Question Answering Machine for Islamic Factual Questions Based on English Translation of Sahih Bukhari

Jafar Assegaf 1301154157

work.jafarassagaf@gmail.com

The basic idea of this project is to create a system that can answer Islamic factual question. With English translation of Hadith Sahih Bukhari book as the corpus for answers. The corpus is not a knowledge base, it is a document consisting 7356 verses of Hadith. Hence, it is inherently harder for the system to read and extract answer directly. The system able to receive an input posed in natural language such as "When was Prophert Muhammad born?" or "What are the best deed we can do to pleases Allah?". A keyword based search was used. First the user question went through preprocessing phase where keyword is extracted. Then, the keywords were used to search the corpus looking for matching verses. After that, the system used scoring and ranking to find the best matched verse and then return the corresponding answer for the question.

System Pros

-System is build using Object-Oriented Programming approach.

-System is build in a virtual environment.

-The code is written using pep8 syntax style for python.

-Some module has already passed unit testing process.

-System image has been built using docker build.

-Code Smell exist as few as possible.

System Functionality

-System able to take query in natural language as an input.

-System able to train SVM classifier using predefined data set.

-System able to process inputted query into a machine readable format for further processing.s

-System able to classify input based on predefined class using SVM.

-System able to extract features and keywords from input.

-System able to construct a query based on extracted features.

-System able to search database for answer using keywords matching.

-System able to construct answer and output it to the command line.

Code Example :

```
def setup_pack():

    base_dir = os.path.abspath(os.path.dirname(__file__))

    with chdir(base_dir):
        with io.open(os.path.join(base_dir, 'qna', 'about.py')) as fp:
            about = {}
            exec(fp.read(), about)

    with io.open(os.path.join(base_dir, 'readme.rst')) as f:
        readme = f.read()

    setup(name=about['__title__'], packages=find_packages(
    ), description=about['__summary__'], long_description=readme, version=about['__version__'])
```

Illustration 1: Setup module

```
def classify_question(en_doc=None, df_question_train=None, df_question_test=None):
    """ Determine whether this is a who, what, when, where or why question """

    if df_question_train is None:
        training_data_path = os.path.join(
            CORPUS_DIR, QUESTION_CLASSIFICATION_TRAINING_DATA)
        df_question_train = pandas.read_csv(
            training_data_path, sep='|', header=0)

    df_question_class = remove_irrelevant_features(df_question_train)

    if df_question_test is None:
        df_question_predict = get_question_predict_data(en_doc=en_doc)
    else:
        df_question_predict = get_question_predict_data(
            df_question_test=df_question_test)

    df_question_train = pre_process(df_question_train)
    df_question_predict = pre_process(df_question_predict)

    df_question_train, df_question_predict = transform_data_matrix(
        df_question_train, df_question_predict)
    # df_question_predict = transform_data_matrix_temp(df_question_predict)

    question_clf = load_classifier_model()

    # logger.debug("Classifier: {0}".format(question_clf))

    # predicted_class, svc_clf = predict_question_class(question_clf, df_question_predict)
    predicted_class, svc_clf = support_vector_machine(
        df_question_train, df_question_class, df_question_predict)

    if df_question_test is not None:
        return predicted_class, svc_clf, df_question_class, df_question_train
    else:
        return predicted_class
```

Illustration 2: Classify Question module

```
(final_env) j4sgf@Bos:~/Documents/Python Project/impal/test$ pytest
============================= test session starts ==============================
platform linux -- Python 3.6.7, pytest-4.4.1, py-1.8.0, pluggy-0.9.0
rootdir: /home/j4sgf/Documents/Python Project/impal
plugins: doubles-1.5.3
collected 3 items

test_classify_question.py .                                              [ 33%]
test_construct_query.py .                                                [ 66%]
test_extract_feature.py .                                                [100%]
```

Illustration 3: Unit testing for three modules

```
test/test_classify_question.py::TestClassifyQuestion::test_classify_question
  /home/j4sgf/Documents/Python Project/final_projectV1/final_env/lib/python3.6/s
ite-packages/sklearn/model_selection/_split.py:2053: FutureWarning: You should s
pecify a value for 'cv' instead of relying on the default value. The default val
ue will change from 3 to 5 in version 0.22.
    warnings.warn(CV_WARNING, FutureWarning)

-- Docs: https://docs.pytest.org/en/latest/warnings.html
==================== 3 passed, 27 warnings in 70.01 seconds ====================
```

Illustration 4: Test result

```
(final_env) j4sgf@Bos:~/Documents/Python Project/impal$ sudo docker build -t fin
al_version .
Sending build context to Docker daemon  18.41MB
Step 1/10 : FROM ubuntu
 ---> 47b19964fb50
Step 2/10 : FROM python:3.6-jessie
 ---> 93a2e69315ea
Step 3/10 : WORKDIR /final_projectV1
 ---> Using cache
 ---> c21182f33c9a
Step 4/10 : COPY . /final_projectV1
 ---> 09747585776a
Step 5/10 : RUN pip install --default-timeout=3000 --trusted-host pypi.python.or
g -r requirements-docker.txt
 ---> Running in b0a3a1a09726
Collecting docker==3.7.0 (from -r requirements-docker.txt (line 1))
```

Illustration 5: Building docker image

```
(final_env) j4sgf@Bos:~/Documents/Python Project/impal$ sudo docker images
[sudo] password for j4sgf:
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
final_version       latest              08b6f73a9b56        15 minutes ago
1.46GB
impal               latest              32fb27cd6fd6        7 days ago
1.75GB
```

Illustration 6: Docker image repository