Step One: Change tokenize txt files and read in csv files

* Run Preprocess.py
  + Input:
    - Data/OpinionArticles-text.txt
    - Data/NACC27k-text.txt
  + Procedure:
    - Default read text in Spanish
    - Custom tokenizer and apply to txt
    - Filter out tokens which are not noun, verb or adj
    - Generate csv files
  + Output:
    - Processing 13331 word document (OpinionArticles)
    - Processing 26941 word document (NACC)
    - Output/OpinionArticles-TASI.csv
    - Output/NACCf27k-text-TASI.csv
  + Tip:
    - In the original NACC27k-text.txt, there are three sentences that include different combinations of numbers:
      * Si la Compañera Jefa sacó 11. 700. 000 votos y Binner sacó 3. 700. 000 quiere decir que el escalón fue de 8 millones de votos.
      * Publicado: 23. 11. 2012 | 11: 16
      * Agencias AFP y EFE 05. 02. 2013 | 11: 45
    - These three sentences are removed from current NACC27k-text.txt

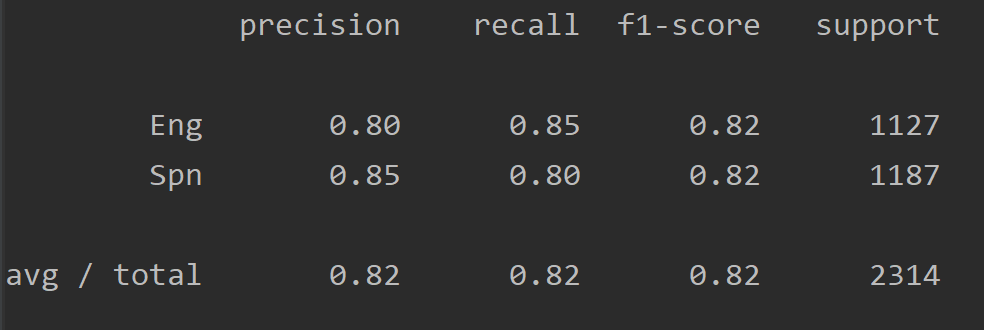
Step Two: Merge the gold standard label to each token in generated csv file

* Run merge\_and\_check.py
  + Input:
    - Output/OpinionArticles-TASI.csv
    - Output/NACCf27k-text-TASI.csv
    - Data/OpinionArticlesRetokenized-GS.csv
    - Data/NACC27k-GoldStandard.tsv
  + Procedure:
    - Default read text in Spanish
    - Custom tokenizer and apply to txt
    - Filter out tokens which are not noun, verb or adj
    - Generate csv files
  + Output:
    - Processing 13331 word in gold standard document (OpinionArticles)
    - Processing 27058 word in gold standard document (NACC)
    - Number of non-matching token for OpinionArticles is 0
    - Output/diff\_df\_OA.csv
    - Number of non-matching token for NACC is 85
    - Output/diff\_df\_NACC.csv
    - Output/target\_full\_df\_OA.csv
    - Processing 2669 word in target document (OpinionArticles)
    - Output/target\_full\_df\_NACC.csv
    - Processing 4188 word in target document (NACC)

Things to improve here:

* Check the difference token between spacy tokenizer and gold standard. (see file Output/diff\_df\_NACC.csv)
* Correct format of gold standards and make consistent: include token, language(Spn, Eng, French, German, etc), Anglicism(TRUE, FALSE), Adapted(TRUE, FALSE) (see files: Data/OpinionArticlesRetokenized-GS.csv, Data/NACC27k-GoldStandard.tsv)

Step Three: Compare high frequent lemma list for English and Spanish and train a N-gram model to give probability on language label

* Run FreqLemmas\_Compare.py
  + Input:
    - Data/Training/EngHighFreqLemmas.txt
    - Data/Training/SpnHighFreqLemmas.txt
    - Output/target\_full\_df\_OA.csv
    - Output/target\_full\_df\_NACC.csv
  + Procedure:
    - Read in two high frequent list separately and assign labels
    - Merge English and Spanish high frequent list to check overlapping
    - Merge two lists to prepare for N-gram training
    - Check the performance of the n-gram model
    - Assign probability to tokens from opinion articles and NACC
  + Output:
    - Processing 4999 rows in Eng High Freq
    - Processing 5000 rows in Spn High Freq
    - Processing 4374 unique words in Eng High Freq
    - Output/EngHighFreqLemmas.csv
    - Processing 4879 unique words in Spn High Freq
    - Output/SpnHighFreqLemmas.csv
    - Number of tokens in both high frequency list is 136
    - Output/diff\_df\_High\_Freq.csv
    - 
    - Output/target\_full\_df\_OA\_Prob.csv
    - Output/target\_full\_df\_NACC\_Prob.csv

Things to improve here:

* Check the overlapping tokens in both English and Spanish frequent list. (see file Output/diff\_df\_High\_Freq.csv)