Fall 2021: CSEE5590/490 – Special Topics

Python and Deep Learning Module-2 - ICP-7

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.

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

Source Code:
Provided in your assignment folder and assignment repo.

Assignment:
For all the exercises import the right module from NLTK. You need to go through the slides to find them.

1. On the given file (text_classification) apply the following:
   a) **SVM** and see how accuracy changes.
   b) **KNeighborsClassifier** and see how accuracy changes.
   c) Print the **classification_report** for each classifier.
      a. Example image provided by end of the file.
   d) Set the tfidf vectorizer parameter to use bigram and see how the accuracy changes
      TfidfVectorizer(ngram_range=(1,2))
      a. You can apply this step to one classifier only.
   e) Set tfidf vectorizer argument to use stop_words='english' and see how accuracy changes.
      a. You can apply this step to one classifier only.

2. Extract information from the following web URL using BeautifulSoup library:
   https://en.wikipedia.org/wiki/Natural_language_processing
   (optional) save the result or extracted paragraphs in a file "input.txt" and use it for your work.

   - Apply this to a subset of the paragraphs extracted from the page.

     - Tokenization
     - POS
     - Stemming
     - Lemmatization
     - Trigram
     - Named Entity Recognition
     - **Plot the Frequencies of words**.

** Follow the IPC rubric guidelines.

Submission Guidelines:

1. Once finished present your work to TA during class time.
2. Once evaluated submit your source code and documentation to GitHub and represent the work in a ReadMe file properly (short summary for the ICP).

After class submission:

1. Complete your work and submit to your repo before the deadline.
2. Record a short video (1~3) minute, explaining the technical part and method used.
3. Add video link to ReadMe file.

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/*

	precision	recall	f1-score	support
alt.atheism	0.80	0.52	0.63	319
comp.graphics	0.81	0.65	0.72	389
comp.os.ms-windows.misc	0.82	0.65	0.73	394
comp.sys.ibm.pc.hardware	0.67	0.78	0.72	392
comp.sys.mac.hardware	0.86	0.77	0.81	385
comp.windows.x	0.89	0.75	0.82	395
misc.forsale	0.93	0.69	0.80	390
rec.autos	0.85	0.92	0.88	396
rec.motorcycles	0.94	0.93	0.93	398
rec.sport.baseball	0.92	0.90	0.91	397
rec.sport.hockey	0.89	0.97	0.93	399
sci.crypt	0.59	0.97	0.74	396
sci.electronics	0.84	0.60	0.70	393
sci.med	0.92	0.74	0.82	396
sci.space	0.84	0.89	0.87	394
soc.religion.christian	0.44	0.98	0.61	398
talk.politics.guns	0.64	0.94	0.76	364
talk.politics.mideast	0.93	0.91	0.92	376
talk.politics.misc	0.96	0.42	0.58	310
talk.religion.misc	0.97	0.14	0.24	251
accuracy			0.77	7532
macro avg	0.83	0.76	0.76	7532
weighted avg	0.82	0.77	0.77	7532

Classification Report