**Note:**

1. Deadline for submission: Feb 5th, 2021
2. Please submit your homework to this [Sharepoint Path](https://microsoftapc.sharepoint.com/:f:/t/AISchoolChina/EkW9FBryx9BOpKXPqkIZMWQBQBMt54JYqWi6xtPSluqEBA?e=Ni4LDO). (Name the folder with your alias)
3. Task #1 & #2 are mandatory. Task #3 is optional.
4. Dataset: [datasets.zip](https://microsoftapc.sharepoint.com/:u:/t/AISchoolChina/EUKQS0b5ibBPrivtaRt72C8BwISbVlzMPN7FbAHUPKynJA?e=fcjOAP), password:micr

**Task #1:**

In this assignment, you need to train your own word embeddings with the Chinese sentiment analysis dataset and then give the visualization of the word vectors. Further, you need to leverage these pre-trained word embeddings to perform a sentiment classification task (a simple binary classification problem with positive and negative labels). To tackle this task, the following steps are for your reference.

**STEP1:** Leverage a skip-gram based **word2vec model** to train your own word vectors, and then visualize your pre-trained word vectors.

**The implementation procedures of this word2vec model:**

-Calculate the loss function and gradients

-Train your word vectors with gradient descent method. (e.g., SGD, BGD)

-Visualize your word vectors. (You can refer to <http://projector.tensorflow.org/> )

**STEP2:** Perform thesentiment classification task. A simple approach is to first compute the average of all word vectors within a sentence, and then treat this average vector as the features for this sentence so as to train a SVM classifier for addressing this sentiment classification task.

**Task Descriptions:**

* **Input:** Sentences and labels
* **Output:** The visualization of word vectors and the accuracy for this sentiment classification task.

**Data:** A Chinese sentiment analysis dataset *chnsenticorp*, where the label ‘1’ denotes the *positive* sentiment while the label ‘0’ denotes the *negative* sentiment. You can download this dataset according to the following link with a password.

**Task #2:**

In this assignment, you need to output the **Constituent Syntax Tree** of the given sentences. You can leverage the open-source Stanford Parser to address this task. Particularly, to tackle this task, the following steps are for your reference.

**STEP1:** Download the latest installation package (stanford-parser-full-2020-11-17.zip (version 4.2.0)) from Stanford official website (<http://nlp.stanford.edu/software/lex-parser.html#Download>) and unzip to get stanford-parser.jar and stanford-parser-4.2.0-models.jar. You can refer to the following link to obtain more information w.r.t. the environment configuration: [link](https://blog.csdn.net/zrx1024/article/details/87826531?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.control)

**STEP2: Perform Chinese word segmentation. A demo is shown as follows:**

from nltk.tokenize.stanford\_segmenter import StanfordSegmenter

segmenter = StanfordSegmenter()

str="改革春风吹满地，中国人民真争气。"

segmenter.default\_config('zh')

result = segmenter.segment(str)

print(result)

**STEP3:** **Perform Chinese parsering. A demo is shown as follows:**

from nltk.parse.stanford import StanfordParser

chi\_parser = StanfordParser()

sentence = '改革 春风 吹 满地 中国 人民 真 争气'

print(list(chi\_parser.parse(sentence.split())))

**Task Descriptions:**

* **Input:** The given100 sentences.
* **Output:** The constituent syntax tree of each sentence.

**Data:** You can download this dataset according to the following link with a password.

**Task #3:**

In this assignment, you need to implement a model for addressing the text matching task. You can refer to the following steps to achieve your goal.

**STEP1:** Clean the data (e.g., discarding special characters).

**STEP2:** Leverage a siamese network (e.g., siamese-lstm) or a pre-trained model, such as BERT, TinyBERT, or RoBERTa to tackle this text matching task.

**Task Descriptions:**

* **Input:** Text-text pairs with corresponding labels.
* **Output:** The accuracy and F1-score of the prediction on the test set.

**Data:** A Chinese dataset *lcqmc* for the text matching task, where the label ‘1’ denotes *matching* while the label ‘0’ denotes *mismatching*.

**Tips:**

1. The official BERT code url: <https://github.com/google-research/bert>

2. The official TinyBERT code url: <https://github.com/yinmingjun/TinyBERT>

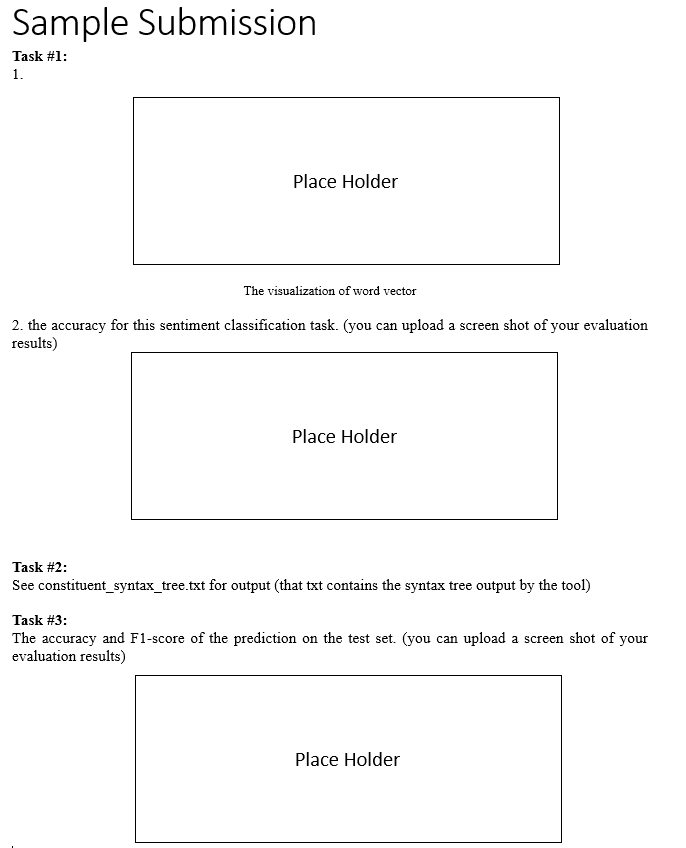
3. The reference performance of two models (i.e., BiLSTM and BERT) on this text matching dataset:

-BiLSTM: 83.4%

-BERT: 87%

**Sample Submission:**

Screen shot:



Sample in Submission folder: [link](https://microsoftapc.sharepoint.com/:f:/t/AISchoolChina/EvrZGKsvi2hDj2hoRODHSUsB6uBXu5ldhRRj1oSsNgECUQ?e=zv7EGn).