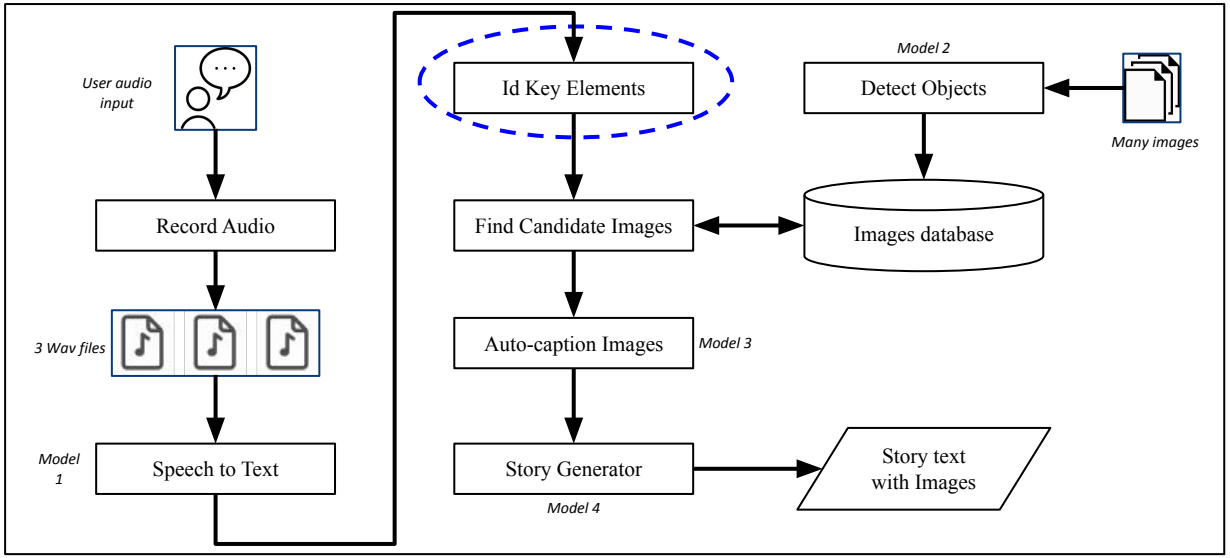






STAATLICH
ANERKANNTE
HOCHSCHULE

Stage 2: Identify Key Elements



Goal and Approach

STAATLICH
ANERKANNTE
HOCHSCHULE

- **Goal**: Process the transcriptions from the STT block and figure out the Keywords to pass to the database querying stage.

- Tried using python modules: NLTK and Spacy. Found Spacy better.
  - > Spacy much faster and is production environment ready tool.
  - > Downside of Spacy is less inbuilt algorithms compared to NLTK.
    - > But found the POS tagging very exhaustive for Spacy and working well
  - > Using the “large” model: https://spacy.io/models/en#en_core_web_lg

- Logic performed using Spacy:
  - > Word tokenization Remove all stop-words POS tagging
  - > Keeping only Noun type words of three tags as the candidate keywords (https://spacy.io/api/annotation#pos-tagging):
    - > Tag = “NN” - Noun, singular or mass
    - > Tag = “NNS” - Noun, plural
    - > Tag = ”NNP” - Noun, noun proper singular
  - > Extracting the Lemma form of the word, not the original word itself
  - > E.g. Spacy information for the input word = “car”
    results = {**"text": "car", "lemma_": "car"**, "pos_": "NOUN", "tag_": "NN", "dep_": "compound", "shape_": "xxx", "is_alpha": true, "is_stop": false}

- From the set of words kept after inspecting the POS-Tag, retained only words that are “objects in the database”. Only these retained words are presented to user for selection via a GUI later.

Unreliable sentence splitting !

STAATLICH ANERKANNTE HOCHSCHULE

- Sentence tokenization is not reliable.

> Input had two sentences from the output of STT block

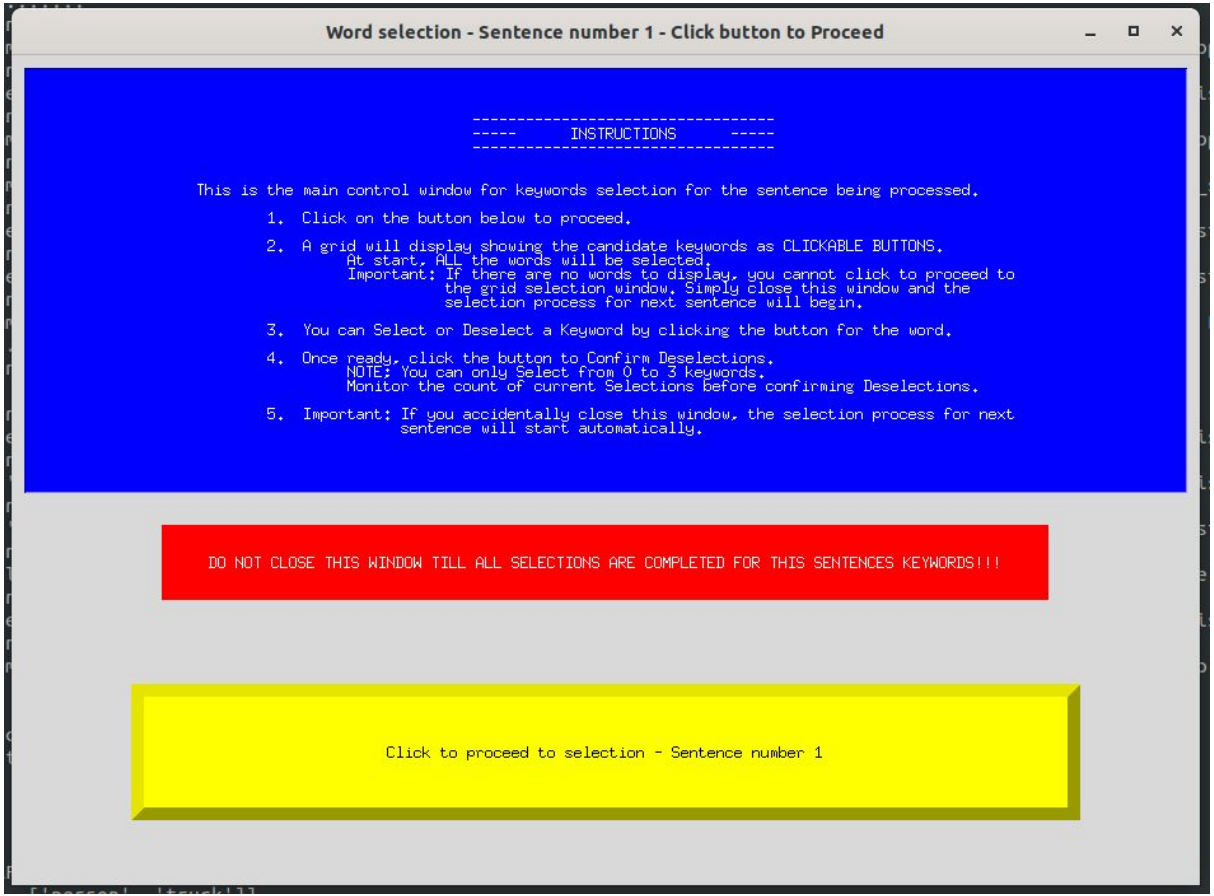
> Spacy: got one wrong
> got one right

> NLTK: got both wrong

```
In [14]: import spacy
         nlp = spacy.load('en_core_web_lg')

In [15]: arr = ['i want a story about a car on the road a child plays with a toy',
                'generate a story about persons walking on the street a truck is on the road']

In [16]: doc = nlp("This is a sentence. This is another sentence.")
         for sent in doc.sents:
             print(sent.text)

         This is a sentence.
         This is another sentence.

In [17]: doc = nlp(arr[0])
         for sent in doc.sents:
             print(sent.text)

         i want a story about a car on the road a child plays with a toy

In [18]: doc = nlp(arr[1])
         for sent in doc.sents:
             print(sent.text)

         generate a story about persons walking on the street
         a truck is on the road
```

```
In [9]:  for entry in arr:
             result.append(sent_tokenize(entry))

In [10]: result

Out[10]: [['i want a story about a car on the road a child plays with a toy'],
          ['generate a story about persons walking on the street a truck is on the road']]
```

Main window - expected user behaviour

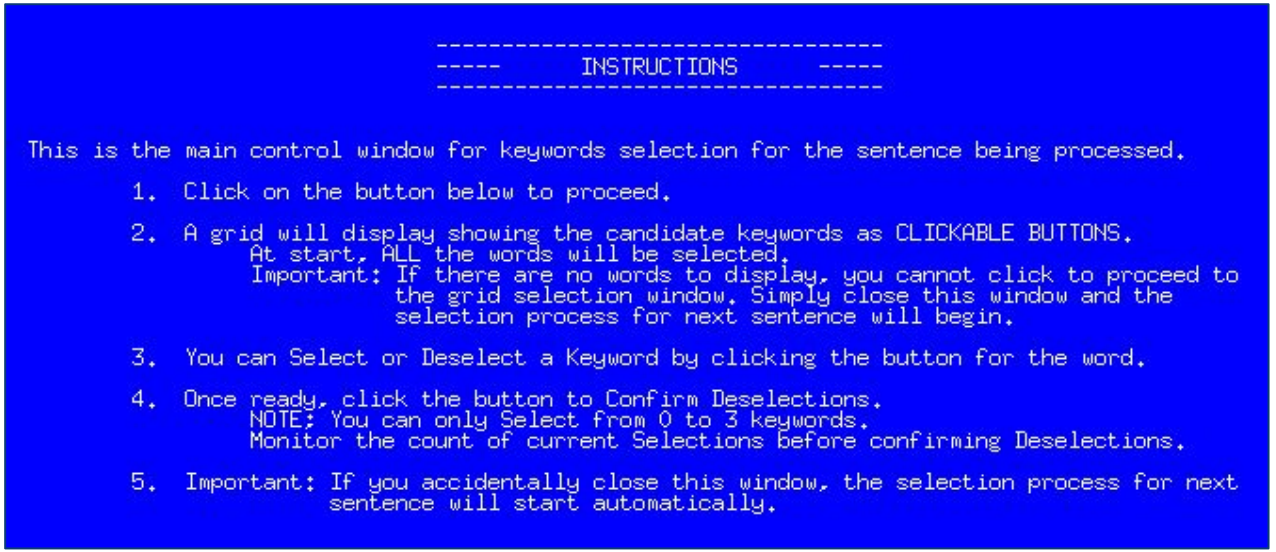


STAATLICH ANERKANNTE HOCHSCHULE

- Present candidate keywords via a GUI to user for final selection
- User to click on Yellow button to proceed
- If no words available for selection, button is disabled and user simply closes the window and moves to selection process for next sentence.
- From the candidate keywords of each of sentence, user must finally Select exactly 0/ 1/ 2/ 3 words.
- Note: logic already pre-checks the words and only presents matches against the current set of object class labels

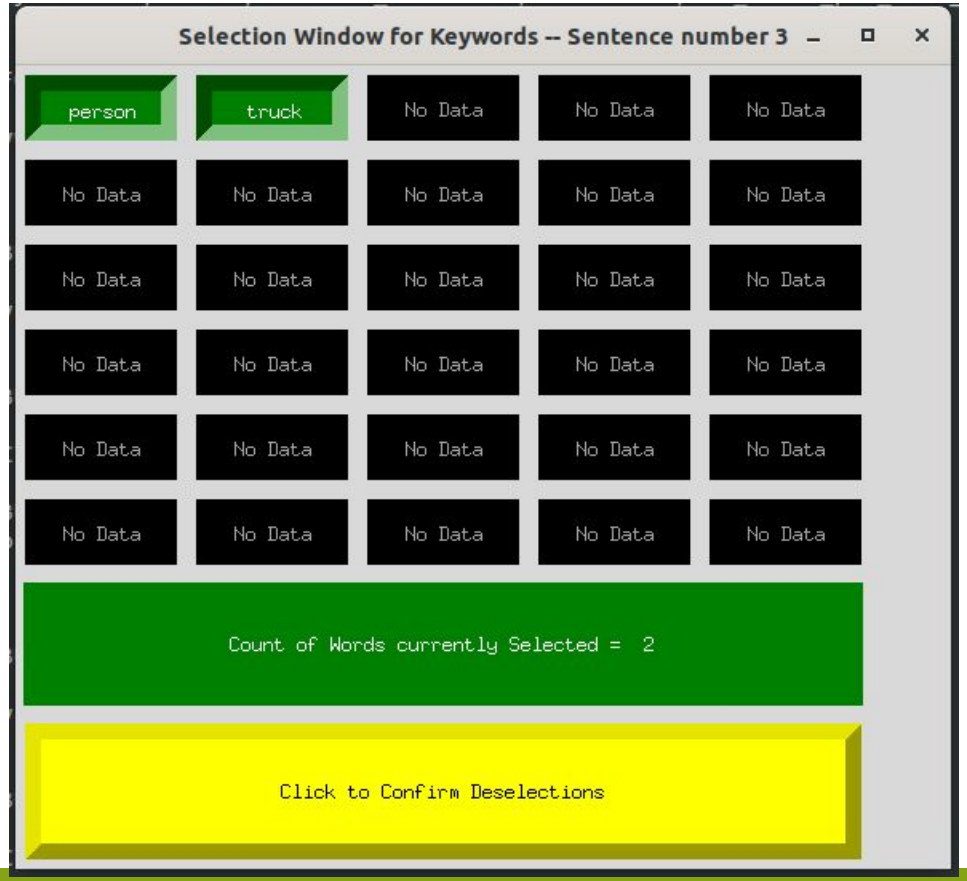
Main window with instructions

STAATLICH
ANERKANNTE
HOCHSCHULE



Subwindow - expected user behaviour

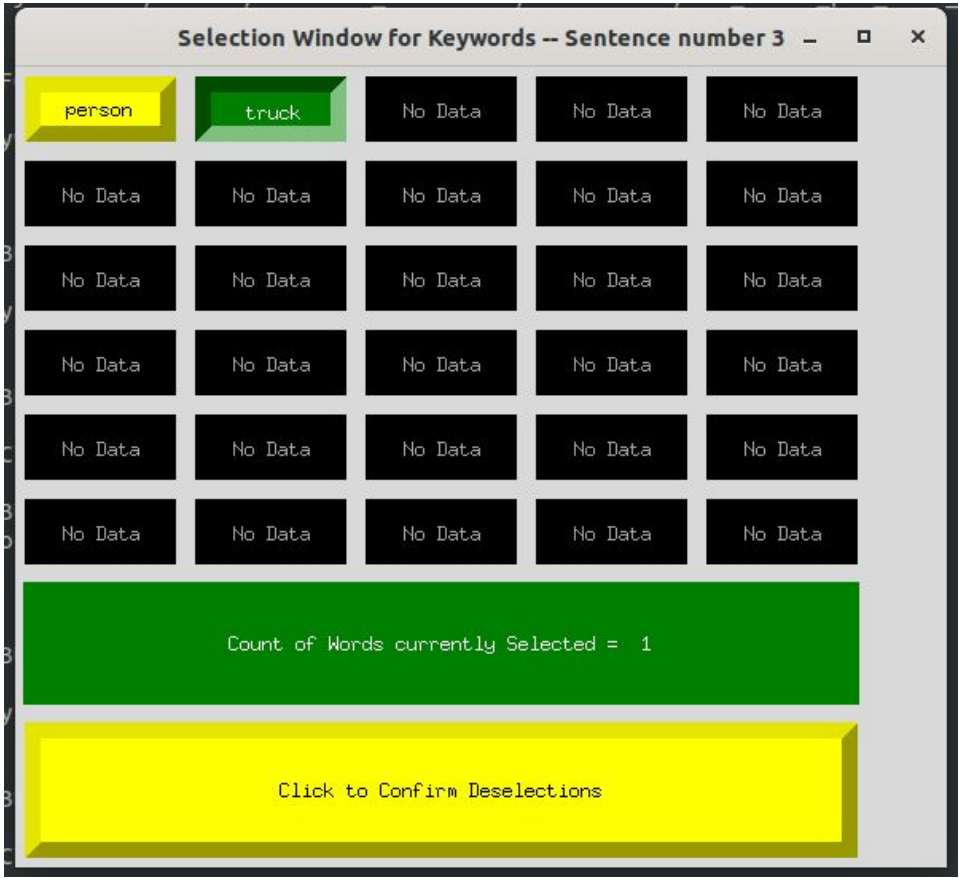
STAATLICH ANERKANNTE HOCHSCHULE



- Up to 30 candidate keywords possible
  - > non-stopwords of type noun which matched object class labels in Neo4j database
- Here 2 candidate keywords shown
- Remaining placeholders indicate No Data.
- All words Selected by default.
- Current count shown in button (value here = 2).
  - > If count > 3, then invalid and color would be Red.
  - > If color is Green, means valid count (0 to 3)
- Confirm Selections button is enabled

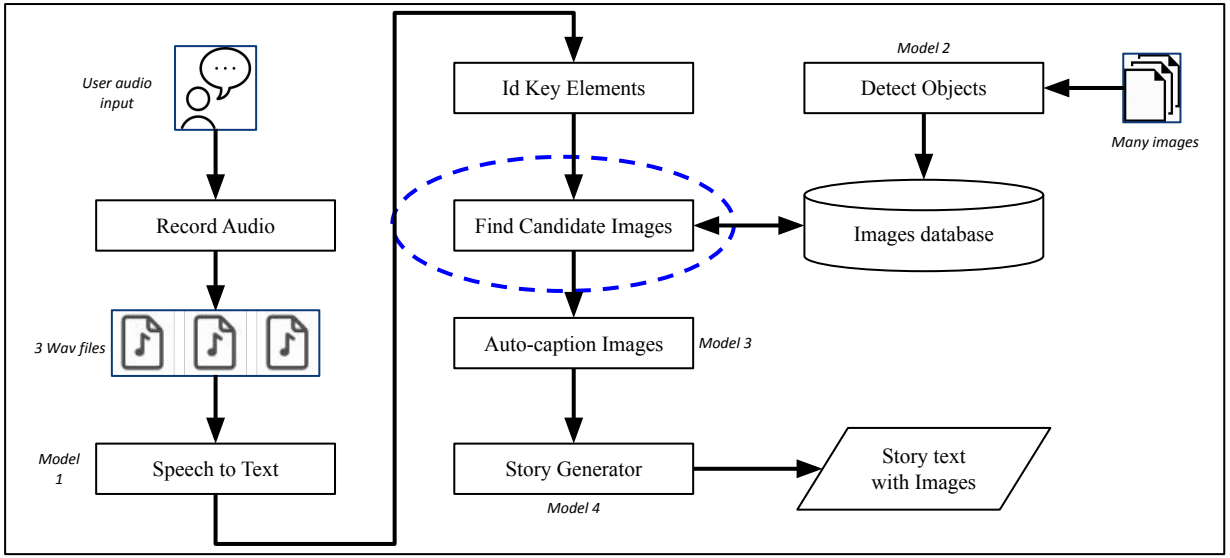
Select / Deselect words

STAATLICH ANERKANNTE HOCHSCHULE



- User clicked word “person” to Deselect it
- Count is now 1
  - > Color remains Green
- User happy with the remaining selections….
- Clicks yellow button to Confirm current Deselections
- Now only “truck” is passed on to Find Candidate Images block

Stage 3: Find Candidate Images



Goal and Approach

STAATLICH ANERKANNTE HOCHSCHULE

- **Goal**: With the user selected final keywords from previous stage, query the Neo4j database to retrieve images containing those objects of interest.

- Up to 20 images returned per query as “candidate images”.
  - > Only if image contains at least one object with a HAS relationship score > 90%
  - > Only from the COCO_Test2017 and Flicker8k datasets
    - > As other datasets used for Yolo model training

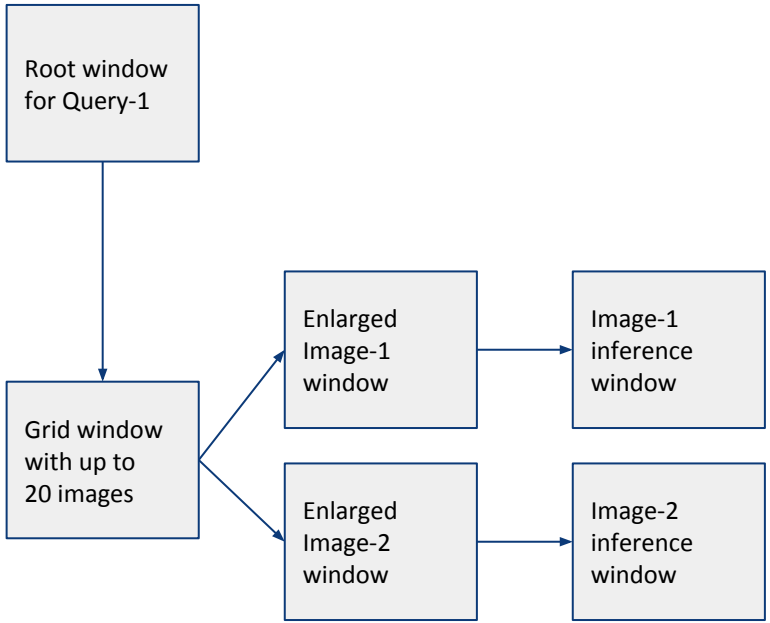
- Allow user to view candidate images and Deselect any images:
  - > Misclassified images.
  - > Reduce the total Selections count to maximum 5 images.

- User selection via GUI interface:
  - > Images thumbnails shown.
    - > Clickable to “Select” or “Deselect”
  - > View Enlarged image for close inspection
  - > Perform one-off object detection inference
  - > Finally only keep the Selected images to pass on to the next stage i.e. Auto caption stage.

- GUI implemented using Tkinter python module

GUI Flow

- Repeats for each of the three queries - once per input sentence
  - > If no images, does not allow grid window selection logic



Neo4j query results may be incorrect

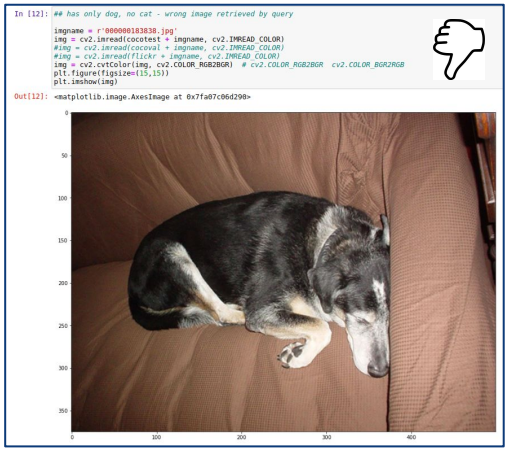
STAATLICH ANERKANNTE HOCHSCHULE

- Example:
  - > Queries for objects “dog” and “cat”
  - > First image - wrong - has only dog
  - > Second image - correct - has both cat and dog

- ***Therefore allow user to deselect such wrong images***

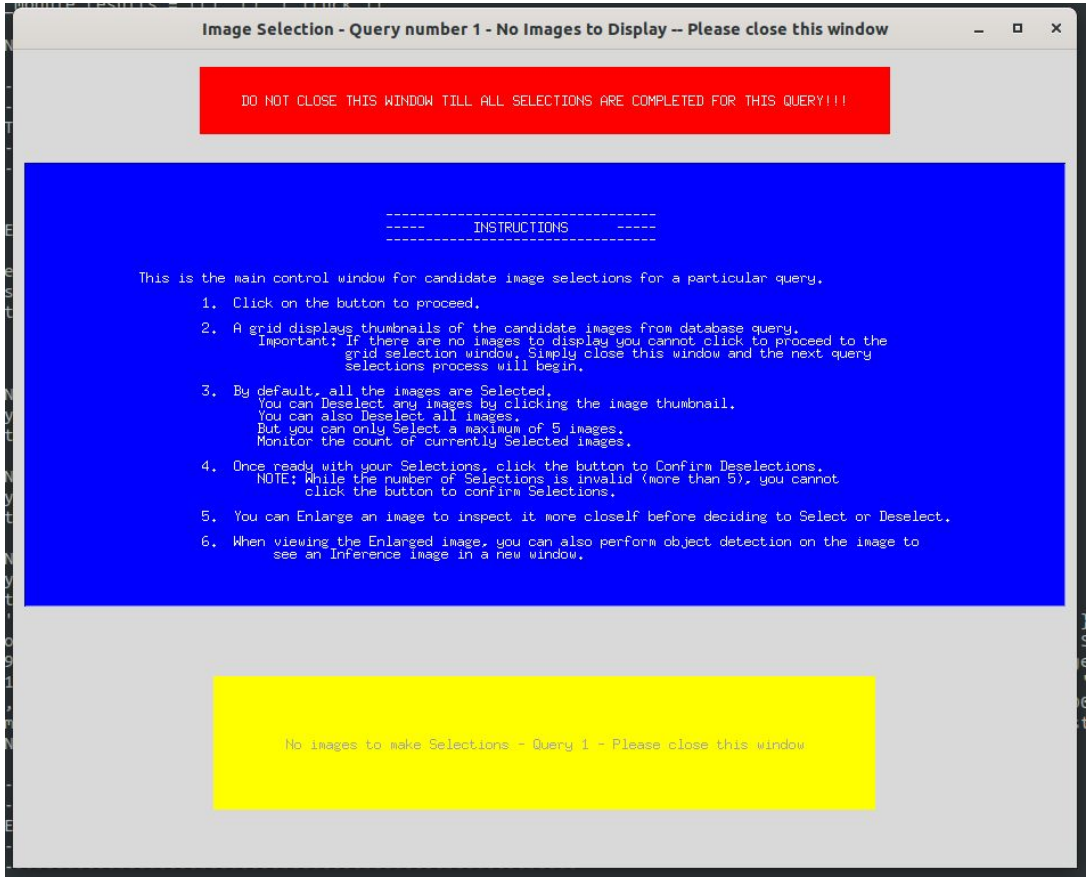
- Partial solution implemented::
  - > Used minimum threshold value of 90% during Neo4j query
  - > But is not foolproof

```
In [9]: for each_objects_list in input_list_for_query[:1]:
            print(f"\n\n{each_objects_list}:\n{query_neo4j_db(each_objects_list, 10)}")


['dog', 'cat']:
(0, [{'i.name': '000000183838.jpg', 'i.dataset': 'coco_test_2017'}, {'i.name': '000000313767.jpg', 'i.dataset': 'co
co_test_2017'}, {'i.name': '000000341174.jpg', 'i.dataset': 'coco_test_2017'}, {'i.name': '000000302015.jpg', 'i.da
taset': 'coco_test_2017'}, {'i.name': '000000101257.jpg', 'i.dataset': 'coco_test_2017'}, {'i.name': '000000209673.
jpg', 'i.dataset': 'coco_test_2017'}, {'i.name': '000000430186.jpg', 'i.dataset': 'coco_test_2017'}, {'i.name': '00
0000310779.jpg', 'i.dataset': 'coco_test_2017'}, {'i.name': '000000017269.jpg', 'i.dataset': 'coco_test_2017'},
{'i.name': '000000210056.jpg', 'i.dataset': 'coco_test_2017'}])
```



When a query has no keywords

STAATLICH ANERKANNTE HOCHSCHULE



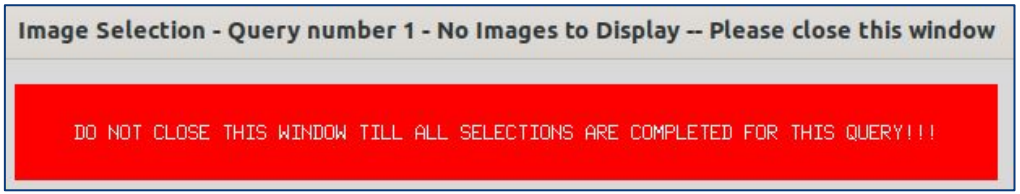
- Database query for this sentences' keywords returned no hits.
  - > ***Cannot click to proceed to grid selection.***
  - > Close window and proceed to selection process for next query results

When a query has no keywords

STAATLICH ANERKANNTE HOCHSCHULE

```
                    ----------------------------------
                    -----      INSTRUCTIONS      -----
                    ----------------------------------


This is the main control window for candidate image selections for a particular query.

      1.  Click on the button to proceed.

      2.  A grid displays thumbnails of the candidate images from database query.
          Important: If there are no images to display you cannot click to proceed to the
                     grid selection window. Simply close this window and the next query
                     selections process will begin.

      3.  By default, all the images are Selected.
          You can Deselect any images by clicking the image thumbnail.
          You can also Deselect all images.
          But you can only Select a maximum of 5 images.
          Monitor the count of currently Selected images.

      4.  Once ready with your Selections, click the button to Confirm Deselections.
          NOTE: While the number of Selections is invalid (more than 5), you cannot
                click the button to confirm Selections.

      5.  You can Enlarge an image to inspect it more cloself before deciding to Select or Deselect.

      6.  When viewing the Enlarged image, you can also perform object detection on the image to
          see an Inference image in a new window.
```

- User to simply close window and proceed to selection process for next query results



Query which has keywords found

- Example of query with results:
  - > Many images found containing the object “Truck”.
  - > ***Proceed to grid selection button is clickable***

Image thumbnails show query results

STAATLICH
ANERKANNTE
HOCHSCHULE

- Grid selection window displays the up to 20 images.

What the user sees

- All images Selected by default - the border is Green
- User clicks **Enlarge** button
  - > Then new window displays enlarged image
- But **count** is 20 (greater than maximum limit of 5)
  => User MUST Deselect images
- **Confirmation** button is disabled

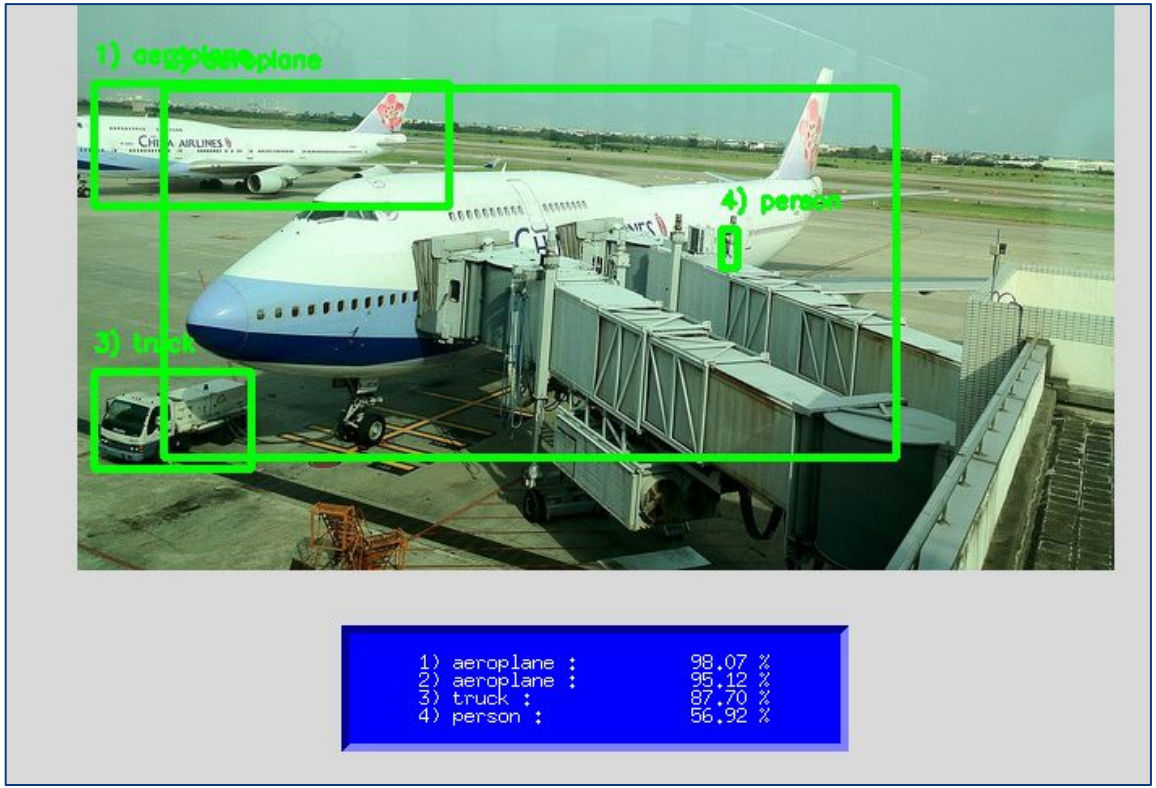
User had clicked Enlarge button

STAATLICH ANERKANNTE HOCHSCHULE

- User clicked Enlarge button to see full size image
- Image path displayed
- Option to click button to perform one-off inference with same detector used to populate database.

Enlarged image window - early version

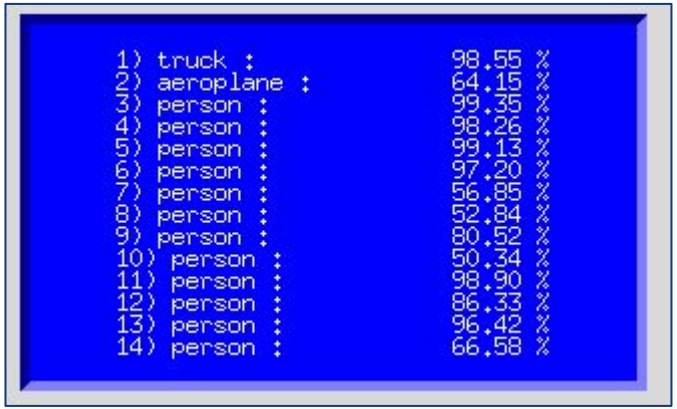
STAATLICH
ANERKANNTE
HOCHSCHULE



- User clicked button to perform object detection inference and can study output image in new window
- Bounding boxes shown with serial number matching blue box information
- **Earlier logic snapshot:**
  - > **Without min. HAS relationship score check. Thus could return image with no check of the all keyword object scores**
  - > **Note: Truck = 87% and Person = 56%**

Enlarged image window - current version

STAATLICH
ANERKANNTE
HOCHSCHULE



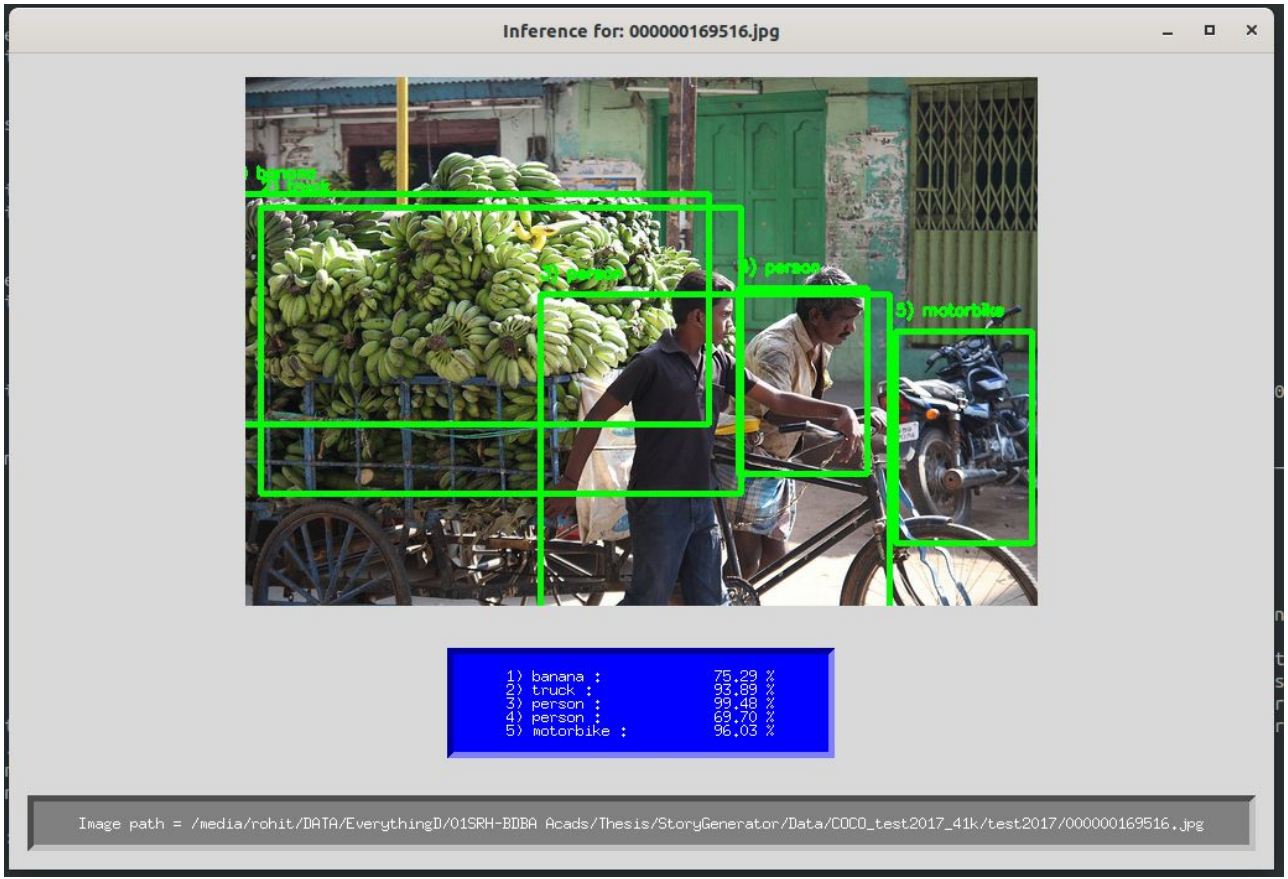
Updated logic:

Checks HAS relationship score > 0.90 for the objects used in query.

Note: At least one object of “Truck” and “Person” have scores > 90%

Detection can still fail !

STAATLICH ANERKANNTE HOCHSCHULE

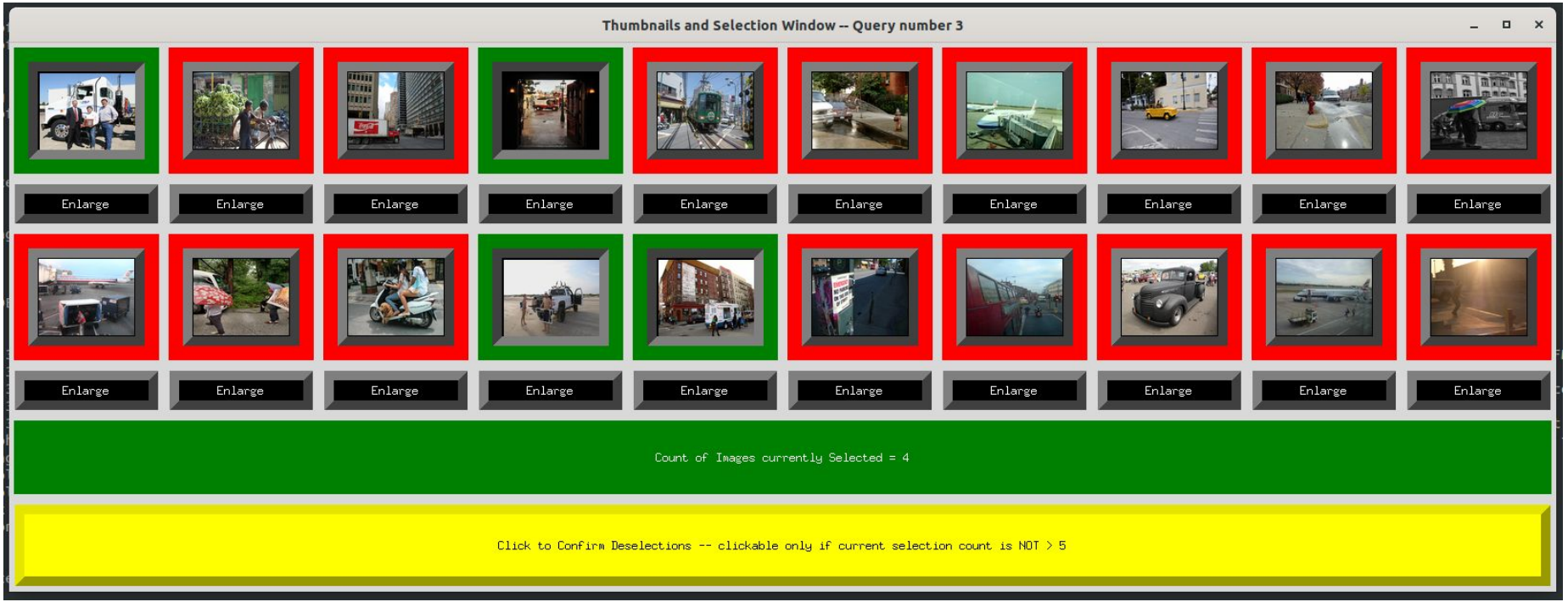


- **Completely wrong image:**
  - > There is no “truck”.
  - > Model scores 93% confidence for “truck”.
- Despite setting minimum HAS score = 90% during Neo4j query, such cases will not be tackled.
- ***GUI selection by user is crucial!***

User Deselected images and Ready

STAATLICH
ANERKANNTE
HOCHSCHULE

- Final Selections for 4 images
- Now the Confirm Deselections button is clickable



Data structure changes in backend

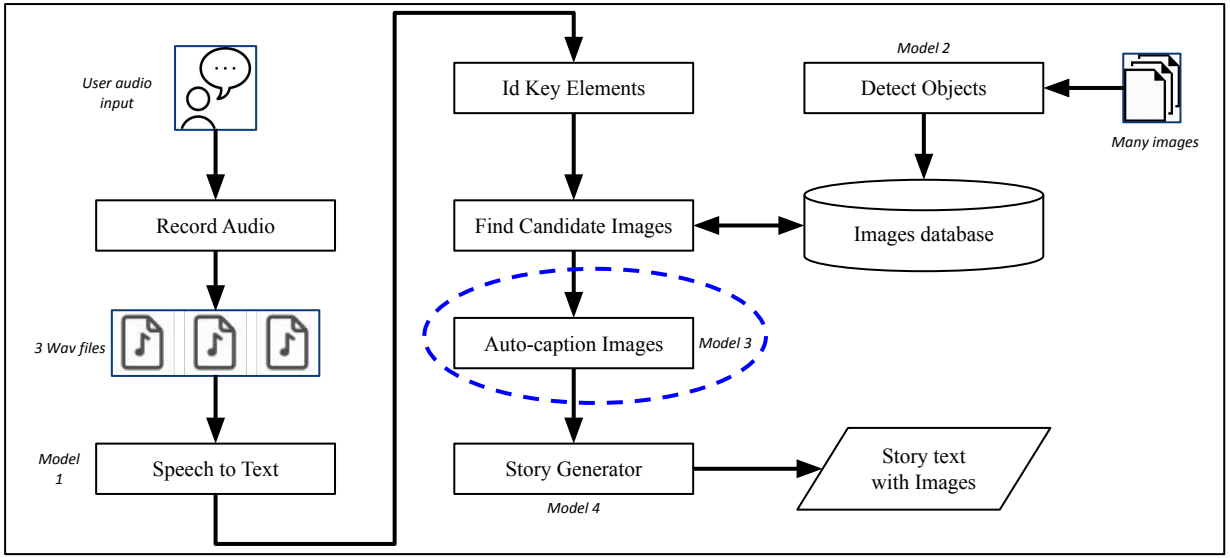
STAATLICH ANERKANNTE HOCHSCHULE

- Data structure before and after selection process:
  - > User deselected 16 of 20 images. So 4 retained and passed to Auto-captions Block

```
For Query 3
Number of candidate images before selection = 20
Number of Deselections done = 16
Number of images remaining after Deselections = 4

        ------ Query images info BEFORE::
[{'Image': '000000169542.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000169516.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000292186.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000146747.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000313777.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000449668.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000509771.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000012149.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000168815.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000168743.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000518174.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000017467.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000581864.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000225580.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000265504.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000361201.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000304424.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000225081.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000225051.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000499699.jpg', 'Source': 'coco_test_2017'}]

        ------ Positions removed::
[1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19]

        ------ Query images info AFTER::
[{'Image': '000000169542.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000146747.jpg', 'Source': 'coco_test_2017'},
 {'Image': '000000225580.jpg', 'Source': 'coco_test_2017'}, {'Image': '000000265504.jpg', 'Source': 'coco_test_2017'}]
```

STAATLICH
ANERKANNTE
HOCHSCHULE

Stage 4: Auto-caption Images



Goal and Introduction

STAATLICH ANERKANNTE HOCHSCHULE

- **Goal**: Describe an image with a natural language sentence
- Three broad methods: Paper "A survey of deep neural network-based image captioning" : https://doi.org/10.1007/s00371-018-1566-y

  > **Retrieval** aka Ranking based
    > Not used as outdated method.
    > Essentially find images in repository that are "close to" input image. Use the captions of the "close" images to create prediction. Heavily dependant on the breadth of repository

  > **Template** based
    > Two tasks carried out: Object detection to find bag of likely words. Then, create templates with the likely words and uses an LM to generate the prediction.
    > Limitations: does not really understand overall image and only tries to combine objects detected in meaningful way based on preset templates.

  > **End-to-End** Learning based:
    > Use a Deep-CNN to create feature vector to represent image.
    > Use this vector as during Decoding stage that are based on RNNs to output predicted caption
    > Two methods to combine the description and image: *Inject vs Merge*
    > "*Attention Mechanism*": Focuses attention on specific part of image at different stages of prediction timesteps by learning "Context mapping"

General architecture of Caption generator

STAATLICH ANERKANNTE HOCHSCHULE

- **Input** is a Jpeg image.
- **Output** is a sentence describing the image i.e. the caption.
- **Encoder**: Processes the input image into a numerical vector of features that the Decoder can then process.
  - > Typically a pre-trained model used.
  - > Tap last layer just before final softmax of a Deep-CNN classifier network
- **Decoder**: Processes the input image feature vector to predict the caption one word at a time.
  - > *After training*: Process the Image encoding to output the Predicted caption
  - > *During training*: Process the image encoding + Input ground truth captions to learn correct predictions

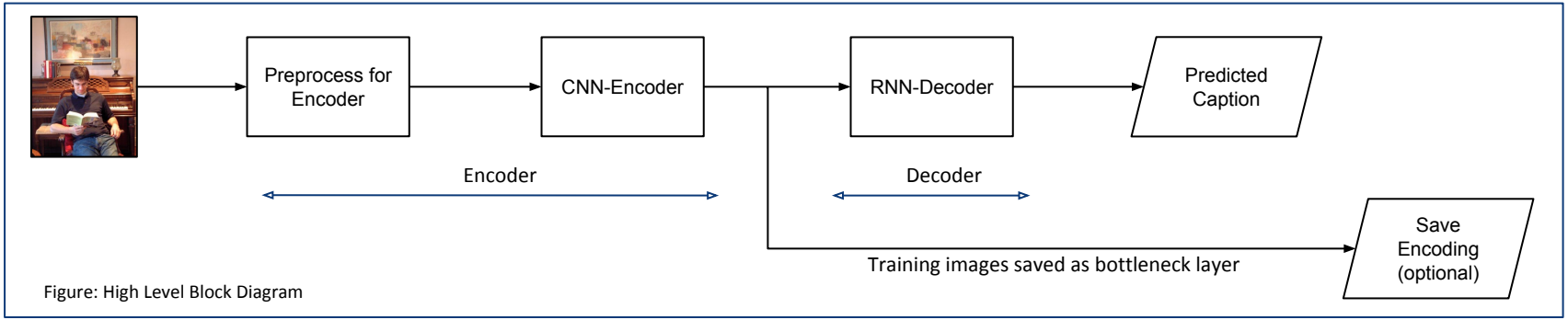


Figure: High Level Block Diagram

Two approaches used for Decoder

STAATLICH ANERKANNTE HOCHSCHULE

- ***Approach 1***: No attention decoder model:
  - > Followed this link: https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8
  - > Architecture based on paper: “Show and Tell: A Neural Image Caption Generator”
    - > Paper link: https://arxiv.org/abs/1411.4555
  - > Less complex:
    - > 256 LSTM cells as RNN units
    - > Merge architecture

- ***Approach 2***: WITH attention decoder model
  - > Followed this link: https://www.tensorflow.org/tutorials/text/image_captioning
  - > Architecture based on paper: “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”
    - > Paper link: https://arxiv.org/abs/1502.03044
  - > More complex:
    - > 512 GRU cells as RNN units
    - > Inject architecture
    - > Bahdanau attention

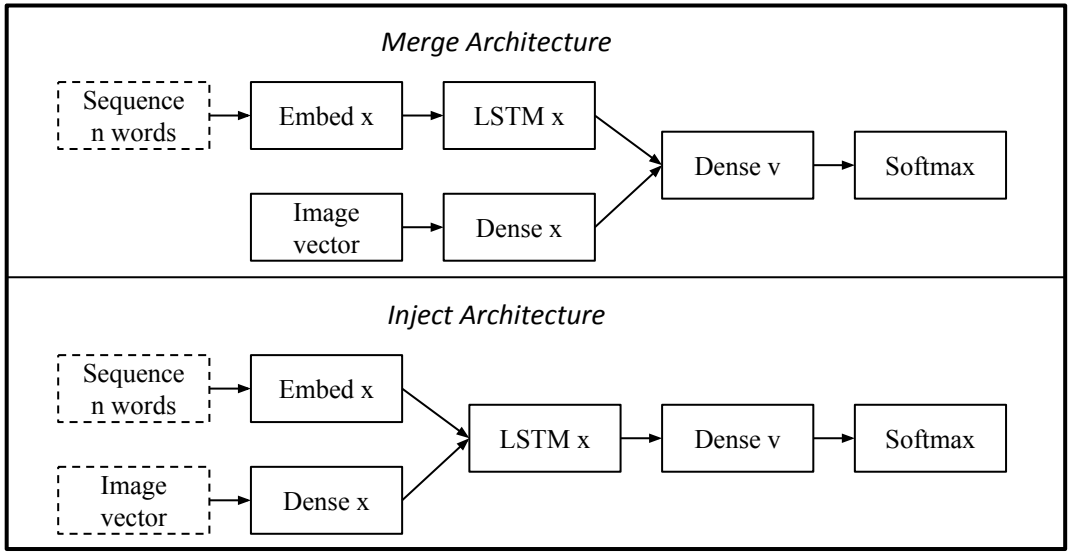
- Same encoder used in both cases:
  - > Google Inception v3 model pre-trained on Imagenet data.
  - > Tapped last layer before final softmax to get low-dimensional representation of image i.e. Encoding
  - > Encoding size = 2048 float values vector

Merge v/s Inject debate?

STAATLICH ANERKANNTE HOCHSCHULE

- Should Image features influence the RNN Language model (***Inject***) OR should they be combined later on in the process (***Merge***)?
- Both work but *Merge is much less memory and processing intensive - so is advantageous*.
  - > "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator" by Tanti et al. 2017
  - > "Where to put the Image in an Image Caption Generator" by M. Tanti et al. 2018

Source: "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator" by Tanti et al. 2017 (https://arxiv.org/abs/1708.02043)



- Approach 1: *No Attention*
  - > Model Type used:
    End-to-End, Merge architecture

- Approach 2: *With Attention*
  - > Model Type used:
    End-to-End, Inject architecture,
    Bahdanau attention model

Decoder model Parameters

STAATLICH ANERKANNTE HOCHSCHULE

- Two types Decoders tried: *Without* and *With* Attention mechanism

Decoder Model Parameters		
Parameter	**Model Type**	**Parameter Description**
Vocab_Size	Both	Number of words model can output - also called Vocabulary
Embedding_Dims	Both	Size of the vector representing each word as per embeddings type chosen
Embedding_Matrix_Shape	Both	Shape of the embedding matrix = (VOCAB_SIZE, EMBEDDING_DIMS)
Max_Length_Caption	Both	Maximum length of ground truth description with “startseq” and “endseq”
Unique words in Vocab	Both	Total unique words in all the ground truth captions
Unique words (high frequency)	Both	Total unique words in the model can handle
High Frequency Decision	Both	With attention: occurs > 10 times ; Without Attention: top 5000 by frequency
Number of RNN Units	Both	Number of RNN cells in the model
Attention Size	only With type	Attention span of the model

Decoder model Parameter - values

STAATLICH ANERKANNTE HOCHSCHULE

- Two types Decoders tried: *Without* and *With* Attention mechanism

Decoder Model Parameters Values		
Parameter	**Approach 1: Without Attention**	**Approach 2: With Attention**
Vocab_Size	6758	5001
Embedding_Dims	200	256
Embedding_Matrix_Shape SHAPE	(6758, 200)	(5001, 256)
Max_Length_Caption	49	52
Unique words in Vocab	24323	28062
Unique words (high frequency)	6757	5000
Number of RNN Units	256	512
Attention Size	NA	10

Data for Decoder training and evaluation

STAATLICH ANERKANNTE HOCHSCHULE

MS COCO 2017 Dataset - 5 ground truth captions per image		
Dataset	**No. of Images**	**Captions?**
Train	118k	Yes
Val	5k	Yes
Test	41k	No

Usage in the thesis work:

- Approach 2 (With attention) training time very long and times out. Therefore no validation set used during training.

Data splits for Approach 1: No Attention			
Original Source	**Dataset type during training**	**No. of Images**	**Descriptions?**
Coco2017_Train	Train	97k	Yes
Coco2017_Train	Validation	3k	Yes
Coco2017_Val	Test	5k	Yes

Data splits for Approach 2: With Attention			
Original Source	**Dataset type during training**	**No. of Images**	**Descriptions?**
Coco2017_Train	Train	100k	Yes
-	Validation	-	-
Coco2017_Train	Test	5k	Yes

STAATLICH
ANERKANNTE
HOCHSCHULE

Approach 1: No Attention Model

Details

Data for model training

- Image Feature (*ImgFeat*) = 2048 vector from Encoder
- Description = cleaned ground truth sentence with insertion of special tokens for start and end
- During preprocessing:
  - > build the “Wordtoix” and “IxtoWord” to map between work tokens and unique integer representation
  - > Find the maximum length of ground truth description and decide the value for MAX_LENGTH_CAPTION
- Each image feature and its caption is converted to a series of training inputs
  - > Xt = vector of words represented as integer with 0-pad
  - > Yt = expected word prediction as integer
  - > Input = ImgFeat + Xt

Image 1	Ground Truth caption 1: “A dog is barking outside.”			
	Cleaned + special tokens: “startseq dog is barking outside endseq”			
	Training data Length before equalising = 6			
Image 2	Ground Truth caption 2: “That is a sweater.”			
	Cleaned + special tokens: “startseq that is sweater endseq”			
	Training data Length before equalising = 5			
Max Length of Caption: Parameter value chosen = 8				
SN	LogicalData	ImgFeat	Xt	Yt
1	1	imgF1	startseq	dog
2	1	imgF1	startseq dog	is
3	1	imgF1	startseq dog is	barking
4	1	imgF1	startseq dog is barking	outside
5	1	imgF1	startseq dog is barking outside	endseq
6	2	imgF2	startseq	that
7	2	imgF2	startseq that	is
8	2	imgF2	startseq that is	sweater
9	2	imgF2	startseq that is sweater	endseq
Above word inputs get mapped using the wordtoix data structure and 0 padding				
1	1	imgF1	1 0 0 0 0 0 0 0	11
2	1	imgF1	1 11 0 0 0 0 0 0	12
3	1	imgF1	1 11 12 0 0 0 0 0	13
4	1	imgF1	1 11 12 13 0 0 0 0	14
5	1	imgF1	1 11 12 13 14 0 0 0	9
6	2	imgF2	1 0 0 0 0 0 0 0	21
7	2	imgF2	1 21 0 0 0 0 0 0	12
8	2	imgF2	1 21 12 0 0 0 0 0	23
9	2	imgF2	1 21 12 23 0 0 0 0	9

Figure: Data preparation for decoder training

Approach 1: No Attention: Detailed architecture

STAATLICH ANERKANNTE HOCHSCHULE

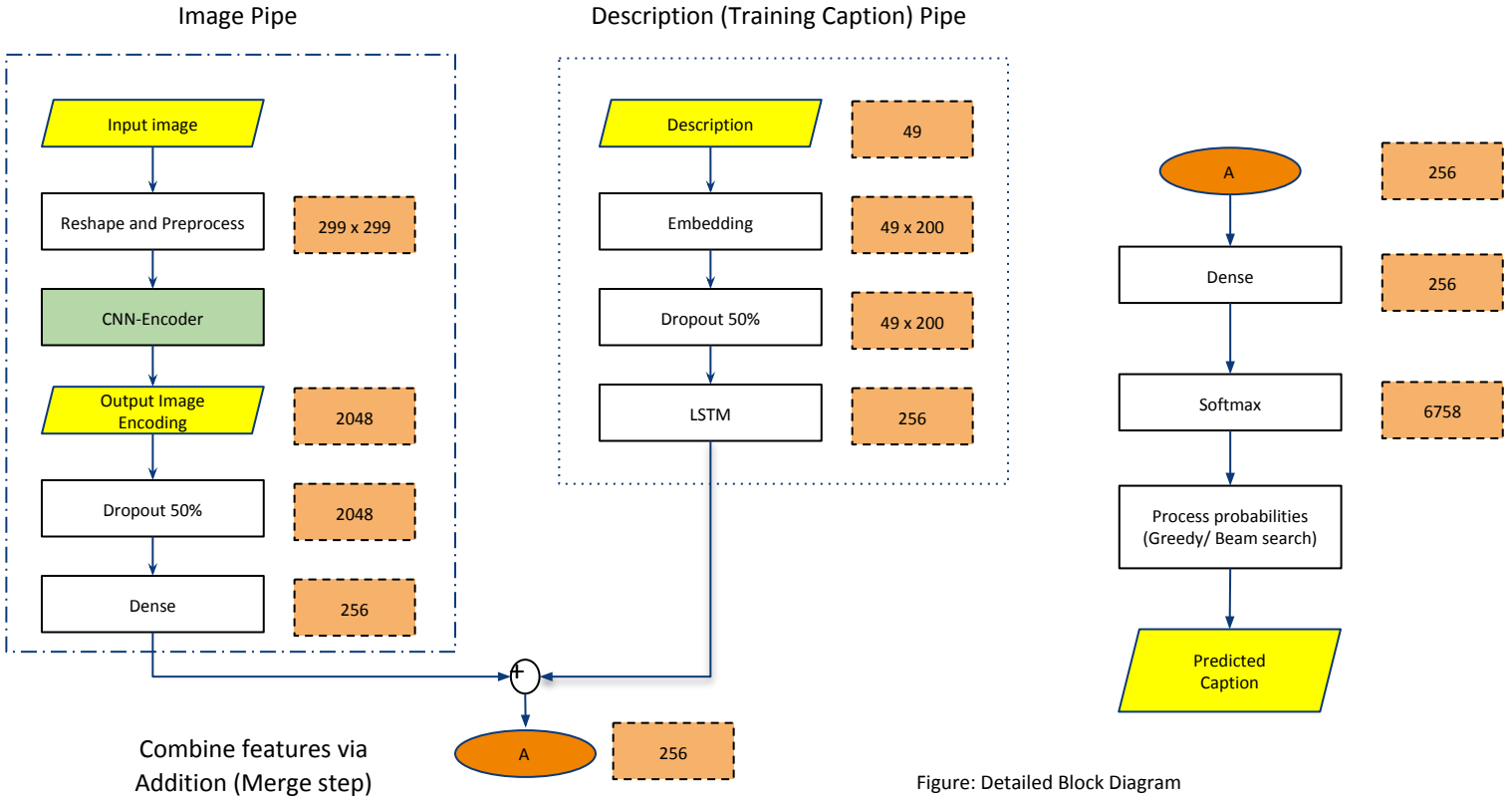


Figure: Detailed Block Diagram

Preprocessing steps

STAATLICH
ANERKANNTE
HOCHSCHULE

- Load descriptions from the “annotations json file” and cleanup sentences:
  - Lowercase
  - Remove punctuations and special characters (including the period at end of descriptions)
  - Remove words with length = 1
    - Prevent orphan letters causing a mess. E.g. punctuation removal from “there’s a great” -> “there s a great” -> “there great”
    - unfortunately legitimate “a” will also be dropped
  - Drop non-alphabet words. E.g. gr8

- Insert the special tokens for start and end of descriptions:
  - “startseq” and “endseq” - required only for the training data

- Use this data to:
  - Create the full Vocabulary i.e. find all the unique words from the descriptions
  - Based on chosen threshold for “high frequency words”, cull the full Vocabulary to the one to actually use for model
  - Using the Culled Vocabulary, calculate the Maximum Length of Caption

- Create the data structures for “wordtoix” and “ixtoword” - mapping string tokens to unique integer representations

- Create embeddings matrix based on culled vocabulary:
  - using GloVe-200 consisting of 400,000 words

Preprocessing - example of cleanup effect

Clean up descriptions for a random image

Note the change before and after

```
In [24]: ## example of caption with accidental newline \n in the caption
         descriptions_test['000000482917']

Out[24]: ['A dog sitting between its masters feet on a footstool watching tv\n',
          'A dog between the feet of a person looking at a TV.',
          'A dog and a person are watching television together.',
          'A person is sitting with their dog watching tv.',
          'A man relaxing at home, watching television with his dog.']

In [25]: # prepare translation table for removing punctuation
         ## string.punctuation  gives   '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~' and will take care of all these characters being made in
         to a space
         tran_table = str.maketrans(string.punctuation, ' ' * len(string.punctuation))
         for key, desc_list in descriptions_test.items():
             for idx in range(len(desc_list)):
                 desc = desc_list[idx]
                 # replace all punctuation with space in description before tokenizing
                 desc = desc.translate(tran_table)
                 # tokenize
                 desc = desc.split()
                 # convert to lower case
                 desc = [word.lower() for word in desc]
                 # remove hanging 's' and 'a'
                 desc = [word for word in desc if len(word)>1]
                 # remove any non-alphabetic tokens
                 desc = [word for word in desc if word.isalpha()]
                 # overwrite with cleaned description
                 desc_list[idx] =  ' '.join(desc)

In [26]: ## example of caption with accidental newline \n in the caption  -- POST CLEANUP
         descriptions_test['000000482917']

Out[26]: ['dog sitting between its masters feet on footstool watching tv',
          'dog between the feet of person looking at tv',
          'dog and person are watching television together',
          'person is sitting with their dog watching tv',
          'man relaxing at home watching television with his dog']
```

Model Parameters - Code Snippets

STAATLICH ANERKANNTE HOCHSCHULE

Sizes of the descriptions and the Encodings for Train and Val data = 97k and 3k respectively

[15]:
```
print(f"Encodings data:\nlen(img_encodings_train) = {len(img_encodings_train)}\t\tlen(img_encodings_val) = {len(img_encodings_val)}")
print(f"Descriptions data:\nlen(descriptions_train) = {len(descriptions_train)}\t\tlen(descriptions_val) = {len(descriptions_val)}")
print(f"\nCHECK : reloaded values = 97k for Train , 3k for Validation")
```

```
Encodings data:
len(img_encodings_train) = 97000                len(img_encodings_val) = 3000
Descriptions data:
len(descriptions_train) = 97000         len(descriptions_val) = 3000

CHECK : reloaded values = 97k for Train , 3k for Validation
```

Calculate the Total Unique words in vocabulary based on descriptions = 24323

[16]:
```
## at this stage the descriptions_train already has the start and end tokens added to it
vocabulary = set()
for key in descriptions_train.keys():
    [vocabulary.update(d.split()) for d in descriptions_train[key]]
print(f"Original Vocabulary Size with all words = {len(vocabulary)}")
print(f"\nCHECK : reloaded value = 24323")
```

```
Original Vocabulary Size with all words = 24323

CHECK : reloaded value = 24323
```

Model Parameters - Code Snippets

STAATLICH ANERKANNTE HOCHSCHULE

Calculate the High Frequency words = 6757

[17]:

```
# Create a list of all the training captions, find the freq and retain words where the freq > threshold chosen

all_desc_in_training_samples = []
for key, val in descriptions_train.items():
    for cap in val:
        all_desc_in_training_samples.append(cap)

MIN_WORD_COUNT_THRESHOLD = 10
word_counts = {}
nsents = 0
for each_desc in all_desc_in_training_samples:
    nsents += 1
    for w in each_desc.split(' '):
        word_counts[w] = word_counts.get(w, 0) + 1

vocab_threshold = [w for w in word_counts if word_counts[w] >= MIN_WORD_COUNT_THRESHOLD]

print(f"Culled vocabulary to only retain words occurring more than threshold = {MIN_WORD_COUNT_THRESHOLD} times.\nNew vocab size , len(vocab_threshold) = {len(vocab_threshold)}")
print(f"\nCHECK : reloaded value = 6757")
```

```
Culled vocabulary to only retain words occurring more than threshold = 10 times.
New vocab size , len(vocab_threshold) = 6757

CHECK : reloaded value = 6757
```

Model Parameters - Code Snippets

STAATLICH
ANERKANNTE
HOCHSCHULE

Calculate Max. caption length = 49

[18]:

```
## determine the maximum sequence length - parameter MAX_LENGTH_CAPTION used during the RNN deocder model setup

## convert a dictionary of clean descriptions to a list of descriptions
def extract_each_desc(_descriptions):
    all_desc = list()
    for key in _descriptions.keys():
        [all_desc.append(d) for d in _descriptions[key]]
    return all_desc

## find the longest description length
def find_max_length_desc(_descriptions):
    desc_sentences = extract_each_desc(_descriptions)
    return max(len(d.split()) for d in desc_sentences)

MAX_LENGTH_CAPTION = find_max_length_desc(descriptions_train)  ## will be used directly later while defining Decoder model
print(f"Max Description Length: {MAX_LENGTH_CAPTION}")
print(f"\nCHECK : reloaded value = 49")
```

```
Max Description Length: 49

CHECK : reloaded value = 49
```

Model Parameters - Code Snippets

STAATLICH ANERKANNTE HOCHSCHULE

Set VOCAB_SIZE using the “wordtoix” or “ixtoword” data length + 1

Additional 1 for the 0 pad token

[19]:

```
## the value now, as it will be used as:: VOCAB_SIZE = len(wordtoix) + 1
print(f"len(wordtoix) = {len(wordtoix)}")
print(f"\nCHECK : reloaded value = 6757")

VOCAB_SIZE = len(wordtoix) + 1
print(f"\n\nSet the   VOCAB_SIZE = len(wordtoix) + 1 = {VOCAB_SIZE}")

EMBEDDING_DIMS = 200
print(f"\n\nSet the   EMBEDDING_DIMS = {EMBEDDING_DIMS}")
```

```
len(wordtoix) = 6757

CHECK : reloaded value = 6757


Set the   VOCAB_SIZE = len(wordtoix) + 1 = 6758


Set the   EMBEDDING_DIMS = 200
```

See the indices in “wordtoix” for special tokens and some random word

[20]:

```
## see the index output by wordtoix for the start and end sequence tokens as well as some random one word
print( wordtoix.get('startseq') , wordtoix.get('endseq') , wordtoix.get('cat') )
```

```
1 9 526
```

Model Comparison - Training Parameters

STAATLICH ANERKANNTE HOCHSCHULE

Training Hyperparameters

Model 1			
#Ep	**Ep (From-To)**	**LR**	**BS**
2	1-2	0.0005	128
5	3-7	0.0002	128
3	8-10	0.0001	64

Model 2			
#Ep	**Ep (From-To)**	**LR**	**BS**
13	1-13	0.001	64
2	14-15	0.001	128
3	16-18	0.0005	32

- Models trained with different hyperparameters
- Common Parameters:
  - > Optimizer = Adam
  - > Loss function = Categorical cross-entropy

Legend	
Ep	Epoch
#Ep	Number of Epochs
LR	Learning Rate
BS	Batch Size

Model 2 losses lower so *Better model*

STAATLICH ANERKANNTE HOCHSCHULE

Training and Validation Set Losses for Model 1 v/s Model 2 : ***Chosen Model 2 to proceed***
Train dataset = 97k images, Validation dataset = 3k images

Ep	Model 1 Losses		Model 2 Losses	
	Train	Val	Train	Val
2	3.4819	3.52842	3.1985	3.27492
4	3.2286	3.36663	3.0032	3.17518
6	3.1432	3.29084	2.9315	3.14292
8	3.0858	3.24664	2.8673	3.12330
10	**3.0448**	**3.21695**	**2.8443**	**3.11437**
12	-	-	2.8397	3.11331
14	-	-	2.8013	***3.08822***
16	-	-	2.7977	3.09369
18	-	-	**2.7704**	**3.08865**

What we see:

Train and Val losses: Consistent fall with epochs - *no overfitting*

After 10 epochs Model 2 is better

About Batch sizes:
Training: BS varies
Val: BS = 64 always

Inconsistent: Possibly wrong weights file or just chance

Common Parameters:

Optimizer = Adam

Loss function = Categorical cross-entropy

Training Losses: BS varies
Val Losses: BS = 64

Legend	
Ep	Epoch
LR	Learning Rate
BS	Batch Size

Model Evaluation with Bleu Scores

STAATLICH ANERKANNTE HOCHSCHULE

- Bleu scorer of NLTK used:
  - > import nltk.translate.bleu_score as nltk_bleu
  - > Bleu score = nltk_bleu.sentence_bleu([list of GT descriptions] , “the predicted caption from model”)

- Some random lowest and highest scores from both models below

Model 1 - After 10 epochs

```
In [12]: dfbs.head()
```

Out[12]:

	img	infcap	bsnltk
2294	000000484760	clock tower with clock on it	8.580523e-155
2299	000000485130	bed with two beds and two beds	3.831503e-78
819	000000197870	bird perched on the ground next to bird	4.351978e-78
1959	000000015517	train traveling down bridge next to bridge	4.815777e-78
2461	000000439522	man in black jacket and black jacket and black...	5.385075e-02

```
In [13]: dfbs.tail()
```

Out[13]:

	img	infcap	bsnltk
1702	000000149770	man riding surfboard on top of wave	1.0
3635	000000199442	man riding wave on top of surfboard	1.0
1663	000000450303	group of people sitting around table with laptops	1.0
611	000000383606	bathroom with sink and mirror	1.0
3581	000000320696	man riding wave on top of surfboard	1.0

Model 2 - After 10 epochs

```
In [14]: dfbs.head()
```

Out[14]:

	img	infcap	bsnltk
493	000000484760	clock tower with clock on it	8.580523e-155
2270	000000112626	room with two beds and chair	9.746048e-155
1762	000000507667	an old model model model model fighter jet	2.845685e-78
4200	000000468233	an old model model model cell phone	3.516915e-78
2084	000000453860	an open suitcase with an open door open	3.775819e-78

```
In [15]: dfbs.tail()
```

Out[15]:

	img	infcap	bsnltk
1412	000000373705	red fire hydrant sitting on the side of road	1.0
3401	000000325347	man holding tennis racquet on tennis court	1.0
3410	000000223959	man holding tennis racquet on tennis court	1.0
689	000000466835	bunch of bananas hanging from tree	1.0
4813	000000462031	baseball player holding bat on top of field	1.0

Bleu Scores after 10 epochs

STAATLICH ANERKANNTE HOCHSCHULE

- Bleu Scores after 10 epochs:
  - > only Greedy Search used
  - > Test dataset = 5k images
- Model 2 Scores **slightly higher**: Average by 0.12% and Median by 0.06%
- Standard deviation and relative frequencies almost same
- Overall Model 2 is slightly better than Model 1 : ***continued training Model 2 for more epochs, abandoned Model 1***

Bleu Scores after 10 epochs: Model 1 v/s 2		
Bleu Scores	**Model 1**	**Model 2**
Max	1.0	1.0
Min	0.0	0.0
Median	0.6372	0.6376
Average	0.6327	0.6335
Std. Dev.	0.1643	0.1648

Bleu Scores by Bins – frequency comparison – 5k data points Greedy Search used overall						
	Model 1			Model 2		
Score Bin	**Count**	**Rel. Freq**	**Cum. Freq**	**Count**	**Rel. Freq**	**Cum. Freq**
0.0 – 0.1	9	0.00	0.00	6	0.00	0.00
0.1 – 0.2	26	0.01	0.01	24	0.00	0.01
0.2 – 0.3	82	0.01	0.02	90	0.02	0.02
0.3 – 0.4	294	0.06	0.08	271	0.05	0.08
0.4 – 0.5	664	0.13	0.21	676	0.14	0.21
0.5 – 0.6	977	0.20	0.41	978	0.20	0.41
0.6 – 0.7	1175	0.24	0.65	1190	0.24	0.65
0.7 – 0.8	965	0.19	0.84	948	0.19	0.84
0.8 – 0.9	565	0.12	0.95	572	0.12	0.95
0.9 – 1.0	217	0.05	1.00	203	0.04	1.00
Total	4974	-	-	4958	-	-

Bleu Scores - Model 2 after more epochs

STAATLICH ANERKANNTE HOCHSCHULE

- Bleu Scores for Model 2 after after 10 vs 18 epochs:
  - > only Greedy Search used
  - > Test dataset = 5k images
- 18 Epoch Scores **higher**: Average by 1.44% and Median by 1.87%
- Standard deviations and relative frequencies similar - but improvements in 0.6 to 0.8 range (see Count)
- Overall Model 2 after 18 epochs is better : ***Model 2 with 18 epochs used***

Bleu Scores Model 2: After 10 v/s 18 epochs		
Bleu Scores	**10 Ep**	**18 Ep**
Max	1.0	1.0
Min	0.0	0.0
Median	0.6376	0.6495
Average	0.6335	0.6426
Std. Dev.	0.1648	0.1629

Bleu Scores by Bins – frequency comparison – 5k data points Greedy Search used overall						
	Model 2 – 10 Epochs			Model 2 – 18 Epochs		
Score Bin	**Count**	**Rel. Freq**	**Cum. Freq**	**Count**	**Rel. Freq**	**Cum. Freq**
0.0 – 0.1	6	0.00	0.00	2	0.00	0.00
0.1 – 0.2	24	0.00	0.01	12	0.00	0.00
0.2 – 0.3	90	0.02	0.02	81	0.01	0.02
0.3 – 0.4	271	0.05	0.08	169	0.05	0.07
0.4 – 0.5	676	0.14	0.21	609	0.12	0.19
0.5 – 0.6	978	0.20	0.41	988	0.20	0.39
0.6 – 0.7	1190	0.24	0.65	959	0.23	0.62
0.7 – 0.8	948	0.19	0.84	1028	0.20	0.83
0.8 – 0.9	572	0.12	0.95	570	0.12	0.94
0.9 – 1.0	203	0.04	1.00	230	0.05	1.00
Total	4958	-	-	4648	-	-

Inference method - Greedy or Beam Search ?

STAATLICH ANERKANNTE HOCHSCHULE

- Evaluated Greedy Search v/s Beam Search
  - > Tried with Beam Widths = 3 and 5
- Comparing Model 2 after 18 epochs on Bleu Scores
  - > Test dataset = 5k images
- Scores for Greedy slightly better.
- Scores worst for Beam search with Width = 5
  - > Also inference time substantially longer
- ***Could use Greedy or Beam (3) search***

Bleu Scores Model 2: Greedy v/s Beam search			
Bleu Scores	**Greedy**	**Beam (3)**	**Beam (5)**
Max	1.0	1.0	1.0
Min	0.0	0.0	0.0
Median	0.6495	0.6431	0.6270
Average	0.6426	0.6385	0.6235
Std. Dev.	0.1629	0.1600	0.1617

Inference method - Bleu scores comparison

STAATLICH ANERKANNTE HOCHSCHULE

- Bleu scores for Model 2 after 18 epochs:
  - > Test dataset = 5k images
- NLTK able to score additional 300 inferences with Beam search:
  - > Width=3 results: mostly with 0.6-0.7 scores
- But generally similar results

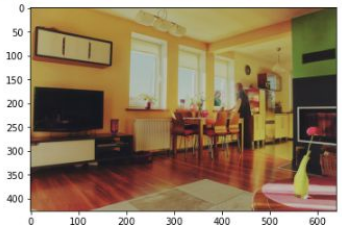
- ***Implemented Beam search with Width = 3 in thesis work***

Bleu Scores by Bins – frequency comparison – 5k data points									
Model 2 – 18 Epochs – Comparison Greedy / Beam Search									
	Greedy			Beam Search (Width = 3)			Beam Search (Width = 5)		
Score Bin	Count	Rel. Freq	Cum. Freq	Count	Rel. Freq	Cum. Freq	Count	Rel. Freq	Cum. Freq
0.0 – 0.1	2	0.00	0.00	4	0.00	0.00	6	0.00	0.00
0.1 – 0.2	12	0.00	0.00	9	0.00	0.00	14	0.00	0.00
0.2 – 0.3	81	0.01	0.02	91	0.02	0.02	112	0.03	0.03
0.3 – 0.4	169	0.05	0.07	272	0.05	0.07	323	0.06	0.09
0.4 – 0.5	609	0.12	0.19	590	0.12	0.19	661	0.13	0.22
0.5 – 0.6	988	0.20	0.39	999	0.20	0.39	1031	0.21	0.43
0.6 – 0.7	959	0.23	0.62	1235	0.25	0.64	1201	0.24	0.67
0.7 – 0.8	1028	0.20	0.83	1008	0.21	0.85	964	0.19	0.86
0.8 – 0.9	570	0.12	0.94	565	0.11	0.96	495	0.10	0.96
0.9 – 1.0	230	0.05	1.00	189	0.04	1.00	159	0.04	1.00
Total	4648	-	-	4962	-	-	4966	-	-

Greedy v/s Beam (width=3) inference examples

STAATLICH ANERKANNTE HOCHSCHULE

```
Inferrring image 1 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/D

Cleaned descriptions:
Orig Desc 1 :: woman stands in the dining area at the table
Orig Desc 2 :: room with chairs table and woman in it
Orig Desc 3 :: woman standing in kitchen by window
Orig Desc 4 :: person standing at table in room
Orig Desc 5 :: living area with television and table
```

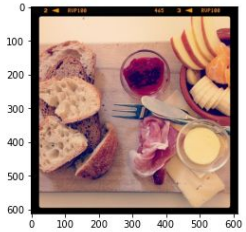


```
Greedy caption :: living room with couch and television
Beam3 caption :: living room filled with furniture and flat screen tv
```

```
Inferrring image 2 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/Data/C

Cleaned descriptions:
Orig Desc 1 :: big burly grizzly bear is show with grass in the background
Orig Desc 2 :: the large brown bear has black nose
Orig Desc 3 :: closeup of brown bear sitting in grassy area
Orig Desc 4 :: large bear that is sitting on grass
Orig Desc 5 :: close up picture of brown bear face
```

```
Greedy caption :: brown bear walking across lush green field
Beam3 caption :: brown bear standing on top of grass covered field
```

```
Inferrring image 36 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/D

Cleaned descriptions:
Orig Desc 1 :: bread and fruit are on table with knife and fork
Orig Desc 2 :: some bread ham fruit jam and glass of orange juice
Orig Desc 3 :: an assortment of meats and cheeses with bread
Orig Desc 4 :: variety of different types of food on table
Orig Desc 5 :: selection of breads meat spreads and cheeses
```



```
Greedy caption :: plate of food with some fruit and vegetables
Beam3 caption :: table topped with plates of food and drink
```

```
Inferrring image 4 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/Data/COCO

Cleaned descriptions:
Orig Desc 1 :: stop sign is mounted upside down on it post
Orig Desc 2 :: stop sign that is hanging upside down
Orig Desc 3 :: an upside down stop sign by the road
Orig Desc 4 :: stop sign put upside down on metal pole
Orig Desc 5 :: stop sign installed upside down on street corner
```

```
Greedy caption :: stop sign is shown with graffiti on it
Beam3 caption :: red stop sign sitting on the side of the road
```

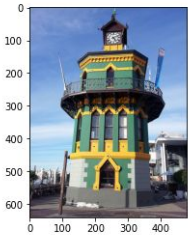
From random outputs for 100 images of Test dataset:

nbViewwer link:
https://nbviewer.jupyter.org/github/rbewoor/thesis/blob/master/ImgCap/BleuScoreAccuracy/demo_ImgCap_Reload_Lappy_Infer_Greedy_Beam3_1.ipynb

Greedy v/s Beam (width=3) inference examples

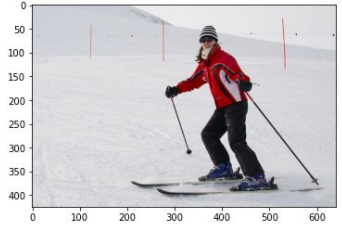
STAATLICH ANERKANNTE HOCHSCHULE

```
Inferrring image 37 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/Data

Cleaned descriptions:
Orig Desc 1 :: clock is situated atop colorful tower
Orig Desc 2 :: tall green clock tower sitting in the middle of sidewalk
Orig Desc 3 :: colourful building has clock at the top
Orig Desc 4 :: tall green and yellow building with clock at the top
Orig Desc 5 :: large green building with clock at the top of it
```



```
Greedy caption :: red and white clock tower with clock on it
Beam3 caption :: red fire hydrant on the side of the road
```

```
Inferrring image 6 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/D

Cleaned descriptions:
Orig Desc 1 :: woman posing for the camera standing on skis
Orig Desc 2 :: woman standing on skiis while posing for the camera
Orig Desc 3 :: woman in red jacket skiing down slope
Orig Desc 4 :: young woman is skiing down the mountain slope
Orig Desc 5 :: person on skis makes her way through the snow
```



```
Greedy caption :: man in red jacket skiing down hill
Beam3 caption :: man riding skis on top of snow covered slope
```

```
Inferrring image 57 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/Data/COCO_val2017

Cleaned descriptions:
Orig Desc 1 :: grey colored jet plane flying over snowy mountain range
Orig Desc 2 :: silver jet is flying high in the sky above the clouds
Orig Desc 3 :: jet fighter flying through the sky above snow covered mountain
Orig Desc 4 :: an airplane with the swiss flag symbol is flying through the mountains
Orig Desc 5 :: an airplane flying solo above blue terrain
```

```
Greedy caption :: fighter jet flying through the air with smoke coming out of it
Beam3 caption :: fighter jet is flying through the sky
```

```
Inferrring image 24 of 100:
/media/rohit/DATA/EverythingD/01SRH-BDBA Acads/Thesis/StoryGenerator/Data/COCO_val2017_5k/v

Cleaned descriptions:
Orig Desc 1 :: group of three chefs preparing food in kitchen
Orig Desc 2 :: man making pizza in kitchen
Orig Desc 3 :: some people in kitchen preparing food on counter
Orig Desc 4 :: couple of guys cooking in restaruant kitchen that is open to the restaurant
Orig Desc 5 :: cooks are gathered preparing meal in the kitchen
```

```
Greedy caption :: man is cutting cake with knife
Beam3 caption :: man and woman preparing food in kitchen
```

From random outputs for 100 images of Test dataset:

nbViewwer link:
https://nbviewer.jupyter.org/github/rbewoor/thesis/blob/master/ImgCap/BleuScoreAccuracy/demo_ImgCap_Reload_Lappy_Infer_Greedy_Beam3_1.ipynb

STAATLICH
ANERKANNTE
HOCHSCHULE

Approach 2: With Attention Model

Details

Approach 2: No Attention: Detailed architecture PENDING

STAATLICH
ANERKANNTE
HOCHSCHULE

Model Parameters - Code Snippets

STAATLICH ANERKANNTE HOCHSCHULE

Top 5000 words as vocabulary and tokenizing the training captions.

```
In [49]: # Choose the top 5000 words from the vocabulary
         top_k = 5000
         tokenizer = tf.keras.preprocessing.text.Tokenizer(num_words=top_k,
                                                           oov_token="<unk>",
                                                           filters='!"#$%&()*+.,-/:;=?@[\]^_`{|}~ ')
         tokenizer.fit_on_texts(train_captions)
         train_seqs = tokenizer.texts_to_sequences(train_captions)

In [50]: type(train_seqs)

Out[50]: list

In [51]: train_seqs[:10]

Out[51]: [[3, 848, 6, 2804, 6, 61, 27, 1990, 242, 10, 437, 4],
          [3, 2, 430, 11, 3410, 8, 1024, 396, 498, 1139, 4],
          [3, 64, 20, 1038, 144, 9, 191, 948, 6, 735, 4],
          [3, 301, 727, 26, 346, 210, 264, 10, 437, 4],
          [3, 2, 172, 6, 1139, 27, 445, 191, 61, 4],
          [3, 2, 119, 113, 61, 97, 7, 33, 6, 7, 129, 4],
          [3, 2, 119, 18, 35, 679, 2, 129, 4],
          [3, 2, 119, 773, 9, 155, 206, 8, 7, 435, 4],
          [3, 16, 207, 18, 8, 2, 129, 144, 102, 4],
          [3, 2, 119, 18, 21, 13, 2, 435, 144, 9, 130, 4]]
```

Model Parameters - Code Snippets

STAATLICH ANERKANNTE HOCHSCHULE

Zero padding the training data to make length = maximum caption length of 52

```
In [60]: img_name_train[:2]

Out[60]: ['/kaggle/input/coco-2017-dataset/coco2017/train2017/000000000009.jpg',
          '/kaggle/input/coco-2017-dataset/coco2017/train2017/000000000009.jpg']

In [61]: cap_train[:2]

Out[61]: [array([   3,  848,    6, 2804,    6,   61,   27, 1990,  242,   10,  437,
                    4,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0],   dtype=int32),
           array([   3,    2,  430,   11, 3410,    8, 1024,  396,  498, 1139,    4,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
                    0,    0,    0,    0,    0,    0,    0],   dtype=int32)]
```

Model Parameters - Code Snippets

STAATLICH
ANERKANNTE
HOCHSCHULE

Parameters for With Attention Decoder model

In [69]:
```
# Feel free to change these parameters according to your system's configuration

if DUMMY_RUN_FLAG == True:
    BATCH_SIZE = 1
else:
    BATCH_SIZE = 128
print(f"BATCH_SIZE used = {BATCH_SIZE}")
BUFFER_SIZE = 250
embedding_dim = 256
units = 512
vocab_size = top_k + 1
num_steps = len(img_name_train) // BATCH_SIZE
# Shape of the vector extracted from InceptionV3 is (64, 2048)
# These two variables represent that vector shape
features_shape = 2048
attention_features_shape = 64
```

```
BATCH_SIZE used = 128
```

In [70]:
```
num_steps
```

Out[70]:
```
3908
```

Model - Training Parameters and Losses

STAATLICH ANERKANNTE HOCHSCHULE

Ep	Train Losses
1	0.676393
4	0.549900
7	0.521033
10	0.501702
13	0.487315
16	0.476580
19	0.468022
21	0.463329
22	0.461324

What we see:

Train losses continue falling

No validation losses to double-check no overfitting (due to run-time time-out constraints)

But improbable that over-fitting has occurred

Training Hyperparameters

#Ep	Ep (From-To)	LR	BS
22	1-22	0.001	128

- Optimizer = Adam
- Loss function = Categorical Cross-entropy

Legend	
Ep	Epoch
#Ep	Number of Epochs
LR	Learning Rate
BS	Batch Size

Model Evaluation with Bleu scores

STAATLICH ANERKANNTE HOCHSCHULE

- Bleu scorer of NLTK library used:
  - > Expects prediction caption and list of ground-truth captions for comparison
- Python usage:
  - > import nltk.translate.bleu_score as nltk_bleu
  - > Bleu score = nltk_bleu.sentence_bleu([list of GT descriptions] , “the predicted caption from model”)
- Some random lowest and highest scores from both models below

```
In [11]: dfbs.head()
```

Out[11]:

	img	infcap	bsnltk
3829	000000510910	a giraffe and antelope and antelope and antelo...	1.010159e-78
66	000000492154	a man and white and black and white and black ...	1.580031e-78
1839	000000500871	a knife and a knife and a knife and a knife	2.841683e-78
447	000000494003	a vase with a <unk> <unk>	3.134553e-78
3844	000000510980	a man with a cellphone	3.226530e-78

```
In [12]: dfbs.tail()
```

Out[12]:

	img	infcap	bsnltk
4557	000000514567	a man standing in front of a television	1.0
3463	000000509095	a giraffe standing next to a wooden fence	1.0
935	000000496307	a person is eating a slice of pizza	1.0
3418	000000508878	a little boy standing in front of a refrigerator	1.0
2598	000000504598	a laptop computer sitting on top of a desk	1.0

Bleu Scores - after 22 epochs

STAATLICH ANERKANNTE HOCHSCHULE

- Bleu Scores after 22 epochs:
  - > only Greedy Search used
  - > Test dataset = 5k images
- 52% data points scores > 0.6
- Maximum concentration in 0.5 to 0.8 score ranges
- ***Used the model in thesis work***

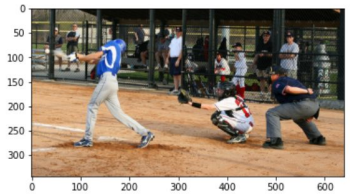
Attention Model after 22 epochs	
Bleu Scores	**Greedy**
Max	1.0
Min	0.0
Median	0.6110
Average	0.6000
Std. Dev.	0.1800

Bleu Scores by Bins – frequency comparison – 5k data points			
Greedy Search			
	Attention Model – 22 epochs		
Score Bin	**Count**	**Rel. Freq**	**Cum. Freq**
0.0 – 0.1	35	0.01	0.01
0.1 – 0.2	94	0.02	0.03
0.2 – 0.3	154	0.03	0.06
0.3 – 0.4	381	0.08	0.14
0.4 – 0.5	724	0.14	0.28
0.5 – 0.6	997	0.20	0.48
0.6 – 0.7	1099	0.22	0.70
0.7 – 0.8	867	0.17	0.87
0.8 – 0.9	468	0.09	0.96
0.9 – 1.0	167	0.03	1.00
Total	4986	-	-

Greedy search inference examples

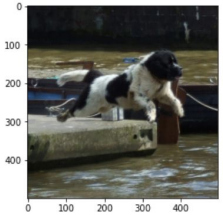
STAATLICH ANERKANNTE HOCHSCHULE

```
------------------------------
Inferrring image 10 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_train2017_1

Ground Truth descriptions:
Orig Desc 1 :: a man swinging a baseball bat on a field.
Orig Desc 2 :: a catcher gets ready to catch a ball in a baseball game.
Orig Desc 3 :: a player is swinging the bat at a ball in a baseball game.
Orig Desc 4 :: the baseball player is swinging his bat as the player gets ready to catch.
Orig Desc 5 :: a couple of baseball players are out on the field
```



```
Greedy caption :: a baseball player swinging a bat at a ball
```

```
------------------------------
Inferrring image 12 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerato

Ground Truth descriptions:
Orig Desc 1 :: a dog jumps into a dirty pond of water.
Orig Desc 2 :: a dog takes the plunge so that he can cool off.
Orig Desc 3 :: a big black and white dog jumping into the water.
Orig Desc 4 :: a black and white dog preparing to jump in water.
Orig Desc 5 :: a dog leaping into some muddy looking water.
```



```
Greedy caption :: a dog on a surfboard in the water
```

```
------------------------------
Inferrring image 14 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_trai

Ground Truth descriptions:
Orig Desc 1 :: two mounted police on horses, one of them smiling
Orig Desc 2 :: two park policemen on top of horses in a park.
Orig Desc 3 :: some police officers on horses in the road.
Orig Desc 4 :: park police sit side by side on their horses with a person nearby.
Orig Desc 5 :: two policemen are riding on horses which are standing still.
```

```
Greedy caption :: two men riding horses on a street
```

```
------------------------------
Inferrring image 30 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_train2017_1

Ground Truth descriptions:
Orig Desc 1 :: a woman with a yellow frisbee in her hands.
Orig Desc 2 :: a heavy set woman in a blue shirt holds a yellow frisbee in a grassy area.
Orig Desc 3 :: an elderly lady holding a yellow frisbee in a field.
Orig Desc 4 :: a woman tossing a frisbee in the park.
Orig Desc 5 :: a lady holding a yellow frisbee and smiling about it.
```

```
Greedy caption :: a woman holding a frisbee in a park
```

From random outputs for 100 images of Test dataset:

nbViewwer link:
https://nbviewer.jupyter.org/github/rbewoor/thesis/blob/master/ImgCap_ATTEND/demo_ImgCap_Attend_Reload_Lappy_Infer_Greedy_1.ipynb

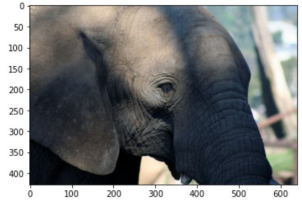
Greedy v/s Beam (width=3) inference examples

STAATLICH ANERKANNTE HOCHSCHULE

```
------------------------------
Inferrring image 17 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_trai

Ground Truth descriptions:
Orig Desc 1 :: a picture of a suitcase on top of a bed.
Orig Desc 2 :: a close up of a dusty old luggage bag
Orig Desc 3 :: a small old suitcase covered in various stickers
Orig Desc 4 :: a dusty vintage suitcase covered with travel stickers.
Orig Desc 5 :: a luggage case covered in stickers is shown.
```

```
Greedy caption :: a man holding a blue and white sheet of stickers on the ground
```

```
------------------------------
Inferrring image 13 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data

Ground Truth descriptions:
Orig Desc 1 :: closeup of a an elephants face with trees in background.
Orig Desc 2 :: this is the close up view of an elephant 's head.
Orig Desc 3 :: a elephant that is standing in the grass.
Orig Desc 4 :: a baby elephant standing next to a tree and grass.
Orig Desc 5 :: a close up profile of an elephant with short tusks
```

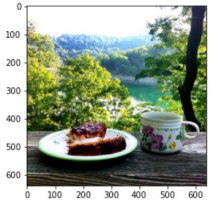


```
Greedy caption :: an elephant standing in a field
```

```
------------------------------
Inferrring image 18 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_t

Ground Truth descriptions:
Orig Desc 1 :: the top of a bus with clock tower in the distance.
Orig Desc 2 :: an image of a tour bus with passengers on it
Orig Desc 3 :: the view of a city street with a clock tower in the back ground.
Orig Desc 4 :: a bus and cars in the street near a giant clock on a big tower.
Orig Desc 5 :: a group of people sit at the top of a bus on the road.
```

```
Greedy caption :: a man standing in front of a large bridge
```

```
------------------------------
Inferrring image 20 of 100:
/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_train2017_118k/000000

Ground Truth descriptions:
Orig Desc 1 :: on this picnic table in front of a lake there is plate with toast and jam and a mug.
Orig Desc 2 :: a wooden table with a plate of food and coffee cup.
Orig Desc 3 :: a plate of dessert is next to a coffee mug.
Orig Desc 4 :: this meal is on a table near a tree.
Orig Desc 5 :: a cake on a plate on a wooden table next to a cup
```



```
Greedy caption :: a table with a plate of food
```

From random outputs for 100 images of Test dataset:

nbViewwer link:
https://nbviewer.jupyter.org/github/rbewoor/thesis/blob/master/ImgCap_ATTEND/demo_ImgCap_Attend_Reload_Lappy_Infer_Greedy_1.ipynb

STAATLICH
ANERKANNTE
HOCHSCHULE

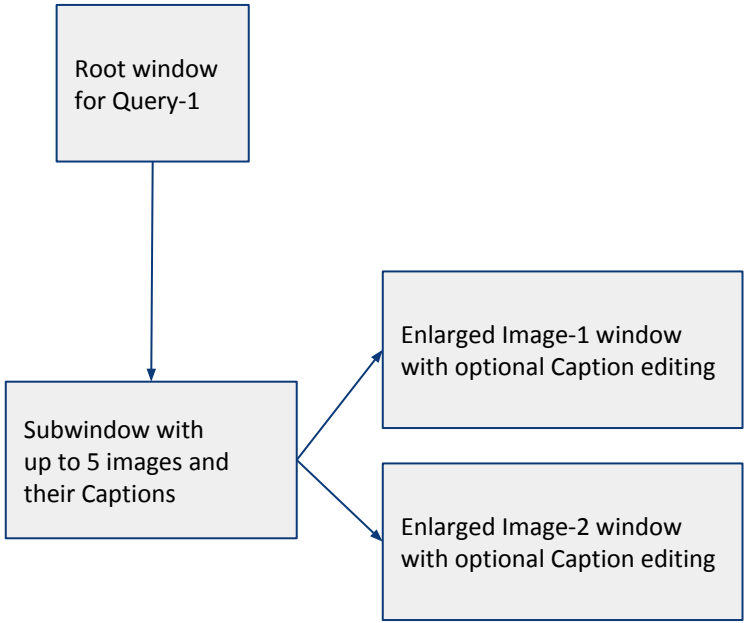
Graphical User Interface

GUI Flow - Approach

STAATLICH ANERKANNTE HOCHSCHULE

- **Goal**: Use the data from the Image Caption inference stage; present the images with their captions to the user for inspection.
  - > Allow user to deselect images
    - > these images and their caption will NOT be sent to the Story Generator block.
  - > Allow editing of the captions for corrections before passing data to Story Generation block
  - > Editing of either or both of the captions for No-attention and With-attention is possible

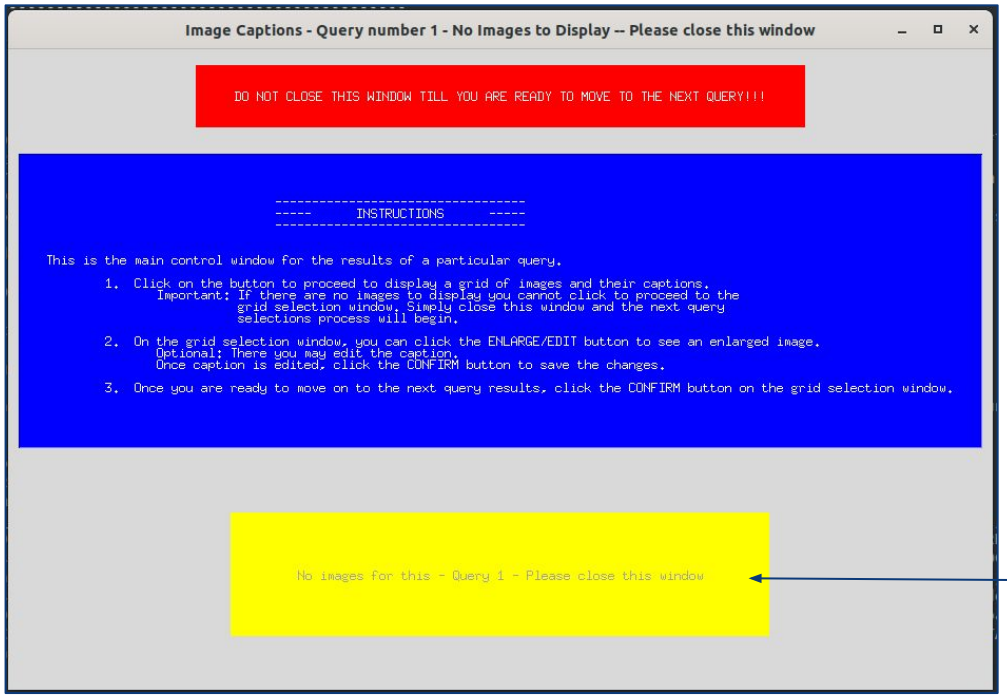
- Repeats for each of the three queries - once per input sentence
  - > If no images, does not allow grid window selection logic



Query with NO images passed

STAATLICH ANERKANNTE HOCHSCHULE

- Main window - no images for this query
  - > User must simply close window and proceed to next query

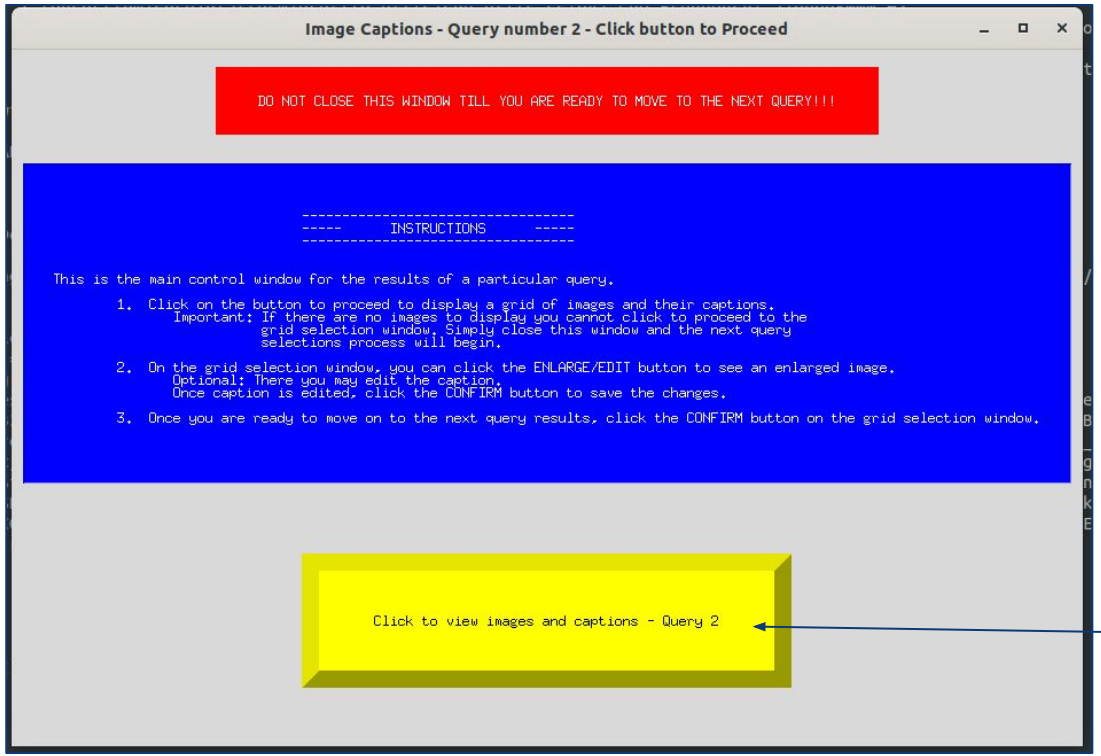


Disabled button

Query with images passed

STAATLICH
ANERKANNTE
HOCHSCHULE

- Main window - query with hits for images



Clickable button - to go to next subwindow

What the user sees !

STAATLICH ANERKANNTE HOCHSCHULE

- Displaying images with original captions - *4 of 5 maximum possible images selected and displayed for this query*

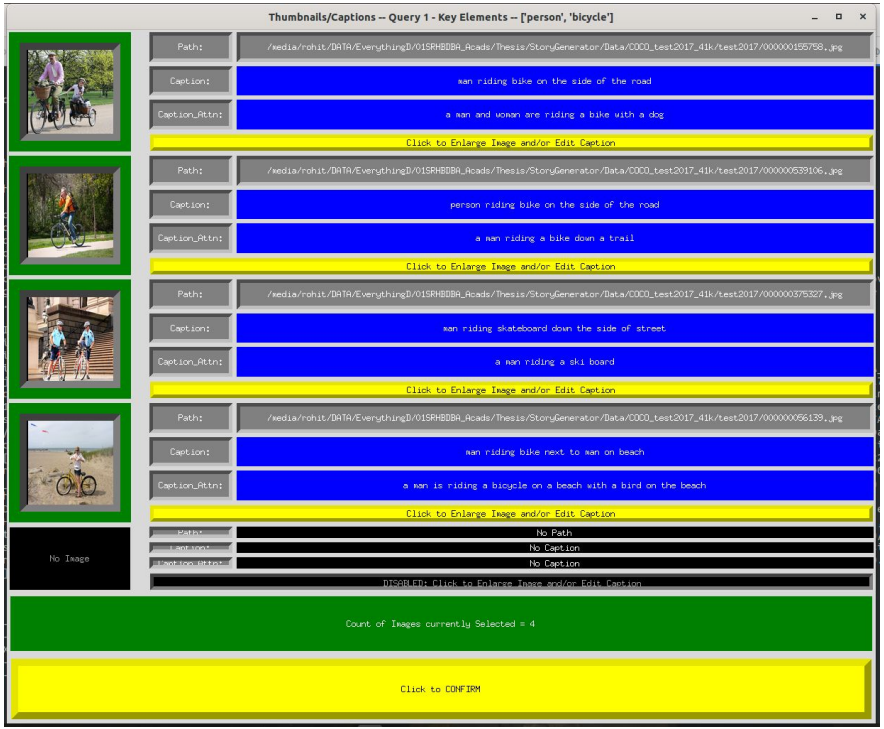


Image thumbnails on far left: Clickable to deselect

Image absolute path

Original Captions for Image (blue boxes)

Clickable button to Enlarge Image + Edit Caption (yellow box)

Fifth image placeholder black as no Image selected

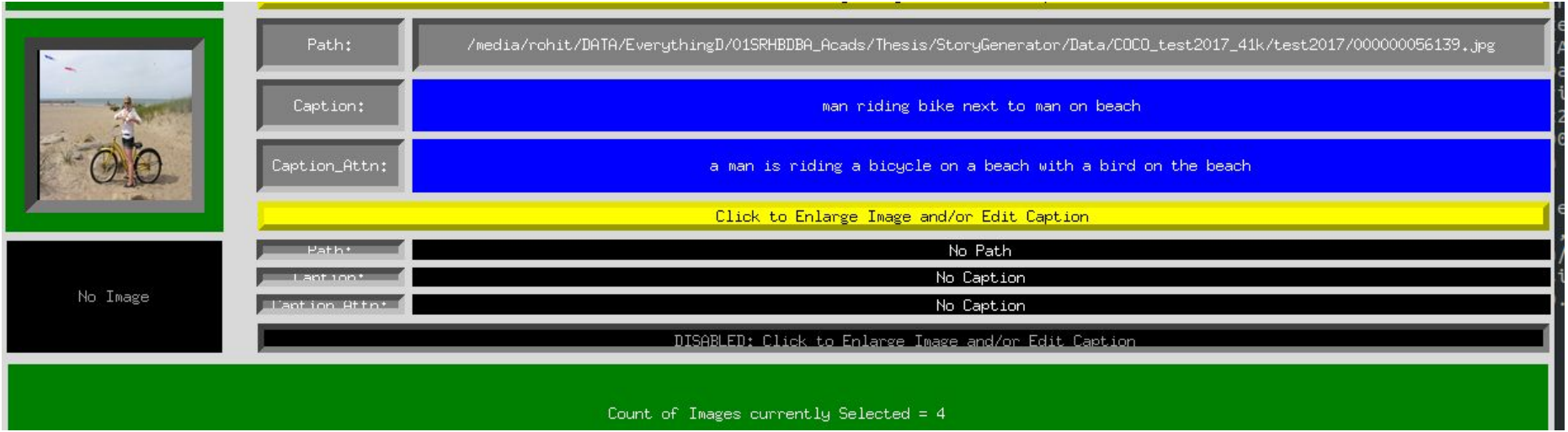
Count of currently selected Images (defaults to ALL)

Final Confirm button

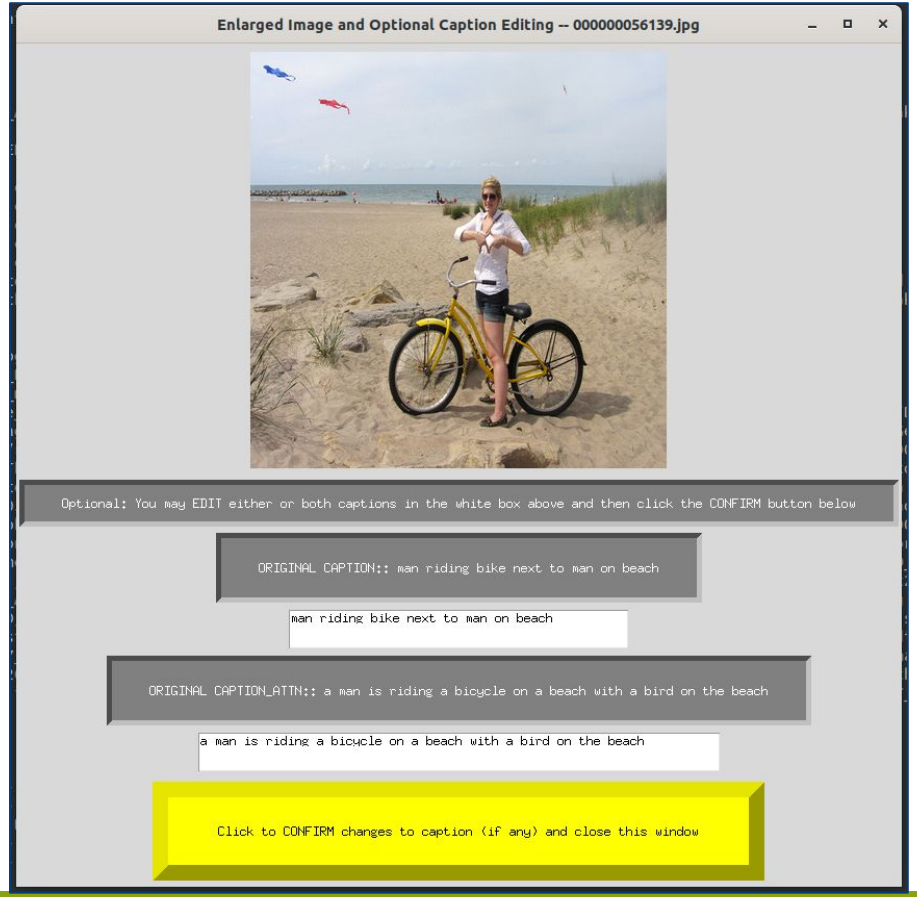
What the user sees !

STAATLICH
ANERKANNTE
HOCHSCHULE

- For readability - part of the previous screenshot



User clicked Enlarge and Edit button



New window shows:

Enlarged image +

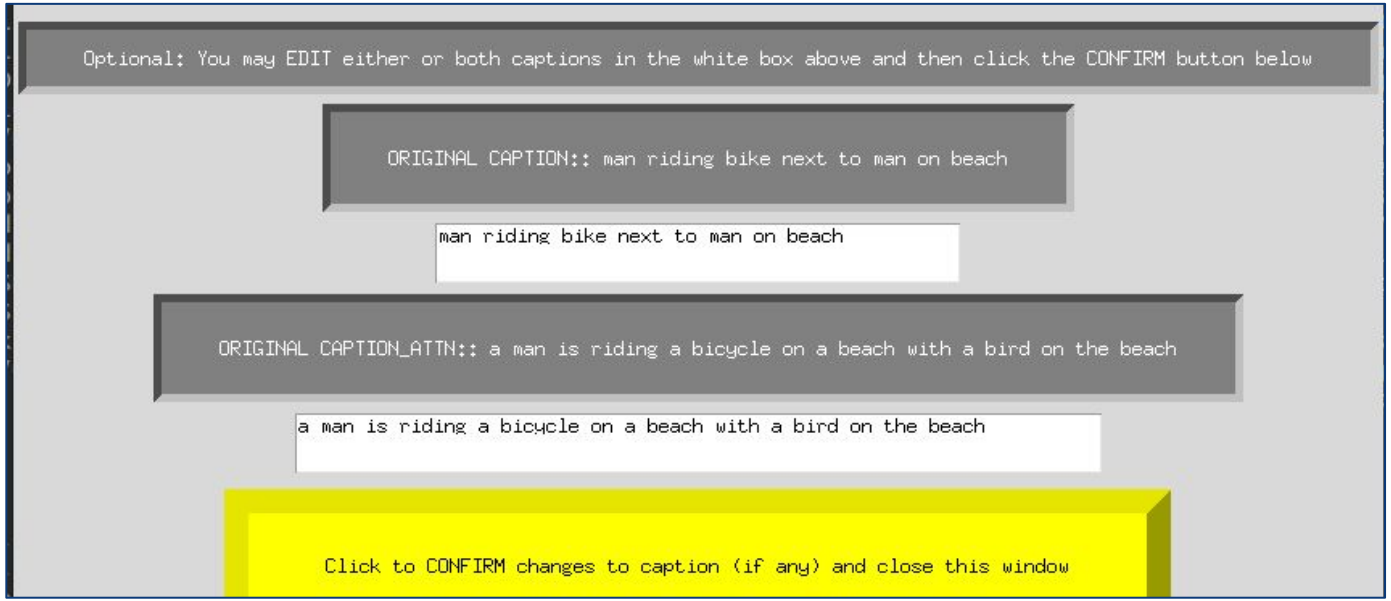
Original captions +

Editable captions (for optional correction)

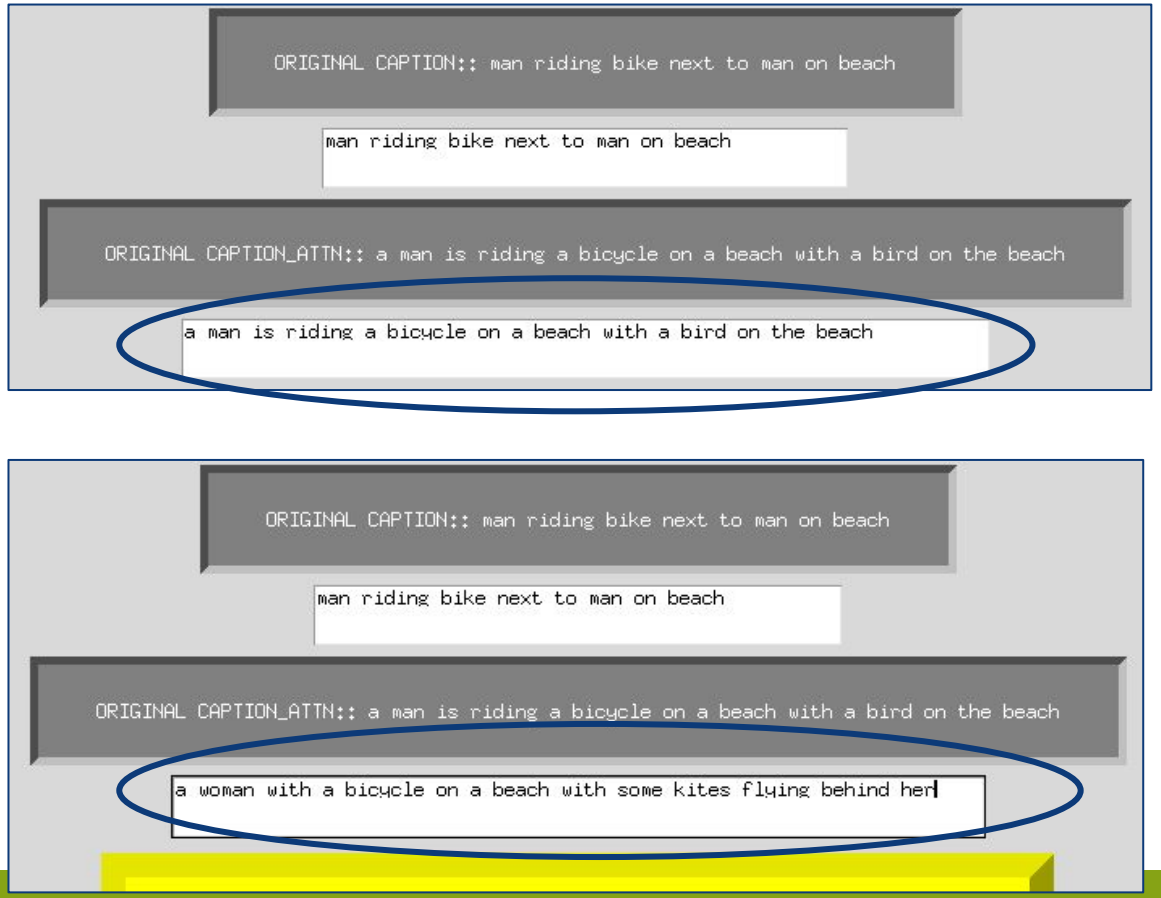
User clicked Enlarge and Edit button

STAATLICH ANERKANNTE HOCHSCHULE

- For readability: Enlarged image + original captions + editable captions (for optional correction)



User edits Attention caption only

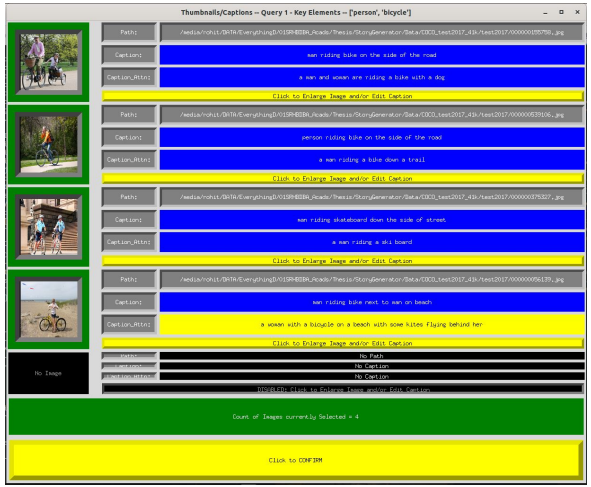


User edited only the caption in the second white box (With-Attention caption)

Caption changes

Then clicks yellow button to CONFIRM

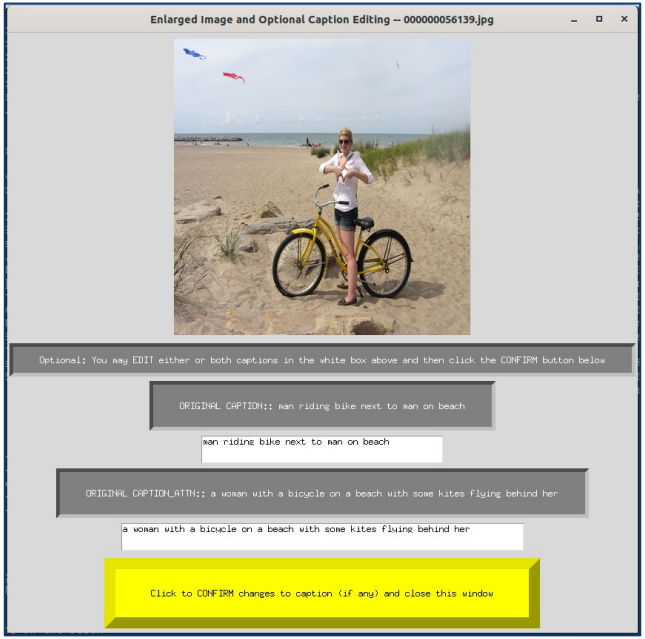
After returning from Enlarge window



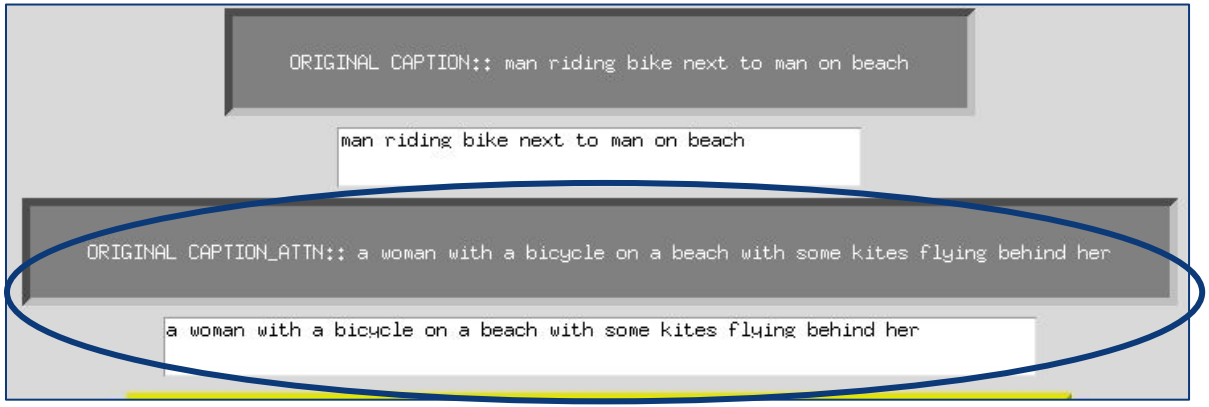
- Old caption replaced with new text
  - > background color change - Blue to Yellow
- Remaining captions unchanged with Blue backgrounds

Edited caption shown on re-entering

STAATLICH ANERKANNTE HOCHSCHULE

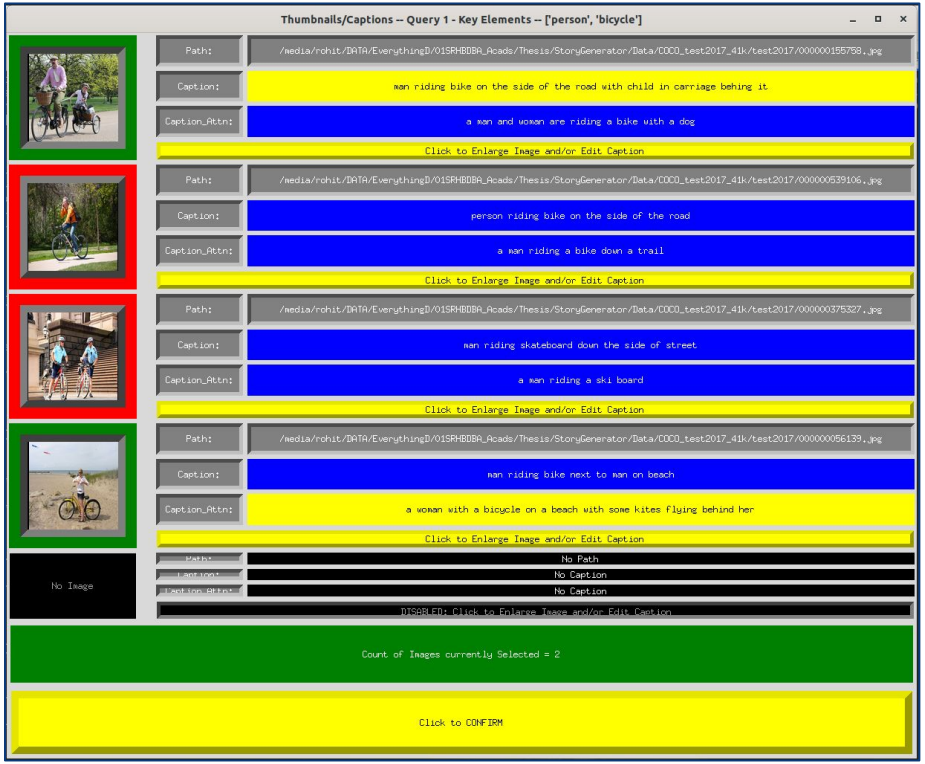


- On re-entering window: Original Caption section now shows just edited caption
- If desired, user may change the caption again like before

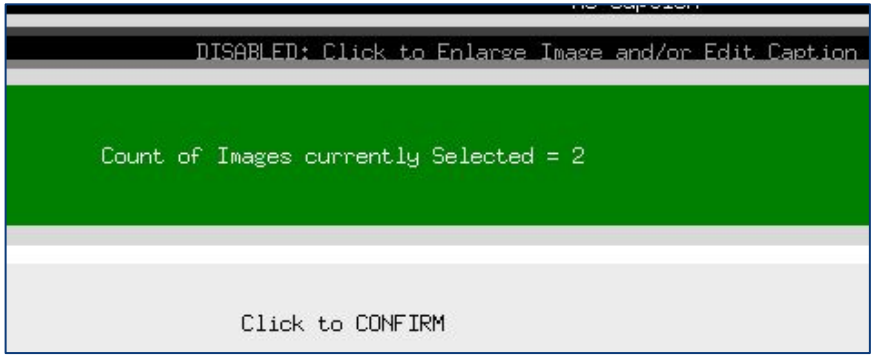


Ready for final Confirmation for this Query

STAATLICH ANERKANNTE HOCHSCHULE



- For this query:
  - > User edited one caption (first and last images)
    - > only two caption boxes are yellow
  - > User Deselected middle two images
    - > note red border around thumbnail



Data structure changes in backend

STAATLICH ANERKANNTE HOCHSCHULE

- Query 1: Deselected 2 of 4 images, Edited one caption only in each of the two selected images
- Query 2: Retained both images, Edited both captions for both images
- Query 3: Deselected all 3 images

```
************************************************************
******** SUMMARY for Image Captioning GUI - all queries *********
*********     AFTER removing the deselected images      *********
************************************************************
Data structure BEFORE =
[{'key_elements': ['person', 'bicycle'], 'selected_images_no_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'man riding bike on t
he side of the road'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000539106.jpg', 'person riding bike on the side of the road'], ['/media/rohit/DATA/
EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000375327.jpg', 'man riding skateboard down the side of street'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/Story
Generator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'man riding bike next to man on beach']], 'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COC
O_test2017_41k/test2017/000000155758.jpg', 'a man and woman are riding a bike with a dog'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000539106.jpg'
, 'a man riding a bike down a trail'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000375327.jpg', 'a man riding a ski board'], ['/media/rohit/DATA/Ev
erythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'a man is riding a bicycle on a beach with a bird on the beach']]}, {'key_elements': ['person', 'tvmonitor'], 's
elected_images_no_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'man standing in living room holding nintendo wii game controlle
r'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'man and woman are playing video game']], 'selected_images_with_attn': [['/media/rohi
t/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'a man playing a game with a remote controller'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesi
s/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'two men playing wii in a living room']]}, {'key_elements': ['handbag'], 'selected_images_no_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_
Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000117100.jpg', 'two suitcases sitting next to each other on the side of the road'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGen
erator/Data/COCO_test2017_41k/test2017/000000093925.jpg', 'man in suit and tie standing in front of building'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test20
17/000000313060.jpg', 'woman is standing in front of her cell phone']], 'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/0000001
17100.jpg', 'a little girl sitting on a black suitcase'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000093925.jpg', 'a man in a suit and tie'], ['/m
edia/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000313060.jpg', 'a man is sitting on a cell phone while looking at a cell phone']]}]
Deselected image positions all queries = [[1, 2], [], [0, 1, 2]]
Data structure AFTER GUI; changed captions and removing Deselected images =
[{'key_elements': ['person', 'bicycle'], 'selected_images_no_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'man riding bike on t
he side of the road with child in carriage behing it'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'man riding bike next to man on be
ach']], 'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'a man and woman are riding a bike with a dog'], ['/
media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'a woman with a bicycle on a beach with some kites flying behind her']]}, {'key_elements': ['
person', 'tvmonitor'], 'selected_images_no_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'QUERY 2 - IMAGE 1 - EDITED STANDARD CA
PTION'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'man and woman are playing video game']], 'selected_images_with_attn': [['/media/
rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'QUERY 2 - IMAGE 1 - EDITED ATTENTION CAPTION'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Th
esis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'QUERY 2 - IMAGE 2 - EDITED ONLY ATTENTION CAPTION']]}, {'key_elements': ['handbag'], 'selected_images_no_attn': [], 'selected_images_with_a
ttn': []}]
```

Data structures before and after this stage - Console output

Data structure changes in backend

STAATLICH ANERKANNTE HOCHSCHULE

```
*****************************************************************
********* SUMMARY for Image Captioning GUI - all queries *********
*********     AFTER removing the deselected images      *********
*****************************************************************
Data structure BEFORE =
[{'key_elements': ['person', 'bicycle'], 'selected_images_no_attn':
[['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'man riding bike on the side of the road'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000539106.jpg', 'person riding bike on the side of the road'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000375327.jpg', 'man riding skateboard down the side of street'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'man riding bike next to man on beach']],
'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'a man and woman are riding a
bike with a dog'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000539106.jpg', 'a man riding a bike down a trail'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000375327.jpg', 'a man riding a ski board'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'a man is riding a bicycle on a beach with a bird on the
beach']]}, {'key_elements': ['person', 'tvmonitor'], 'selected_images_no_attn':
[['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'man standing in living room holding nintendo wii game
controller'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'man and woman are playing video game']],
'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'a man playing a game with a
remote controller'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'two men playing wii in a living room']]},
{'key_elements': ['handbag'], 'selected_images_no_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000117100.jpg', 'two
suitcases sitting next to each other on the side of the road'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000093925.jpg',
'man in suit and tie standing in front of building'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000313060.jpg', 'woman is
standing in front of her cell phone']], 'selected_images_with_attn':
[['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000117100.jpg', 'a little girl sitting on a black suitcase'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000093925.jpg', 'a man in a suit and tie'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000313060.jpg', 'a man is sitting on a cell phone while looking at a cell
phone']]}]
```

Data structures before and after this stage - text extracted for readability

Data structure changes in backend

STAATLICH ANERKANNTE HOCHSCHULE

****
********* SUMMARY for Image Captioning GUI - all queries *********
********* AFTER removing the deselected images *********
****
Data structure BEFORE =
[{'key_elements': ['person', 'bicycle'], 'selected_images_no_attn':
….
….
….
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000313060.jpg', 'a man is sitting on a cell phone while looking at a cell phone']]}]
Deselected image positions all queries = [[1, 2], [], [0, 1, 2]]
Data structure AFTER GUI; changed captions and removing Deselected images =
[{'key_elements': ['person', 'bicycle'], 'selected_images_no_attn':
[['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'man riding bike on the side of the road with child in carriage behing it'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'man riding bike next to man on beach']], 'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000155758.jpg', 'a man and woman are riding a bike with a dog'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000056139.jpg', 'a woman with a bicycle on a beach with some kites flying behind her']]}, {'key_elements': ['person', 'tvmonitor'], 'selected_images_no_attn':
[['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'QUERY 2 - IMAGE 1 - EDITED STANDARD CAPTION'], ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'man and woman are playing video game']], 'selected_images_with_attn': [['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000382917.jpg', 'QUERY 2 - IMAGE 1 - EDITED ATTENTION CAPTION'],
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'QUERY 2 - IMAGE 2 - EDITED ONLY ATTENTION CAPTION']]}, {'key_elements': ['handbag'], 'selected_images_no_attn': [], 'selected_images_with_attn': []}]

Data structures before and after this stage - text extracted for readability

Data structure changes in backend

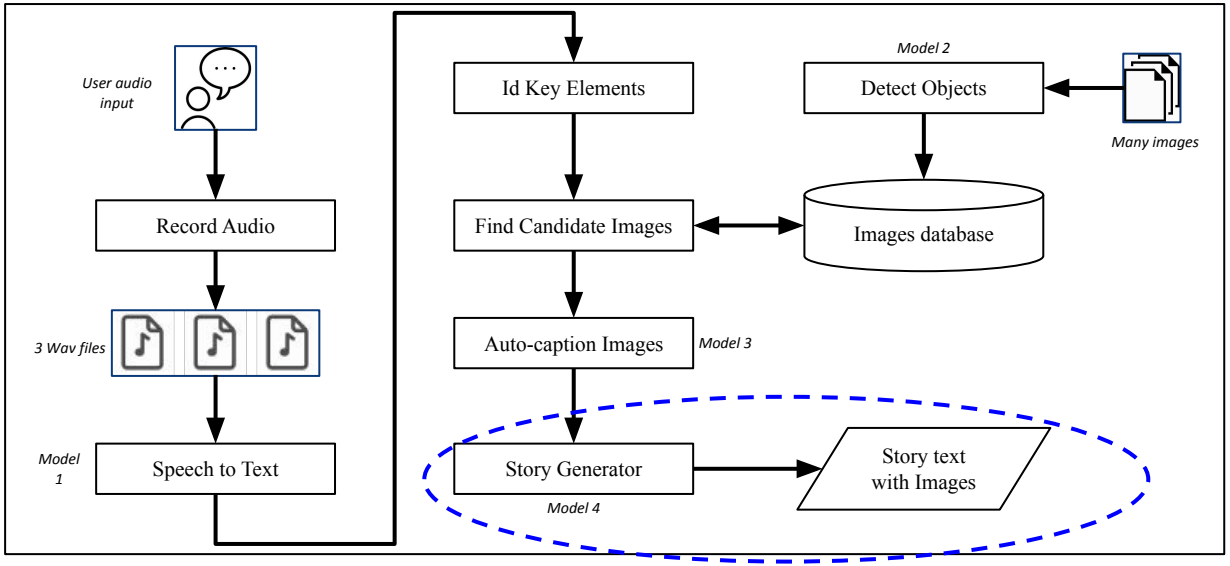
STAATLICH ANERKANNTE HOCHSCHULE

- Updated data structure saved to file for processing by Story Generator block later

```
****************************************************************
********* SUMMARY for Image Captioning GUI - all queries *********
*********      AFTER removing the deselected images      *********
****************************************************************
Data structure BEFORE =
[{'key_elements': ['person', 'bicycle'], 'selected_images_no_attn':
….
….
….
['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000313060.jpg', 'a man is sitting on a cell phone while looking at a cell phone']]}]
….
….
….
 ['/media/rohit/DATA/EverythingD/01SRHBDBA_Acads/Thesis/StoryGenerator/Data/COCO_test2017_41k/test2017/000000438521.jpg', 'QUERY 2 - IMAGE 2 - EDITED ONLY ATTENTION CAPTION']]}, {'key_elements': ['handbag'], 'selected_images_no_attn': [], 'selected_images_with_attn': []}]

LOG_LEVEL INFO ::
Saved results to file for the Story Generator stage.
File :: /home/rohit/PyWDUbuntu/thesis/combined_execution/ImgCapAfterGui/op_img_cap_after_gui.txt
```

Data structures before and after this stage - text extracted for readability

Stage 5:
Story Generator



Goal and Introduction

STAATLICH
ANERKANNTE
HOCHSCHULE

- **Goal**: Output natural language text as a story based on seed values provided from earlier stages in pipeline
- Field called “Natural Language Generation”. Two sub-fields:
  - > *text-to-text generation*: applications that take existing texts as their input, and automatically produce a new, coherent text as output
  - > *data-to-text generation*: applications that automatically generate text from non-linguistic data
- Thesis work maps to text-to-text generation use-case
- Language Model: “Machine learning model able to look at part or the full text of a sentence and predict the next word”
  - > Character level models also possible but very resource intensive to train and not popular currently
- Selected Generative Pre-Training version 2 (GPT-2) as model to use

About GTP-2

STAATLICH
ANERKANNTE
HOCHSCHULE

- Released by OpenAI project in 2019
  - > Link to project: https://openai.com/blog/better-language-models/
  - > GPT-2 is a direct scale-up of GPT, with more than 10X the parameters and trained on more than 10X the amount of data
  - > Subsequently much larger GPT-3 also released but special access via paid API's required
- Transformer based language model
- Trained on 8 million web pages (approx. 40GB of data)
- 1.5 billion parameters in model
- GPT-2 outperforms other language models trained on specific domains (like Wikipedia, news, or books) without needing to use these domain-specific training datasets
- On language tasks like question answering, reading comprehension, summarization, and translation, GPT-2 begins to learn these tasks from the raw text, using no task-specific training data

Fine tuning GTP-2 for use case

STAATLICH ANERKANNTE HOCHSCHULE

- Cloned code from this Github repo: https://github.com/minimaxir/gpt-2-simple
  - > Allows fine-tuning on use-case text
  - > Options to download different pre-trained models of varying sizes and increasing requirement of compute resources to fine-tune
    - > "Small" :: 124 million parameters :: 0.5 GB
    - > "Medium" :: 355 million parameters :: 1.5 GB
    - > "Large" :: 774 million parameters :: unknown size
    - > "Extra large" :: 1158 million parameters :: unknown size
    - > "Large" and "Extra large" :: too huge to even consider, as cannot even train on Google colab or Kaggle clouds
  - > **Used Medium model in thesis work**
- Data used in fine-tune training run:
  - > The Children's Book Test, The bAbI project by Facebook. https://research.fb.com/downloads/babi/
  - > 11 files from the CBT dataset :: approx. 100 MB of data

Inference Logic

STAATLICH ANERKANNTE HOCHSCHULE

- Treated the captions generated by auto-caption models as the seed values

- Treating the No-Attention and With-Attention captions as separate sets of possible inputs:
  - > Combined the captions in all possible ways to generate each possible seed
  - > Combined seed used to generate the story using GPT-2

- Parameters that can be set while performing inference:
  - > “Temperature” parameter: value from 0.0 to 1.0
    - > Lower value results in less random completions. With value set to 0.0, the model is deterministic and repetitive.
    - > Higher value results in more randomness of output - which is desired in this use case.
  - > “Length”: number of tokens in the results. Set to 300 in use case
  - > “Top_k”: Integer value controlling diversity
    - > Value = 1 => only 1 word is considered for each step (token), resulting in deterministic completions
    - > Value = 0 => Default value used. No restrictions
    - > Value = 40 => 40 words are considered at each step
    - > Recommended to use value 40 as a starting point
  - > “Nsamples”: Integer value specifying how many samples to output per input seed

Examples of Inference Results

STAATLICH ANERKANNTE HOCHSCHULE

- Stage 1) The Speech-to-Text output from the 3 wav files:
  - > ['a person ride a bicycle in the park', 'a person watches the news on the tvmonitor', 'the handbag has many items in it']

- Stage 2) Key elements identified using NLP POS-tagging:
  - > [['person', 'bicycle'], ['tvmonitor', 'person'], ['handbag']]
  - > each inner list corresponds to one spoken sentence transcription

- Note: Parts of the data structures are over-written with ***ignore*** - these were just the paths of the images

- "selected_images_no_attn" and "selected_images_with_attn": caption generator outputs for without and with attention respectively

- Stage 3) Auto-caption model outputs:
  - > [{'key_elements': ['person', 'bicycle'], ***'selected_images_no_attn'***: [[''***ignore***'', 'man riding bike on the side of the road'], [''***ignore***'', 'person riding bike on the side of the road']], 'selected_images_with_attn': [[''***ignore***'', 'a man and woman are riding a bike with a dog'], ['***ignore***'', 'a man on a bike down a bike']]}, {'key_elements': ['tvmonitor', 'person'], ***'selected_images_no_attn'***: [[''***ignore***'', 'an image of tv that is sitting on bed'], [''***ignore***'', 'desk with computer monitor and laptop on it']], 'selected_images_with_attn': [[''***ignore***'', 'a cat laying on top of a bed'], [''***ignore***'', 'a microwave and a microwave and a microwave']]}, {'key_elements': ['handbag'], 'selected_images_no_attn': [[''***ignore***'', 'woman sitting on bench talking on her cell phone']], 'selected_images_with_attn': [[''***ignore***'', 'two women sitting on a bench with a bag']]}]

Examples of Inference Results

STAATLICH
ANERKANNTE
HOCHSCHULE

- Note: Edited some of the captions in the Gui

- The auto-caption outputs after corrections using GUI:
  - > [['A man and child are riding a bike in a park with some dogs being walked far away.', 'A person laying on top of a bed watching tv.', 'Two women sitting on a bench with a bag.'], ['A man and child are riding a bike in a park with some dogs being walked far away.', 'A microwave and a microwave and a microwave.', 'Two women sitting on a bench with a bag.'], ['A wonman on a bike in the park.', 'A person laying on top of a bed watching tv.', 'Two women sitting on a bench with a bag.'], ['A wonman on a bike in the park.', 'A microwave and a microwave and a microwave.', 'Two women sitting on a bench with a bag.']]
  - > Each of the inner lists in data structure above is the seed considered for generating a story.

Jupyter Notebook of many Inferences

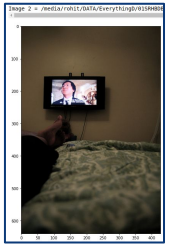
STAATLICH
ANERKANNTE
HOCHSCHULE

- Please see this link of jupyter notebook code uploaded to Github:
  https://nbviewer.jupyter.org/github/rbewoor/thesis/blob/master/StatusUpdate/StyGen_Show_Results_2_staus_update_run_2.ipynb
  - > Story generated with parameters:
    - > Length = 300
    - > Temperature = 0.95
    - > Number of Samples = 2

- Please see this link of jupyter notebook code uploaded to Github:
  https://nbviewer.jupyter.org/github/rbewoor/thesis/blob/master/StatusUpdate/StyGen_Show_Results_2_staus_update_run_1.ipynb
  - > Story generated with parameters:
    - > Length = 300
    - > Temperature = 0.50
    - > Number of Samples = 2

Example - High Temperature - No Attention

- Generator settings: Length = 300, Temperature = 0.95, Top_k = 0
- Caption: No attention
- Story: ********** Story 1 of 4 **********

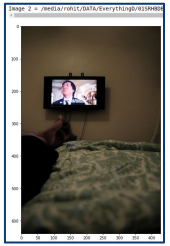


Input Seed:
Man riding bike in a park with a child. An image of tv that is sitting on bed. Woman sitting on bench talking on her cell phone.

Output Story:
Man riding bike in a park with a child. An image of tv that is sitting on bed. Woman sitting on bench talking on her cell phone. Who now ? '' Peter looked between them sheepishly, as he heard the next cry from under the table. `` I guess she wants the first person who speaks to her, '' answered the doctor, holding Peter and looking up at him. Peter awoke -LRB- rolled up his hair -RRB-, but his eyes were not in any humor as he looked still. `` What ? '' answered Peter, looking at the doctor as if for some answer. He understood no reply and tried to look sober. `` Pretty much what you say. Just what I have advised you to do. She will never grow right out of Algren 's sad life now, and she is sick out of it now, I know. I dare say that it is well He is a comfort for her. She will let him ride her, he will be lesser . '' `` Do not know anything about horses, '' answered Peter with a sigh. `` Hiding will do equally well. She will fall off . '' His own patronization of dogs went out like a wave of misfortune with Duck 's heart pounding under his warm great shoulders. `` Well, matter of fact, '' said old St. John, `` she will never come back. Her paddock will be dark. She reckons she will resent the dog-thing. I feel the need of asking you to go along for a bit of phon

Example - High Temperature - With Attention

- Generator settings: Length = 300, Temperature = 0.95, Top_k = 0
- Caption: With attention
- Story: ********** Story 1 of 4 **********



Input Seed:
A man and child are riding a bike in a park with some dogs being walked far away. A person laying on top of a bed watching tv. Two women sitting on a bench with a bag.

Output Story:
A man and child are riding a bike in a park with some dogs being walked far away. A person laying on top of a bed watching tv. Two women sitting on a bench with a bag. No.... I have not made up my mind yet to defend myself and Jane Fairfax must be sad. I wish the boys had been there just for the sake of fancy. Nobody will ever mistake my name for Jane Fairfax -- as she used to be . '' `` And what did you have to say about Jane Fairfax ? '' Jane Fairfax smiled wildly. `` I did not ask about Jane. I fancied you were a cow . '' The dolls were so reminded of Jane Fairfax, who they felt their wits again, for Jane Fairfax was a dreadful name, but no one wanted to believe she belonged to Jane Fairfax. The girls looked in confusion and disappointment. To be sure she was not quite as bold as Robin Thicke, and perhaps she was a cleaner, for sure it was not Robin Thicke. But the idea of a street clerk like Jane Fairfax -- a woman in her case, and not much about Jane -- making cheerful speeches for strangers! Jane hired many messengers for her fairies in the city, but none of them talked his name. `` And who the hell do you mean to call so ? '' cried Dan out in a puzzled tone. `` Jane Fairfax, '' replied Jane at last. `` Jane Fairfax ! '' said Dan. `` Jane Fairfax ! '' Jane Fairfax 's eyes had gone wild for the time ; she seemed to sink into silence and a pleasant thought

Example - Low Temperature - No Attention

STAATLICH
ANERKANNTE
HOCHSCHULE

- Generator settings: Length = 300, Temperature = 0.50, Top_k = 0
- Caption: No attention
- Story: ************* Story 1 of 4 *************

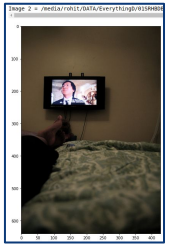
Input Seed:
Man riding bike in a park with a child. An image of tv that is sitting on bed. Woman sitting on bench talking on her cell phone.

Output Story:
Man riding bike in a park with a child. An image of tv that is sitting on bed. Woman sitting on bench talking on her cell phone. I do not want to see that. I am not a man. I will get out, and we will talk together. I will tell you all I want to know about the war. I can tell you all I want to know about the war . '' `` My dear, it is all a lie, '' said the girl. `` I do not want to see the war. I want to know what it is. I am not a man. I am a woman. I have been in the war. I have been in it all the time. I am a woman. I

Example - Low Temperature - With Attention

STAATLICH ANERKANNTE HOCHSCHULE

- Generator settings: Length = 300, Temperature = 0.50, Top_k = 0
- Caption: With attention
- Story: ********** Story 1 of 4 **********



Input Seed:
A man and child are riding a bike in a park with some dogs being walked far away. A person laying on top of a bed watching tv. Two women sitting on a bench with a bag.

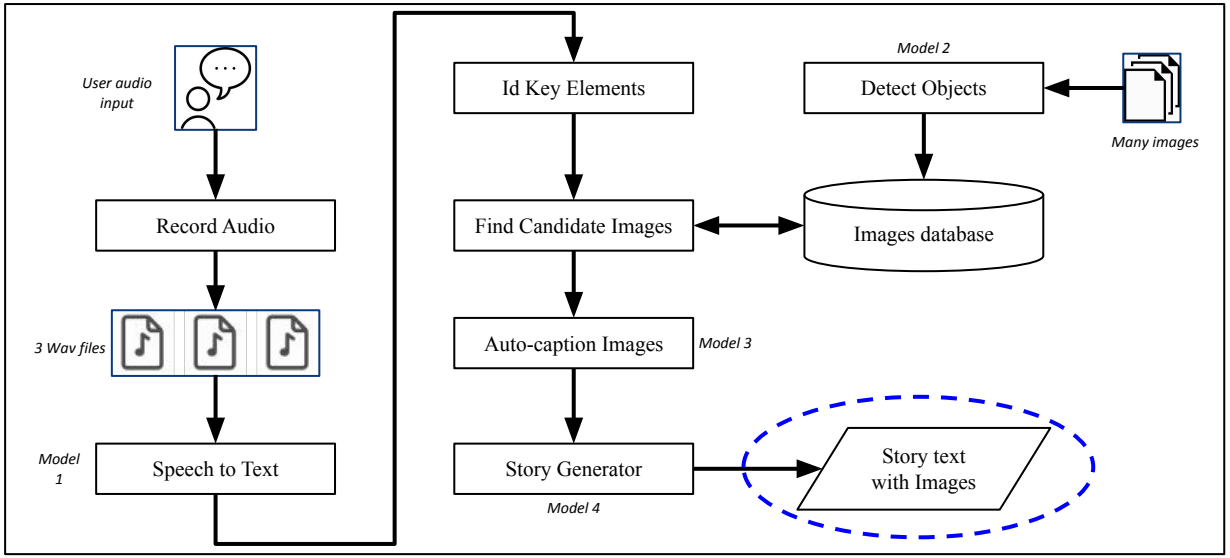
Output Story:
A man and child are riding a bike in a park with some dogs being walked far away. A person laying on top of a bed watching tv. Two women sitting on a bench with a bag. A man and child are sitting on a bench with a bag on their shoulders. A man and child are sitting on a bench with a bag on their head. A man and child are sitting on a bench with a bag on their shoulders. A man and child are sitting on a bench with a bag on their head. A man and child are sitting on a bench with a bag on their shoulders. A man and child are sitting on a bench with a bag on their head. A man and child are sitting on a bench with a bag on their shoulders. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with a bag on their shoulders. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with a bag on their shoulders. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with a bag on their heads. A man and child are sitting on a bench with

Quality of Stories

STAATLICH
ANERKANNTE
HOCHSCHULE

- Overall seems to be quite poor
  - > Will attempt using some proper input sentences now and include results in thesis writing
  - > Also, will try to send something from evaluation metrics perspective - pending right now
- Higher temperature allows for more creative outputs and prevents sentence repetition
- Low Temperature = 0.5 made the sentences repeat nonsensically

Evaluation of Stories - PENDING



Changes to Expose

STAATLICH
ANERKANNTE
HOCHSCHULE

1. Limit the names of objects found from images in database:
   a. Only the labels of the object detection model can be “found” in the images.
   b. Implication: User must speak using exactly these labels while structuring the input sentences.
   c. Currently: 80 labels part of the detection database
      i. labels = ['aeroplane', 'apple', 'backpack', 'banana', 'baseball bat', 'baseball glove', \
         'bear', 'bed', 'bench', 'bicycle', 'bird', 'boat', 'book', 'bottle', 'bowl', \
         'broccoli', 'bus', 'cake', 'car', 'carrot', 'cat', 'cell phone', 'chair', \
         'clock', 'cow', 'cup', 'diningtable', 'dog', 'donut', 'elephant', 'fire hydrant', \
         'fork', 'frisbee', 'giraffe', 'hair drier', 'handbag', 'horse', 'hot dog', \
         'keyboard', 'kite', 'knife', 'laptop', 'microwave', 'motorbike', 'mouse', \
         'orange', 'oven', 'parking meter', 'person', 'pizza', 'pottedplant', \
         'refrigerator', 'remote', 'sandwich', 'scissors', 'sheep', 'sink', 'skateboard', 'skis', \
         'snowboard', 'sofa', 'spoon', 'sports ball', 'stop sign', 'suitcase', 'surfboard', \
         'teddy bear', 'tennis racket', 'tie', 'toaster', 'toilet', 'toothbrush', 'traffic light', \
         'train', 'truck', 'tvmonitor', 'umbrella', 'vase', 'wine glass', 'zebra']
   d. Will call out this limitation out in proposal

2. Voice input simulated using wav files to represent each of the three sentences. These need to be recorded separately and then presented to the system.
   a. Implication: User needs to record their input externally (e.g. using Audacity, etc) to create a wav file in required format (16kHz sampling, 16-bit, Mono).
   b. Only if time permits, will try to automate this aspect to directly record users speech and include wav file creation as part of pipeline.

Changes to Expose

STAATLICH
ANERKANNTE
HOCHSCHULE

3. While recording the audio files, the user must speak in “active” and not “passive” voice:
    a. E.g. “person is walking his dog”, NOT “dog is being walked by a person”
        i. Shorter and cleaner sentence for model.
        ii. Generally natural speech is in Active voice, so language models will have more training on such sentences.
        iii. Less chance of transcription errors.
    b. Only if time permits will relax this criterion later on.

4. User will be presented opportunities at various stages in pipeline to edit the data being processed:
    a. Keywords identification - select exactly 1/ 2/ 3 words per input wav file.
    b. Images extracted from database - view them and deselect images if required.
        i. Maximum limit of selection = 5 images.
        ii. Remove an image due to false detection of object

5. Limiting the types of words and number of words processed as keywords for the “Identify key elements” block:
    a. Irrespective of what all was said by user in the input wav file, only the Noun words in the transcription of each wav file will be processed.
    b. Thus increase chance of finding a suitable image with all the objects present.

Changes to Expose

STAATLICH
ANERKANNTE
HOCHSCHULE

6. Only English language is allowed.

7. Images passed to the “Auto caption images” block will be limited to 5 images per input sentence.

8. Motivation: what exactly to add?
   a. Easy way to provide short stories on the fly for children
   b. User (usually parent) controls what the story says. If the child slightly older (and speaks clearly) they can ask for stories that interest them.
   c. No paper, no delivery of the book, unlimited stories

9. Inputs from Prof. Sprick during 23.07 meeting:
   a. Motivation structure above okayed.
   b. Make overall story process even more interactive:
      i. Store sounds from typical keywords. For example:
         > Dog: “dog barking”
         > Truck: “truck honking”, or “truck going past on road”, “truck engine starting”
         > Spoon: “sounds of cutlery being used”
      ii. Once story is ready, use a Text-to-speech (TTS) block to “speak the story”. Superimpose suitable sounds in background when appropriate word is being spoken.
      iii. Accepted idea as it is excellent. But will only be attempted if time permits.

Changes to Expose

STAATLICH
ANERKANNTE
HOCHSCHULE

6. inputs from Mr. Frank Schulz during 31.07 meeting:
    a. Once overall models working together, use the pipeline parameters as part of research question to guage efficacy of the use-case implementation.
    b. For example:
        i. how many images should be sent to auto-caption
        ii. How many words per input audio file should be kept as the candidate key elements

7. Each audio input file (or user input via mic if possible) to be exactly one sentence. Thus model expects exactly three input sentences as the start point user input.

References

STAATLICH
ANERKANNTE
HOCHSCHULE

1. Paper: A. Hannun et al. Deep Speech: Scaling up end-to-end speech recognition. 17.12.2014. https://arxiv.org/abs/1412.5567
2. Website: How to Perform Object Detection With YOLOv3 in Keras. https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/
3. Github Code: https://github.com/rbewoor/keras-yolo3 (forked from https://github.com/jbrownlee/keras-yolo3 on 05.06.2020)
4. Website: YOLOv3 pre-trained weights downloaded from: https://pjreddie.com/media/files/yolov3.weights
5. Paper: J. Redmon et al. You Only Look Once: Unified, Real-Time Object Detection. 09-05-2016. https://arxiv.org/abs/1506.02640
6. Paper: J. Redmon et al. YOLO9000: Better, Faster, Stronger. 25-12-2016. https://arxiv.org/abs/1612.08242
7. Paper: J. Redmon et al. YOLOv3: An Incremental Improvement. 08-04-2018. https://arxiv.org/abs/1804.02767
8. Website: How to Visualize a Deep Learning Neural Network Model in Keras. https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/
9. Website: All About YOLO Object Detection and its 3 versions (Paper Summary and Codes). https://medium.com/data-science-in-your-pocket/all-about-yolo-object-detection-and-its-3-versions-paper-summary-and-codes-2742d24f56e
10. Website: YOLO v3 theory explained, https://medium.com/analytics-vidhya/yolo-v3-theory-explained-33100f6d193 as on 10.06.2020
11. Paper: M. Tanti et al. What is the Role of RNNs in an Image Caption Generator?. 07.08.2017. https://arxiv.org/abs/1708.02043
12. Paper: M. Tanti et al Where to put the Image in an Image Caption Generator. https://arxiv.org/abs/1703.09137

References

STAATLICH
ANERKANNTE
HOCHSCHULE

13. Paper: X. Liu et al. A survey of deep neural network-based image captioning. 09.06.2018. https://doi.org/10.1007/s00371-018-1566-y
14. Website: Image Captioning with Keras, https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8 as on 10.09.2020
15. Github Code: Automatic Image Captioning, https://gist.github.com/nttuan8/a621aa6700995db6db71b0a768a8552f (cloned on 20.09.2020)
16. Paper: O. Vinyals et al. Show and Tell: A Neural Image Caption Generator. 20.04.2015. https://arxiv.org/abs/1411.4555
17. Paper: K. Xu et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 19.04.2016. https://arxiv.org/abs/1502.03044
18. Paper: A. Gatt et al. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. 29-01-2018. https://arxiv.org/abs/1703.09902
19. Paper GPT-1: A. Redford et al. Improving Language Understanding by Generative Pre-Training. 2018. https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035
20. Paper GPT-2: A. Redford et al. Improving Language Understanding by Generative Pre-Training. 2019. https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe
21. Github Code: gpt-2-simple, https://github.com/minimaxir/gpt-2-simple (forked on 22.10.2020)
22.

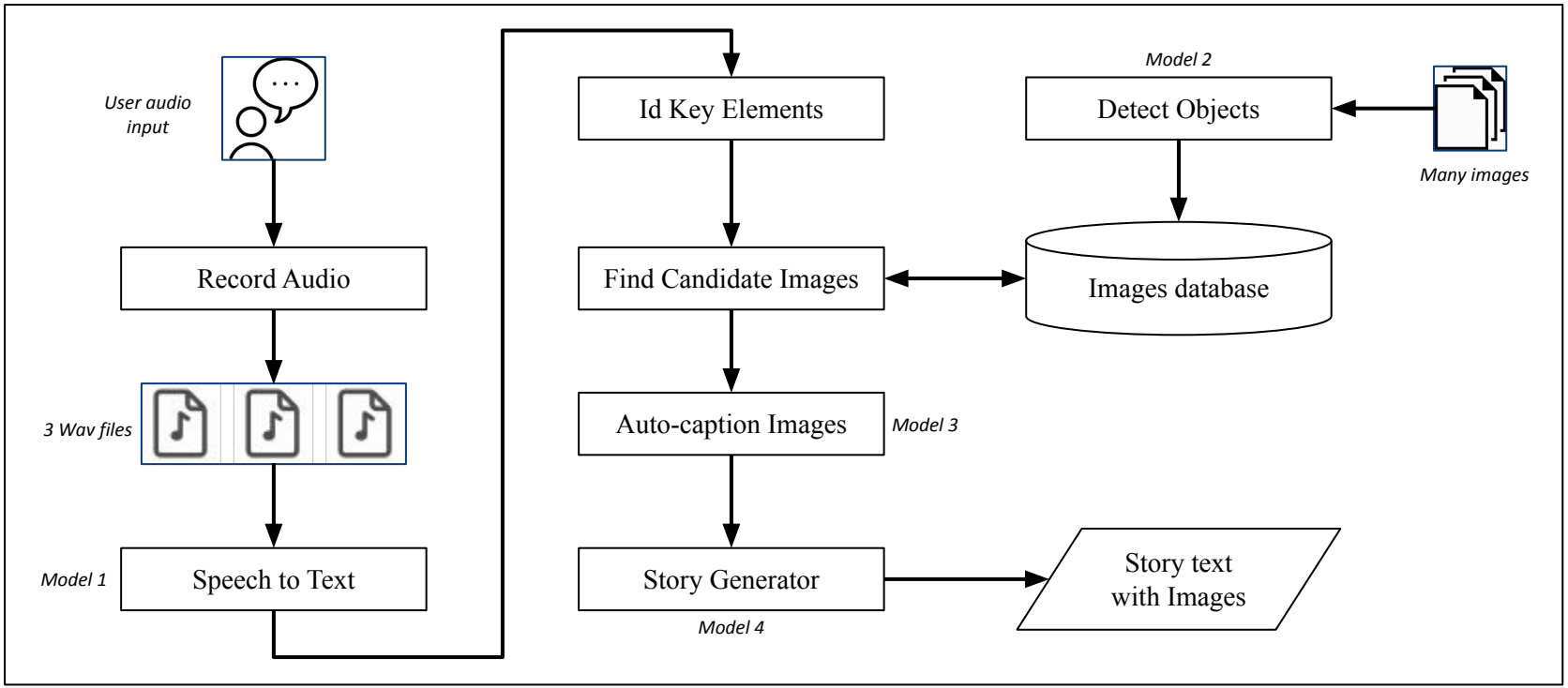
STAATLICH
ANERKANNTE
HOCHSCHULE

Thank you!

Please ignore slides below as they are for my future use in writing thesis

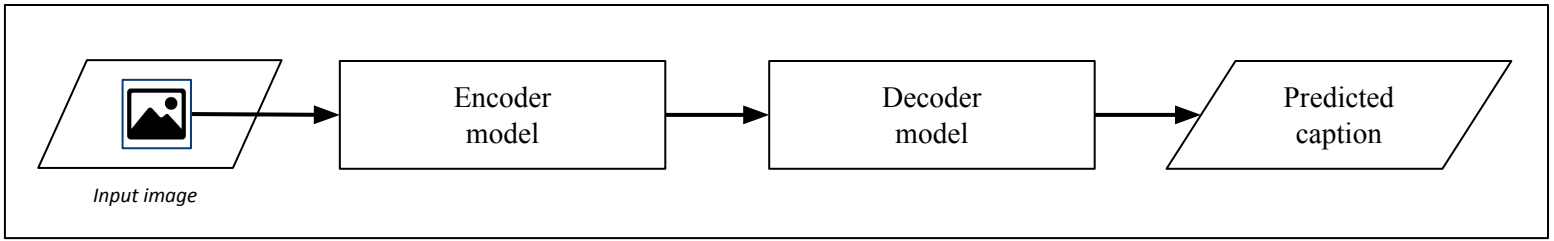
IGNORE:- For editing later if required
General arch of work

STAATLICH
ANERKANNTE
HOCHSCHULE

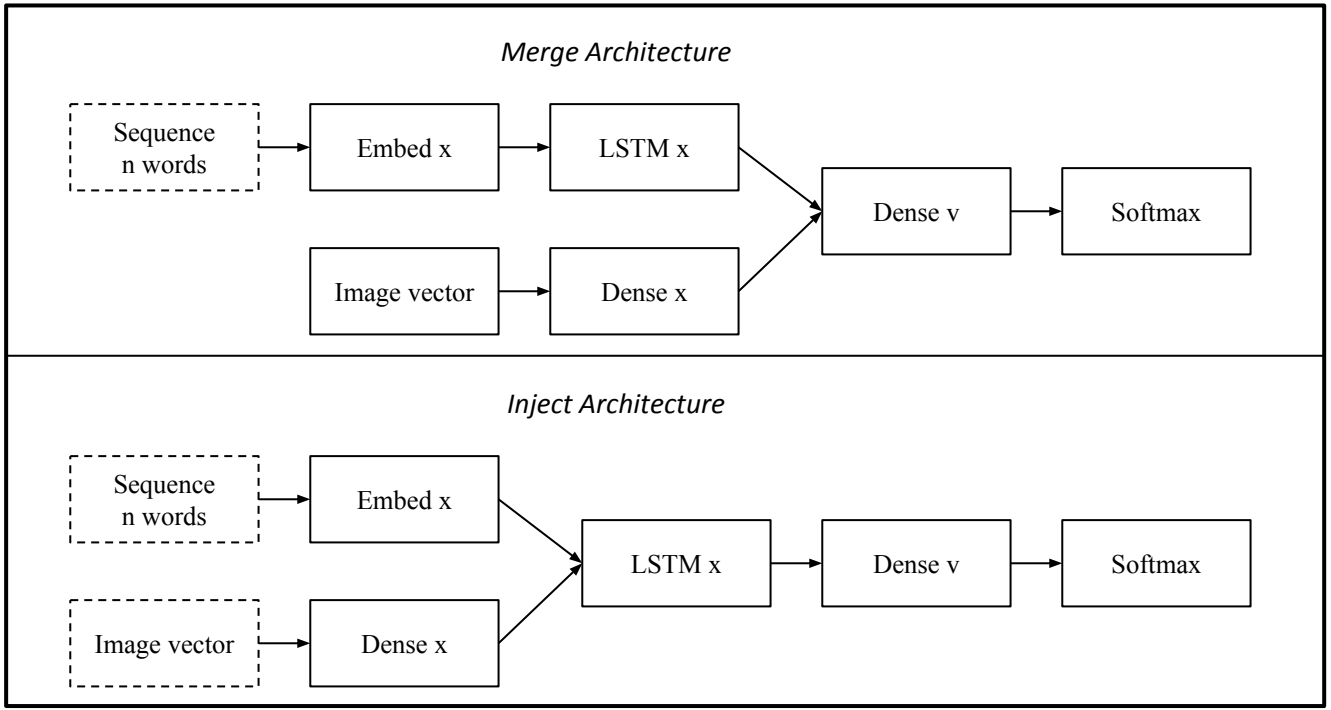


STAATLICH
ANERKANNTE
HOCHSCHULE

IGNORE:- For editing later if required
General Arch of Auto-caption generator



IGNORE:- For editing later if required
General Arch of Auto-caption generator



Source: From paper "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator" by Tanti et al. 2017
(https://arxiv.org/abs/1708.02043)