OPTION #1

This repository includes python source codes, Jupiter files, and dataset containing 5500 annotated records [1] for the tasks #1 Building a question classification system. The program has been employed the method of LSTM [2] to classify textual questions.

1.Library requirements

- tensorflow (>=2.0)
- keras(>=2.2)
- numpy (>=1.12)
- nltk(>=3.2.0)

2. Structure of program

+ Data preprocessing
- Assigning max length of question by20.
- Splitting data into 80% for training and 20% for validation.
- Removing stopword.
- Transferring data sequences to tokens.
- Padding data tokens to the max length =20.

+ LSTM implementation
- Using tf.keras.Sequential model
- Adding an embedding layer expecting input vocab of size 3000, and output embedding dimension of size 64.
- Using a Dense layer with 7 units (for 0-6) and softmax activation (Corpus has 6 labels, the program usessparse_categorical_crossentropy as loss function and 0 should be a label as well, while the tokenizer object which tokenizes starting with integer 1, instead of integer 0.)

+ Training
- Use epochs=15 to train model.
- Model could be fitted at epochs=7,8.

+ Prediction
- Sample input text = “What metal has the highest melting point ?”

Output: [[1.2480495e-05 6.6372859e-03 1.4034165e-03 9.4354695e-01 3.0001828e-02
3.4741121e-03 1.4923892e-02]]
ENTY

3. Discussion

Why did simpler solution (e.g., SVM, kNN) fail on this dataset?
- Most of simpler solutions (SVM, kNN) had exploited bag of word (tfxidf) for classification. These solutions will have challenges because the type of questions in the dataset is short texts in which their features are sparseness.

4. References

[1] https://cogcomp.seas.upenn.edu/Data/QA/QC/

[2] Chung, J., Gulcehre, C., Cho, K.H., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014