Training spaCy NER Models with Prodigy

EXPLOSION

INTRODUCTION

This handy flowchart contains our most common tips, tricks, and best practices for training and updating spaCy named entity recognition models with Prodigy. Read it to get started or refresh your memory!

Note: In the PDF version of this infographic, click any underlined text to go to our documentation.

RECIPE LEGEND

Prodigy recipes are Python functions that help you process annotations.

ner.correct — Stream in suggestions from the model and accept them, correct them or reject them.

ner.manual — Manually highlight spans of text and add optional pattern matches.

train — Train a model with annotations. Add the `--ner` flag for NER training.

train-curve — Train a model with different data portions (25%, 50%, 75%, 100%) to see how it improves.

review — Review annotations and resolve conflicts.

NODE LEGEND

- Flowchart start
- Favorable outcome: your project is ready to go
- Negative outcome: review, reconsider, and improve
- Our usual suggestion
- Decision node

METADATA

- NER Flowchart v2.0.0
- Release date (yyyy-mm-dd): 2022-10-05
- Prodigy version compatibility: v1.11.0 (2021-08-12) < v2.0 (Expected)

Flowchart

START

Is there an NER model available for your use-case?

- YES → Do you want to add new entity types to a trained model?
  - NO → Improve existing pre-trained model
  - YES → Do the texts of your new types overlap with the text of the pre-trained model's types? E.g.: 1) Adding MONTH to a model that already has DATE 2) Adding COMPANY to a model that already has ORG
    - NO → Do you want to add more than three entities?
      - NO → Add new entity types to a trained model
      - YES → Train a new model from scratch
    - YES → Try using Prodigy's Span Categorization component instead of NER (OR) **3**
- NO → Train a new model from scratch → Get more than 1000 sentences of raw unlabeled data

Improve existing pre-trained model → Get more than 1000 sentences of raw unlabeled data → **ner.correct** → **train** → Are you happy with the results yet?
- YES → **1**
- NO → Did you collect more than 2000 annotations?
  - NO → Collect more annotations
  - YES → **train-curve** → Does the accuracy in the last 25% go up?
    - YES → Is your data similar to what the original model was trained on? E.g., news, web text, etc.
      - YES → Try fine-tuning the hyperparameters (dropout, batch size) → Did it help?
        - YES → (back to train-curve)
        - NO → Try training a new model from scratch instead of modifying a pre-trained model
      - NO → Do your annotations at least appear inside text fragments or full texts?
        - YES → Try training a new model from scratch instead of modifying a pre-trained model
        - NO → Refrain from annotating lists of words or short phrases. Context is necessary for training NER models. Try changing your data and start again. **3**
    - NO → Is your data similar to what the original model was trained on?

Collect more annotations → (loop)

Can you write rules (regular expressions, Matcher patterns) that capture a large number of your entity types?
- NO → **ner.manual**
- YES → Do you get more than 90% accuracy with only rules on your evaluation data?
  - YES → Are you sure your evaluation is correct?
    - YES → **2**
    - NO → Fix your evaluation
  - NO → Write a patterns JSONL file ← Does your training data include examples of all entity types (existing and new)?
    - Data should include all entity types (prevent forgetting)

Does your training data include examples of all entity types (existing and new)?
- YES → Data should include all entity types (prevent forgetting) → Write a patterns JSONL file
- NO → Does the accuracy in the last 25% go up?
  - YES → Collect more annotations → **ner.correct** / **ner.manual** → **train**
  - NO → **train-curve**

Collect more annotations → **ner.correct** **ner.manual** → **train** → Are you happy with the results yet?
- YES → **1**
- NO → Did you collect more than 2000 annotations in total or more than 500 additional annotations?
  - YES → **train-curve**
  - NO → Are you happy with the results yet?

ner.manual

ner.manual → **train** → Are you happy with the results yet?
- YES → **1**
- NO → Did you collect more than 4000 annotations?
  - YES → **train-curve** → Does the accuracy in the last 25% go up?
    - YES → Are you annotating proper nouns & very short phrases?
      - YES → Try fine-tuning the hyperparameters (dropout, batch size) → Did it help?
        - YES → (back up)
        - NO → **4**
      - NO → Span categorization might be a better fit for your project → **3**
    - NO → Are your annotations internally consistent?
      - YES → Try fine-tuning the hyperparameters (dropout, batch size)
      - NO → **4**
  - NO → Collect more annotations → **review**

RESULTS

- **1** Export model
- **2** Stick with rules, probably not worth training a model
- **3** Probably not a good fit for classic NER task
- **4** Conduct error analysis to understand problems

prodigy