To Automatically Annotate New Data

**Files:**

* Person Number Disagreement Full and Final Documentation (*This document*)
  + Reference this for more information on the rules, process, data, reasoning, etc
  + This also contains the best results we have been able to generate so far
* Person Number Agreement (Wilermine possible combinations for pnd).ibynb
  + Use this to run unannotated text files and label them with person number disagreement
  + There may be issues if the column labels do not match the code, e.g. ‘text’ instead of ‘sentence’; this is alright, just change either the name of the column or the column names in the code
* PNA\_45\_180\_3\_27\_2024.txt

**Instructions:**

To run the provided code successfully, you'll need to ensure that you have the required libraries and resources installed. Here's a step-by-step guide on what you need:

Python Environment: Make sure you have a Python environment set up. You can download and install Python from the official website: <https://www.python.org/downloads/>

1. Create a new Folder with a specific name:

Download all the above files to this folder

Before running the main file: **Person Number Agreement (Wilermine possible combinations for pnd).ibynb** Download the required libraries.

1. Required Libraries: Install the necessary Python libraries using pip. Open your terminal or command prompt and run the following commands:

Command:

**pip install numpy pandas scikit-learn matplotlib seaborn spacy nltk gensim joblib**

1. Download Language Model for spaCy: You are using the spaCy library for natural language processing. You'll need to download the English language model. Run the following command in your terminal:

Command:

**python -m spacy download en\_core\_web\_sm**

1. Download NLTK Resources: NLTK requires additional resources such as stop words. Run the following Python code to download the required resources:

Command:

**import nltk**

**nltk.download('stopwords')**

1. Make sure to keep the below files in the same folder along with the Main file which are:

i. PNA\_45\_180\_3\_27\_2024.txt

To test:

1. Download the ' PNA\_45\_180\_3\_27\_2024.txt’.
2. Open and run the main file (**M Person Number Agreement (Wilermine possible combinations for pnd).ibynb**) in Jupyter Notebook.
3. The annotated file will be save as personNum.txt.
4. After running this file, compare the results of multNeg.txt with personNumResults.txt; these should be the same.

**Data:**

Each sentence was annotated by human annotators for use in developing the tagger. PNA\_45\_180\_3\_27\_2024.txt contains a combination of sentences with and without person number disagreement, though predominantly without.

**Rules:**

A rule tagging a sentence means it fits the requirements of that rule, making it true. If it is true, pna is set to 1. If none of the rules apply, pna is set to 0, indicating there is no person number disagreement

The rules were parsed using SpaCy’s dependency parser.

First, the tagger identifies a potential subject, a potential verb, and the dependencies they have between them. If those dependencies meet a certain criteria, the sentence could potentially have person number disagreement. Then, the rules are applied to the sentence determine whether they do or do not have person number disagreement.

1) subject is not plural and is third party singular pronoun followed by non-third party singular verb

if ((word.tag\_ != 'NNS' and word.tag\_ != 'NNPS' and (word.text.lower() in thirdparty\_singular\_pronouns) and nextword.tag\_ =='VBP')):

Sentence1.pna = 1

2) subject is singular followed by a base or non-third party singular verb which are not 'was'

if ((word.tag\_ == 'NN') and (nextword.tag\_ == 'VB' or nextword.tag\_ == 'VBP' and nextword.text.lower() != 'was')):

Sentence1.pna = 1

3) subject is plural followed by 3rd person singular verb

if (((word.tag\_ == 'NNS' or word.tag\_ == 'NNPS')) and nextword.tag\_ =='VBZ'):

Sentence1.pna = 1

4) subject is plural or in second person followed by 'was'

if ((((word.text.lower() in thirdparty\_plural\_pronouns) or (word.text.lower() in personal\_object\_pronouns\_plural)) or (word.pos\_ == 'NUM' and (word.text != "1" and word.text.lower() != "one")) or (word.text.lower() in pluralquantifiers) or((word.tag\_ == 'NNS' or word.tag\_ == 'NNPS') and word.pos\_ != 'PROPN')) and (nextword.text.lower() == "was")):

Sentence1.pna = 1

5) singular pronoun subject followed by plural verb

if ((word.text.lower() in other\_singular\_pronouns) and nextword.text.endswith('s') and nextword.text.lower() != 'was'):

Sentence1.pna = 1

6) compound subject connected through 'and' where the verb is not plural

if parsed[i+1].text == "and" and (nextword.tag\_ =='VBZ' or nextword.text == 'was'):

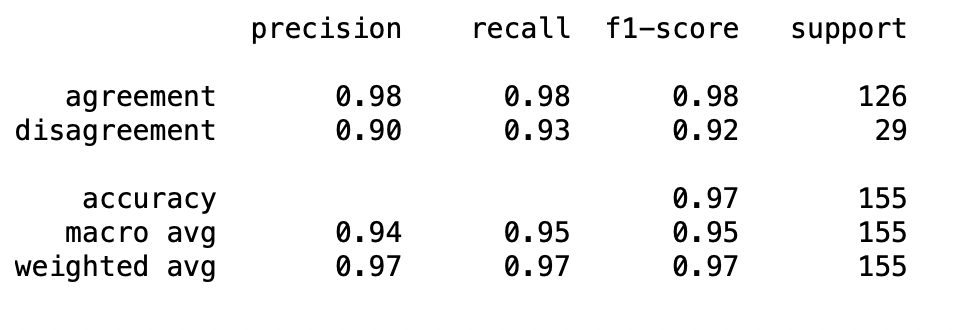
Sentence1.pna = 1

*Key/Definitions*

* 1: person number disagreement, 0: not person number disagreement
  + If there are multiple potential cases of disagreement in a sentence, tags as “1” if any are found to be disagreement, “0” if none are

**Results**

Below is the classification report of the Person Number Disagreement tagger.



As you can see above, the accuracy, recall, and precision are all quite high (at least 0.9). Observing the falsely predicted values, there are a mix of person number agreement sentences falsely predicted as disagreement and vice versa.