Homework 2: Recipe Relevancy Classifier

Eric, Sydney, Jake, Kristen, Yaxin (Group 6)

Introduction

The task is to build a text classifier that can filter documents as either **relevant** or **not relevant** to recipe writing. The following datasets of Reddit r/Cooking comments were provided:

- 5000 unlabelled training dataset (homework2_train.jsonl)
- 200 unlabelled evaluation dataset (homework2_evaluation.jsonl)

Evaluation Dataset

Using Prodigy and ngrok, we each labeled the evaluation dataset (hmwk2-eval-1000.jsonl). General discussions and notes were taken while annotating **[1]**.

Inter-annotator Agreement

We pulled together all our individually labeled data and ran an inter-annotator agreement **[2]**. Our simple agreement measure was 0.71. We were satisfied with this inter-annotator agreement and proceeded to use Prodigy's review recipe to combine and consolidate our labels.

Our threshold for auto-accepting was 4, meaning any label with a 5 to 0 or 4 to 1 vote was considered majority and auto-accepted. We discussed in person any labels that were a 3-2 vote on a case-by-case basis. By doing this, we finalized our annotation guidelines and consolidated our 1000 annotations into 200 annotations (hmwk2-eval-final.jsonl).

Training Dataset

We each manually labeled 200 unique documents, for a total of about 1000 annotations. This was used as our training dataset (hmwk2-train-final.jsonl).

Methodology

We followed the Prodigy-Textcat-IAA slides and used those recipes with our labeled datasets. We trained models on hmwk2-train-final.jsonl (n=1018) and evaluated the models on hmwk2-eval-final.jsonl (n=199). The recipes used and the code run can be viewed on the notebook.

Three experiments were ran locally through Prodigy using an AMD Ryzen 3 7320U 8GB CPU:

- `experiment-1` is a model using spaCy defaults.
- `experiment-2` is a model using spaCy and the `en_core_web_md` base model.
- `experiment-3` is a model using the HF transformer `distilbert_base_uncased`.

Results & Analysis

Experiment	**Model**	**Best F1 Evaluation**	**Model Size (MB)**	**Time to Train***
1	prodigy train (spaCy defaults)	0.86	2.18 MB	< 1 min
2	spaCy + `en_core_web_md`	0.89	133 MB	> 10 min (Killed at ~100 epochs)
3	HF: `distilbert_base_uncased`	N/A	N/A	N/A

*AMD Ryzen 3 7320U 8GB CPU

Experiment 1 was the most seamless and quickest model to train **[3]**. 10/10

During experiment 2, when trying to `data-to-spacy` the training data and `en_core_web_lg` , the system ran out of memory and killed the process, so we opted for `en_core_web_md` instead, which completed successfully. Still, we had to kill the training for this experiment at around 100 epochs due to time constraints and not being willing to wait any longer for the model to fully train **[4]**.

During experiment 3, when trying to train using `distilbert_base_uncased`, the system ran out of memory and killed the process before even starting training **[5]**.

The experiments provided a good overview of the pros and cons of each model, its requirements, and resources required. It also showed that, in this case, larger models may not always provide proportionate gains when looking at Best F1 Evaluation.

For future text classification problems, especially multi-class ones, we will take into consideration all aspects of resource e.g., time, effort, hardware to estimate and better establish minimum requirements needed for successful experimentation and execution. We would look into using Google Colab as a means to address the lack of hardware power needed for larger models such as `distilbert_base_uncased` through Hugging Face.

Appendix

[1] Annotation Guidelines

Recipe: A recipe is a set of instructions or guidelines for preparing a particular dish or drink. It typically includes a list of ingredients along with the quantities needed, as well as detailed steps for cooking or assembling the dish. Recipes are commonly used in cooking and baking to ensure consistent results and to facilitate the replication of dishes by others. They may also include additional information such as cooking times, serving suggestions, and nutritional information.

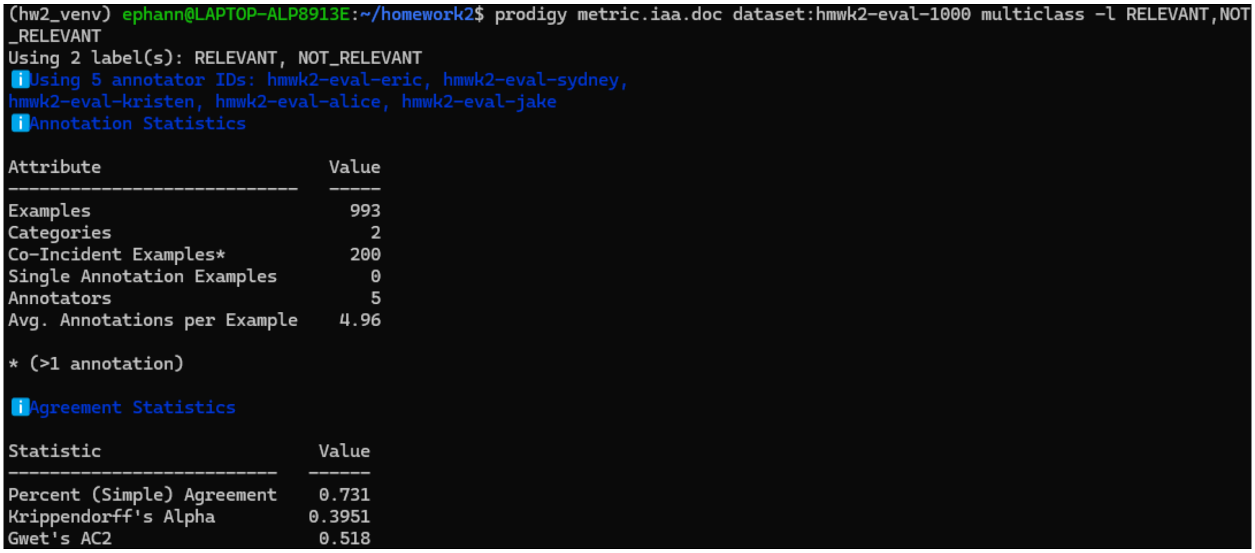
Relevant: ingredients, cooking, equipment, instructions. Even though it's only one word or two. Helpful for understanding the recipe. Supplementary instructions to a fixed recipe. Examples:

- "Replace honey with dijon mustard and you've got mine."
- "Brown the butter when making Rice Krispy Treats, trust me"
- "I wouldn't worry about it too much personally. If you are worried about it, the best option is to use an enameled Dutch oven."

Not Relevant: opinions/thoughts about a fixed recipe, but no new ingredients, equipment, or instructions that could help improve the recipe. Examples:

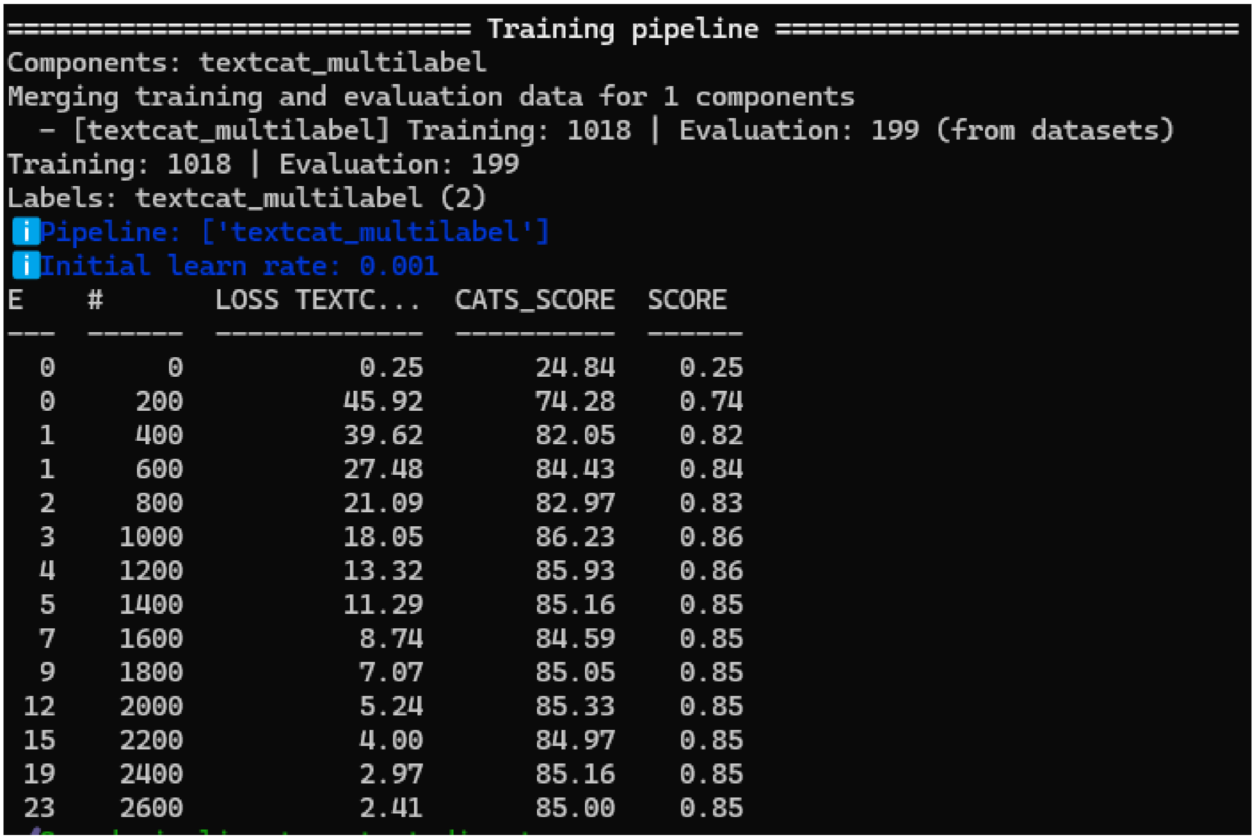
- "Juicier? Objectively yes. More flavorful? Absolutely not."
- "I've cooked fish, steak, burgers, hot dogs, bratwurst, veggies, just about anything on mine and I've never really had a problem with smoke. Doesn't make the room smokey or stink or anything."
- Shopping, prices, and bills for the ingredients/equipment.
- Links, but the text did not contain instructions/ingredients
  - Example: Julia Child: The French Chef - Your Own French Onion Soup https://www.youtube.com/watch?v=dw0Ij1Fxgq4

[2] Inter-annotator Agreement



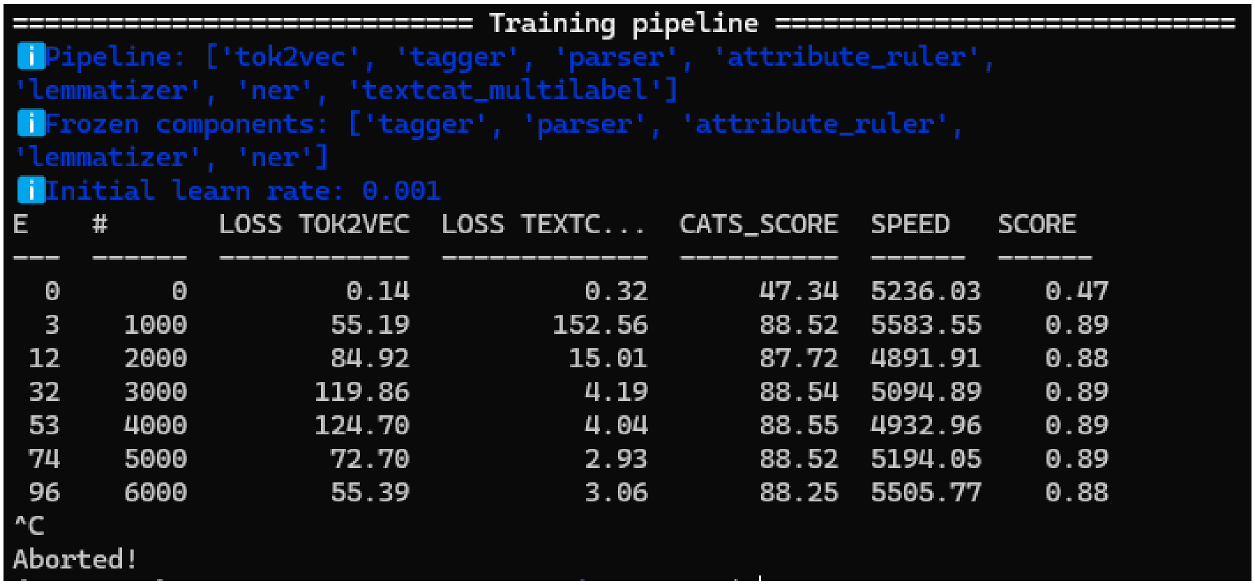
```
(hw2_venv) ephann@LAPTOP-ALP8913E:~/homework2$ prodigy metric.iaa.doc dataset:hmwk2-eval-1000 multiclass -l RELEVANT,NOT
_RELEVANT
Using 2 label(s): RELEVANT, NOT_RELEVANT
ℹ Using 5 annotator IDs: hmwk2-eval-eric, hmwk2-eval-sydney,
hmwk2-eval-kristen, hmwk2-eval-alice, hmwk2-eval-jake
ℹ Annotation Statistics

Attribute                      Value
----------------------------   -----
Examples                         993
Categories                         2
Co-Incident Examples*            200
Single Annotation Examples         0
Annotators                         5
Avg. Annotations per Example    4.96

* (>1 annotation)

ℹ Agreement Statistics

Statistic                    Value
--------------------------   ------
Percent (Simple) Agreement    0.731
Krippendorff's Alpha         0.3951
Gwet's AC2                    0.518
```

[3] Experiment 1



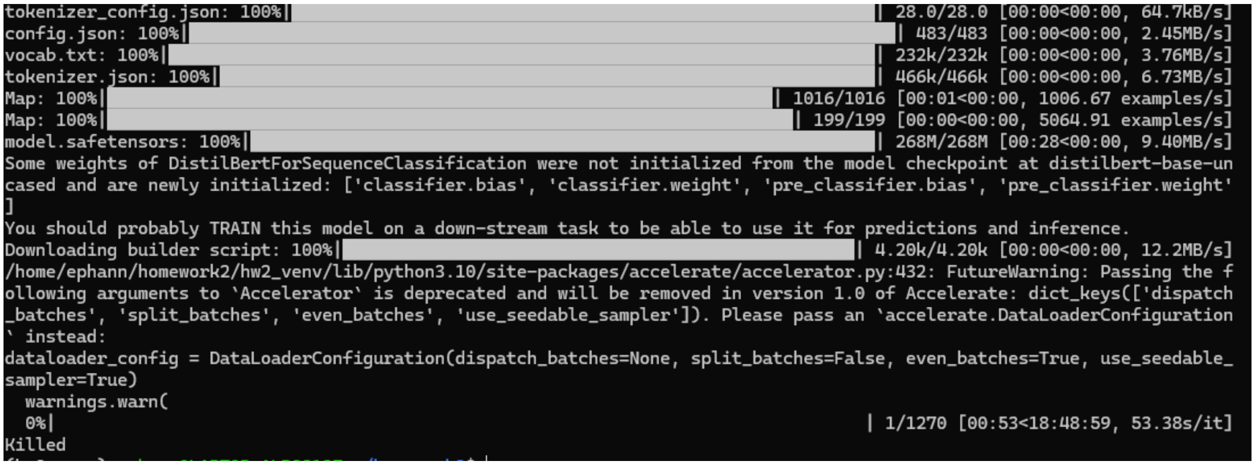
```
============================== Training pipeline ==============================
Components: textcat_multilabel
Merging training and evaluation data for 1 components
  - [textcat_multilabel] Training: 1018 | Evaluation: 199 (from datasets)
Training: 1018 | Evaluation: 199
Labels: textcat_multilabel (2)
ℹ Pipeline: ['textcat_multilabel']
ℹ Initial learn rate: 0.001
E    #       LOSS TEXTC...  CATS_SCORE  SCORE
---  ------  -------------  ----------  ------
  0       0           0.25       24.84    0.25
  0     200          45.92       74.28    0.74
  1     400          39.62       82.05    0.82
  1     600          27.48       84.43    0.84
  2     800          21.09       82.97    0.83
  3    1000          18.05       86.23    0.86
  4    1200          13.32       85.93    0.86
  5    1400          11.29       85.16    0.85
  7    1600           8.74       84.59    0.85
  9    1800           7.07       85.05    0.85
 12    2000           5.24       85.33    0.85
 15    2200           4.00       84.97    0.85
 19    2400           2.97       85.16    0.85
 23    2600           2.41       85.00    0.85
```

[4] Experiment 2



```
================================ Training pipeline ================================
ℹ Pipeline: ['tok2vec', 'tagger', 'parser', 'attribute_ruler',
'lemmatizer', 'ner', 'textcat_multilabel']
ℹ Frozen components: ['tagger', 'parser', 'attribute_ruler',
'lemmatizer', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS TEXTC...  CATS_SCORE  SPEED    SCORE
---  ------  ------------  -------------  ----------  -------  ------
  0       0          0.14           0.32       47.34  5236.03    0.47
  3    1000         55.19         152.56       88.52  5583.55    0.89
 12    2000         84.92          15.01       87.72  4891.91    0.88
 32    3000        119.86           4.19       88.54  5094.89    0.89
 53    4000        124.70           4.04       88.55  4932.96    0.89
 74    5000         72.70           2.93       88.52  5194.05    0.89
 96    6000         55.39           3.06       88.25  5505.77    0.88
^C
Aborted!
```

[5] Experiment 3



```
tokenizer_config.json: 100%|██████████████████████████████████████| 28.0/28.0 [00:00<00:00, 64.7kB/s]
config.json: 100%|███████████████████████████████████████████████████| 483/483 [00:00<00:00, 2.45MB/s]
vocab.txt: 100%|█████████████████████████████████████████████████████| 232k/232k [00:00<00:00, 3.76MB/s]
tokenizer.json: 100%|████████████████████████████████████████████████| 466k/466k [00:00<00:00, 6.73MB/s]
Map: 100%|████████████████████████████████████████████| 1016/1016 [00:01<00:00, 1006.67 examples/s]
Map: 100%|██████████████████████████████████████████████| 199/199 [00:00<00:00, 5064.91 examples/s]
model.safetensors: 100%|█████████████████████████████████████████| 268M/268M [00:28<00:00, 9.40MB/s]
Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-un
cased and are newly initialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias', 'pre_classifier.weight'
]
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Downloading builder script: 100%|████████████████████████████████████| 4.20k/4.20k [00:00<00:00, 12.2MB/s]
/home/ephann/homework2/hw2_venv/lib/python3.10/site-packages/accelerate/accelerator.py:432: FutureWarning: Passing the f
ollowing arguments to `Accelerator` is deprecated and will be removed in version 1.0 of Accelerate: dict_keys(['dispatch
_batches', 'split_batches', 'even_batches', 'use_seedable_sampler']). Please pass an `accelerate.DataLoaderConfiguration
` instead:
dataloader_config = DataLoaderConfiguration(dispatch_batches=None, split_batches=False, even_batches=True, use_seedable_
sampler=True)
  warnings.warn(
  0%|                                                                  | 1/1270 [00:53<18:48:59, 53.38s/it]
Killed
```