Homework 1: Reddit Cooking NER and Data Cleanup

Eric Phann

Approach and Reasoning

First steps

I set up Prodigy and looked through the documentation to get a general understanding of recipes and tips I would need (the NER flowchart was great). I glanced over the .jsonl files to get a general feel of each dataset. I reviewed the homework objective and guidelines and got to work.

Pydata Annotated Reddit comments

I trust the annotations, as I am sure my annotations are just as, if not more, inconsistent and noisy. The only concern I have is differing annotation guidelines, as there were none provided for this dataset. When training a model using a 70/30 split it produced a 0.57 F1 score [1].

Unlabeled Reddit comments

I manually labeled 150 of the unlabeled Reddit comments. I accepted 149 documents and ignored 1 bot command. My final annotation guidelines are on the next page.

GPT3.5 Zero Shot annotations

I did not clean or alter this dataset at all.

Question

When using my manually labeled Reddit comments as the evaluation set, which will perform better: Pydata trained model, zero shot trained model, or using both sets to train a model?

Training the Models

The Pydata model w/ manual annotation eval set had an F1 score of 0.54 [2].
The zero shot model w/ manual annotation eval set had an F1 score of 0.39 [3].
The Pydata + zero shot model w/ manual annotation eval set had an F1 score of 0.56 [4].

Conclusion

The model combining both the Pydata dataset and zero shot data performed the best on my manual annotations as an evaluation set. The Pydata model performed about the same individually and the zero shot model performed somewhat worse individually. This may be due to inconsistencies in annotation guidelines: perhaps mine are more similar to Pydata than GPT3.5's.

Annotation Guidelines

Entity Definitions:

- **DISH**: the final product of a recipe; the end result; whole
- **INGREDIENTS**: the required components of a dish; things that go in a dish; part
- **EQUIPMENT**: tools used to make a dish/complete a recipe

Cases:

- **Potentially overlapping spans**: Case-by-case try to include only the most relevant or important information. e.g., *Korean soups* is not sufficient as a dish nor *tofu Korean soups* so we take the entire *spicy silken tofu Korean soups*
- **More than one (nested) entity:** Do not break the noun phrase into its parts to classify it as a different or multiple entity (this is span classification), consider the whole noun phrase in the context of the document. e.g., do not take *tofu* as an ingredient from *spicy silken tofu Korean soups*
- **Non-English named dish**: Include as long as it is romanized (no matter how rough). e.g., *soondubu-jigae* and not 순두부찌개
- **Dish or ingredient?**: Do the best you can with the context of the document. If not enough context, use best judgment with entity definitions above e.g., *seafood*, *chicken*
- **Measurements**: Do not include e.g. *2 tsps* or *heavy grindings [of salt]*
- **Adjectives/Modifiers**: Try to include objective adjectives and not subjective adjectives when modifying entities e.g., *big/small* rather than *good/bad* and *minced garlic,* and only when critical (recipes don't usually put *heavy grindings of salt*
- **Processed/prepped ingredients**: I think *sliced and layered potatoes* is more important than simply *potatoes* (see first bullet point)
- **Plurality:** Include if needed e.g. *knives*
- **Brand names:** Omit unless the brand name is a stand-in e.g., *Keurig* for a coffee maker is OK but omit *Kuhn Rikon* from *Kuhn Rikon peeler* and *Hellman's* from mayo
- **Colloquial/implicit things**: Use best judgment and include where it makes sense e.g., I would include *12 inch cast iron [skillet]* as equipment
- **Sides**: Ingredient, unless the side could be standalone as as dish e.g. *mac-and-cheese* or *flat dumplings*
- **Toppings**: Ingredient
- **General/broad entities:** Include and use best judgment if it is the only thing available in the doc e.g. *soups*, *stews*, *veggies*
- **Meta-entity**: Do not include *food*, *dish*, *ingredient* or *equipment*
- **Spelling-errors**: Use best judgment, if it is small a type i.e., the model is probably able to generalize it, I included it. If it is bad and confusing, like *flower* instead of *flour*, I did not include it.
- **Sink or sink?**: Do not force entities to be something they are not e.g. *"I didn't sink too much into them"* Sink here does not function as an equipment.
- **Articles**: Do not include articles like *a* or *the*
- **Abbreviations**: Include shorthand/contractions e.g. *mozz* for *mozzarella*

Appendix

[1] Pydata model 70/30 split

```
============================ Training pipeline ============================
Components: ner
Merging training and evaluation data for 1 components
  - [ner] Training: 829 | Evaluation: 354 (30% split)
Training: 813 | Evaluation: 350
Labels: ner (3)
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     17.14    0.00    0.00    0.00    0.00
  0     200        184.77   2840.14   37.17   53.94   28.35    0.37
  1     400        280.02   2355.38   44.28   52.09   38.51    0.44
  2     600        203.81   2177.86   51.64   58.00   46.55    0.52
  3     800        358.76   2007.13   55.67   61.49   50.85    0.56
  4    1000        375.53   1969.40   54.58   55.76   53.45    0.55
  5    1200        441.68   1590.17   53.62   57.86   49.96    0.54
  7    1400        507.00   1521.64   53.89   55.57   52.32    0.54
  9    1600        602.71   1155.34   54.06   57.09   51.34    0.54
 12    1800        903.21   1155.62   54.73   58.70   51.26    0.55
 15    2000        708.79    738.27   56.51   60.26   53.21    0.57
 19    2200        841.17    695.73   55.80   60.98   51.42    0.56
 24    2400        828.04    570.04   55.53   58.75   52.64    0.56
 29    2600        828.75    476.07   54.28   60.12   49.47    0.54
 34    2800        982.43    419.80   55.11   59.58   51.26    0.55
 40    3000        913.08    350.03   57.05   60.13   54.26    0.57
 45    3200       1012.40    415.77   57.00   60.02   54.26    0.57
 50    3400        942.79    322.74   55.62   56.17   55.08    0.56
 55    3600        990.98    299.73   52.74   55.64   50.12    0.53
 61    3800       1630.30    277.54   54.91   60.95   49.96    0.55
 66    4000       1423.05    281.34   56.26   58.51   54.18    0.56
 71    4200       1223.02    279.80   56.00   58.31   53.86    0.56
 76    4400       1107.20    215.86   55.97   59.85   52.56    0.56
 82    4600        929.52    184.59   56.66   60.18   53.53    0.57
```

[2] Pydata model w/ manual annotation eval set

```
============================ Training pipeline ============================
Components: ner
Merging training and evaluation data for 1 components
  - [ner] Training: 1183 | Evaluation: 149 (from datasets)
Training: 1163 | Evaluation: 149
Labels: ner (3)
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     69.71    0.69    0.83    0.59    0.01
  0     200         77.02   3186.17   34.22   48.13   26.55    0.34
  0     400        199.54   2298.23   43.05   50.60   37.46    0.43
  1     600        414.43   2525.42   48.06   53.02   43.95    0.48
  2     800        203.77   2198.28   50.81   56.73   46.02    0.51
  2    1000        281.06   2232.41   54.87   54.87   54.87    0.55
  3    1200       1468.56   2316.59   53.20   60.00   47.79    0.53
  5    1400        537.92   2145.42   45.76   53.78   39.82    0.46
  6    1600        750.15   1946.66   51.91   59.54   46.02    0.52
  8    1800        704.45   1957.44   55.78   60.27   51.92    0.56
 10    2000        763.90   1577.35   56.65   55.00   58.41    0.57
 12    2200        821.09   1314.60   56.93   56.93   56.93    0.57
 16    2400        933.41   1037.42   59.01   64.24   54.57    0.59
 19    2600        982.68    936.36   54.01   56.63   51.62    0.54
 23    2800       1001.50    783.11   53.64   56.54   51.03    0.54
 26    3000        933.28    652.55   53.40   59.14   48.67    0.53
 30    3200        839.46    564.10   55.73   58.63   53.10    0.56
 33    3400       1105.81    594.91   56.88   60.47   53.69    0.57
 37    3600        930.20    514.42   55.40   59.00   52.21    0.55
 40    3800        900.77    430.84   54.69   59.31   50.74    0.55
 44    4000       1162.34    467.57   53.98   57.28   51.03    0.54
```

[3] Zero shot model w/ manual annotation eval set

```
============================ Training pipeline ============================
Components: ner
Merging training and evaluation data for 1 components
  - [ner] Training: 500 | Evaluation: 149 (from datasets)
Training: 500 | Evaluation: 149
Labels: ner (3)
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     48.64    0.00    0.00    0.00    0.00
  0     200        255.58   3570.06    6.82   92.31    3.54    0.07
  1     400        106.68   2185.39   20.21   33.56   14.45    0.20
  3     600        488.49   2279.39   33.05   38.58   28.91    0.33
  4     800        309.52   1921.13   40.31   42.86   38.05    0.40
  6    1000        523.03   1763.04   36.76   47.22   30.09    0.37
  8    1200        621.15   1469.26   31.58   41.04   25.66    0.32
 11    1400        741.78   1217.53   36.36   47.39   29.50    0.36
 14    1600        706.81    829.82   37.57   47.73   30.97    0.38
 17    1800        868.96    732.75   35.08   45.33   28.61    0.35
 22    2000        865.78    539.14   39.18   46.37   33.92    0.39
 27    2200        846.05    467.08   38.14   45.68   32.74    0.38
 33    2400        759.93    345.44   39.03   47.08   33.33    0.39
```

[4] Pydata + zero shot model w/ manual annotation eval set

```
============================ Training pipeline ============================
Components: ner
Merging training and evaluation data for 1 components
  - [ner] Training: 1683 | Evaluation: 149 (from datasets)
Training: 1577 | Evaluation: 149
Labels: ner (3)
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     50.07    0.00    0.00    0.00    0.00
  0     200        119.23   2817.73   33.99   43.18   28.02    0.34
  0     400        229.33   2658.75   42.11   54.72   34.22    0.42
  1     600        149.27   2612.76   52.02   57.50   47.49    0.52
  1     800        212.99   2733.41   43.26   54.22   35.99    0.43
  2    1000        326.03   3001.78   51.22   57.30   46.31    0.51
  2    1200        412.35   2903.01   55.86   58.58   53.39    0.56
  3    1400        610.71   3428.28   52.53   54.81   50.44    0.53
  4    1600        662.52   3388.04   57.18   56.29   58.11    0.57
  5    1800        829.21   3373.35   59.81   65.72   54.87    0.60
  7    2000       1075.51   3635.62   58.53   62.33   55.16    0.59
  9    2200       1289.59   3298.56   57.79   59.32   56.34    0.58
 11    2400       1699.17   3185.84   52.93   57.19   49.26    0.53
 13    2600       1828.18   2690.41   57.40   58.82   56.05    0.57
 16    2800       2045.62   2314.20   57.88   63.60   53.10    0.58
 18    3000       1744.21   1805.41   52.98   56.52   49.85    0.53
 21    3200       1804.07   1621.65   52.72   57.49   48.67    0.53
 23    3400       1638.17   1369.48   56.22   58.65   53.98    0.56
```