Contents

[Step One: Change tokenize txt files and read in csv files 1](#_Toc34583387)

[Step Two: Merge the gold standard label to each token in the generated csv file 1](#_Toc34583388)

[Step Three: Compare high frequent lemma list for English and Spanish and train an N-gram model to give probability on language label 2](#_Toc34583389)

[Step Four: Compare the difference between English and Spanish lemma on tokens from opinion articles 3](#_Toc34583390)

[Step Five: Compare two Spanish Dictionaries 4](#_Toc34583391)

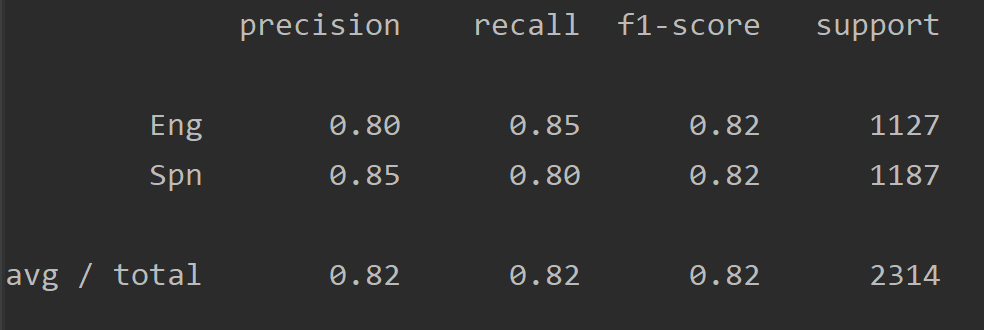
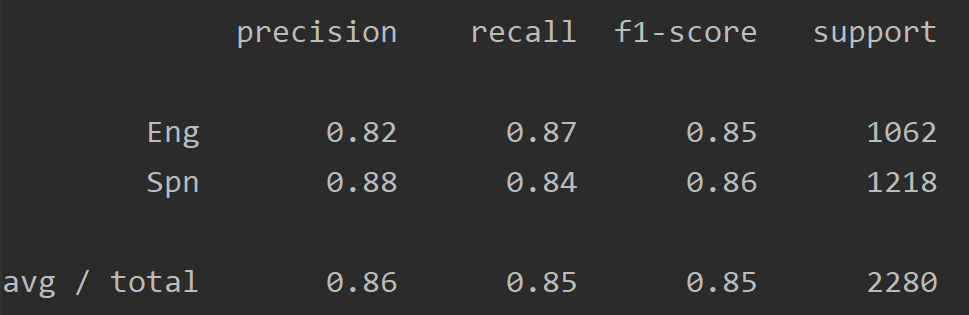
# Step One: Change tokenize txt files and read in csv files

* 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
* Other concerns and implementation
  + Customized tokenizer doesn’t work on strange number format
    - 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 the 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 words in the gold standard document (OpinionArticles)
    - Processing 27058 words in the 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 words in the target document (OpinionArticles)
    - Output/target\_full\_df\_NACC.csv
    - Processing 4188 words in the target document (NACC)
* Other concerns and implementation
  + Check the difference token between spacy tokenizer and the 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 an 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 lists separately and assign labels
    - Merge English and Spanish high frequent list to check overlapping words
    - 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 4877 unique words in Spn High Freq
    - Output/SpnHighFreqLemmas.csv
    - Number of tokens in both high frequent lists is 134
    - Output/diff\_df\_High\_Freq.csv
    - Before fixing the overlapping list:
      * 
    - After fixing the overlapping list:
      * 
    - Output/target\_full\_df\_OA\_Prob.csv
    - Output/target\_full\_df\_NACC\_Prob.csv
* Other concerns and implementation
  + Check the overlapping tokens in both English and Spanish frequent list. (see file Output/diff\_df\_High\_Freq.csv)
    - I have reviewed the list. All items should be considered Spanish except “bar” and “club” which are Anglicisms. I’m not sure how to mark these and I don’t have editing permission in the new TASI repo but you could just remove “bar” and “club” from the Spanish High Freq list to solve this problem. All other overlapping tokens should be treated as Spanish.
    - Implementation:
      * Manually removed “bar” and “club” from the Spanish High Freq list. (change from 4879 to 4877 words)
      * Add script to exclude the overlapping list from the English High Freq list. (remains 4240 words)
      * The result above is updated: the accuracy of the N-gram language detection model increases after fixing the overlapping list.

# Step Four: Compare the difference between English and Spanish lemma on tokens from opinion articles

* Run EDA\_OA.py
  + Input:
    - Data/OpinionArticles-text.txt
  + Procedure:
    - Read text in Spanish and English (using "es\_core\_news\_sm" and "en\_core\_web\_sm")
    - Custom tokenizer and apply to txt
    - As a result, tokens from both es and en are the same. But some tokens have different lemma using es or en.
    - Merge en and es data frame by token and lemma and wrote them in this excel file. (I leave other columns like pos, tag, NE in the file just for reference, and they are not working in the merge.)
  + Output:
    - Processing 13331-word OpinionArticles using spanish tokenizer
    - Processing 13331-word OpinionArticles using english tokenizer
    - Number of non-matching token for OpinionArticles using different tokenizer is 4000
    - Output/diff\_token\_es\_en.csv
  + Other concerns and implementation
    - Check the non-overlapping tokens based on English and Spanish tokenizer lemma. (see file Output/diff\_token\_es\_en.csv)

# Step Five: Compare two Spanish Dictionaries

* Run Dictionaries\_Compare.py
  + Input:
    - Data/Training/SpnHighFreqLemmas.txt
    - Data/Dictionaries/lemario-espanol-2002-10-25.txt
    - Data/Dictionaries/lemario-utf8.txt
  + Procedure:
    - Read SpnHighFreqLemmas.txt and lemario-espanol-2002-10-25.txt in Spanish
    - Since in dictionary lemario-utf8.txt, there are words containing multiple words in one word and words with dash. I separate lemario-utf8.txt into three files: Data/Dictionaries/lemario\_one\_word.txt, Data/Dictionaries/lemario\_multiple\_words.txt, and Data/Dictionaries/lemario\_dash\_word.txt
    - Compare lemario-espanol-2002-10-25.txtwith lemario\_one\_word.txt
    - Write the same words in a file and different words in another file
  + Output:
    - Processing 80383 rows in lemario-espanol
    - Processing 85918 rows in lemario-utf8 including multiple-word and dash-word
    - Processing 85098 rows in one word from lemario-utf8.txt

Processing 80383 unique words in lemario-espanol-2002-10-25

* + - Processing 85098 unique words in one word from lemario-utf8.txt
    - Number of same tokens in two dictionaries is 79431
    - Output/same\_df\_dictionaries.csv
    - Number of different tokens in two dictionaries is 6619
    - Output/diff\_df\_dictionaries.csv
  + Other concern and implementation
    - Check how can we use words containing multiple words in one word and words with dash. (see file Data/Dictionaries/lemario\_multiple\_words.txt, and Data/Dictionaries/lemario\_dash\_word.txt)
    - Check the overlapping and non-overlapping tokens from two Spanish dictionaries. (see file Output/same\_df\_dictionaries.csv, Output/diff\_df\_dictionaries.csv)