Python_Lesson7: Python Programming

Please don't forget to submit your feedback after the class. This helps a lot in increasing effectiveness of the course.
Use the following link to submit your feedback:
https://docs.google.com/forms/d/e/1FAIpQLSdmJkDgBMxr4qv73c9y5k1jtky44-sMmOI1v1jFtNEbUJ6H9A/viewform

Lesson Overview:

In this lesson we will focus on text processing like unigram, bigram, trigram, tokenization, pos tagging, lemmatization, normalization, entity extraction, language model. Learning these features will help us for more meaningful project as document classification, spelling corrector, document summarization, etc.

Use Case Description:

In this use case, we will learn how to correct the mistyped of words in a sentence.
Spelling corrector is about using some of the NLP features we learned during the class, then correcting a mistyped word. Thus, students can see the right application of these features in a project.

Programming elements:

Basic NLP techniques like unigram, bigram, trigram, tokenization, pos tagging, lemmatization, normalization, entity extraction, language model

Source Code:

https://umkc.box.com/s/8vygyn9iqj8ut6k8vn434jmpfoldde20

In class programming:

In class, we further work on the tokenization, pos-tagging, entity extraction, bigram and trigram.
For all the exercises import the right module from NLTK. You need to go through the slides to find them.

1. Extract the following web URL text using BeautifulSoup
https://en.wikipedia.org/wiki/Google
2. Save it in input.txt

3. Apply the following on the text and show output:
   a. Tokenization
   b. POS
   c. Stemming
   d. Lemmatization
   e. Trigram
   f. Named Entity Recognition
4. Change the classifier in the given code to
   a. **KNeighborsClassifier** and see how accuracy changes
   b. change the tfidf vectorizer to use bigram and see how the accuracy changes
   TfidfVectorizer(ngram_range=(1,2))
   c. Put argument stop_words='english' and see how accuracy changes

ICP Submission Guidelines (for In Class students):

1. ICP Submission is in pairs of two students.
2. Once completed, must be presented to TA or Instructor before the completion of the class

3. Submission after class is considered as a late submission. (Check the late submission policy in the syllabus)
4. ICP Code with brief explanation should be pushed to GitHub. Submit GitHub link through the Feedback Form: https://docs.google.com/forms/d/e/1FAIpQLSdmJkDgBMxr4qv73c9y5k1jtky44-sMmOI1v1jFtNEbUJ6H9A/viewform

Online Submission Guidelines (for Online students):
1. Submit your source code and documentation to GitHub and represent the work through wiki page properly (submit your screenshots as well. The screenshot should have both the code and the output)
2. Comment your code appropriately
3. Video Submission (2 – 3 min video showing the demo of the ICP, with brief voice over on the code explanation)
4. Submission after class is considered as a late submission. (Check the late submission policy in the syllabus)
5. Use the following Google link to submit your ICP # (GitHub wiki page link for ICP #): https://docs.google.com/forms/d/e/1FAIpQLSdmJkDgBMxr4qv73c9y5k1jtky44-sMmOI1v1jFtNEbUJ6H9A/viewform

Evaluation Criteria:
1. Completeness of Features
2. Code Quality (https://en.wikipedia.org/wiki/Best_coding_practices)
3. Time
4. Feedback Submission

Note: *Cheating, plagiarism, disruptive behavior and other forms of unacceptable conduct are subject to strong sanctions in accordance with university policy. See detailed description of university policy at the following URL: https://catalog.umkc.edu/special-notices/academic-honesty/*