

# Training spaCy NER Models with Prodigy



## 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.

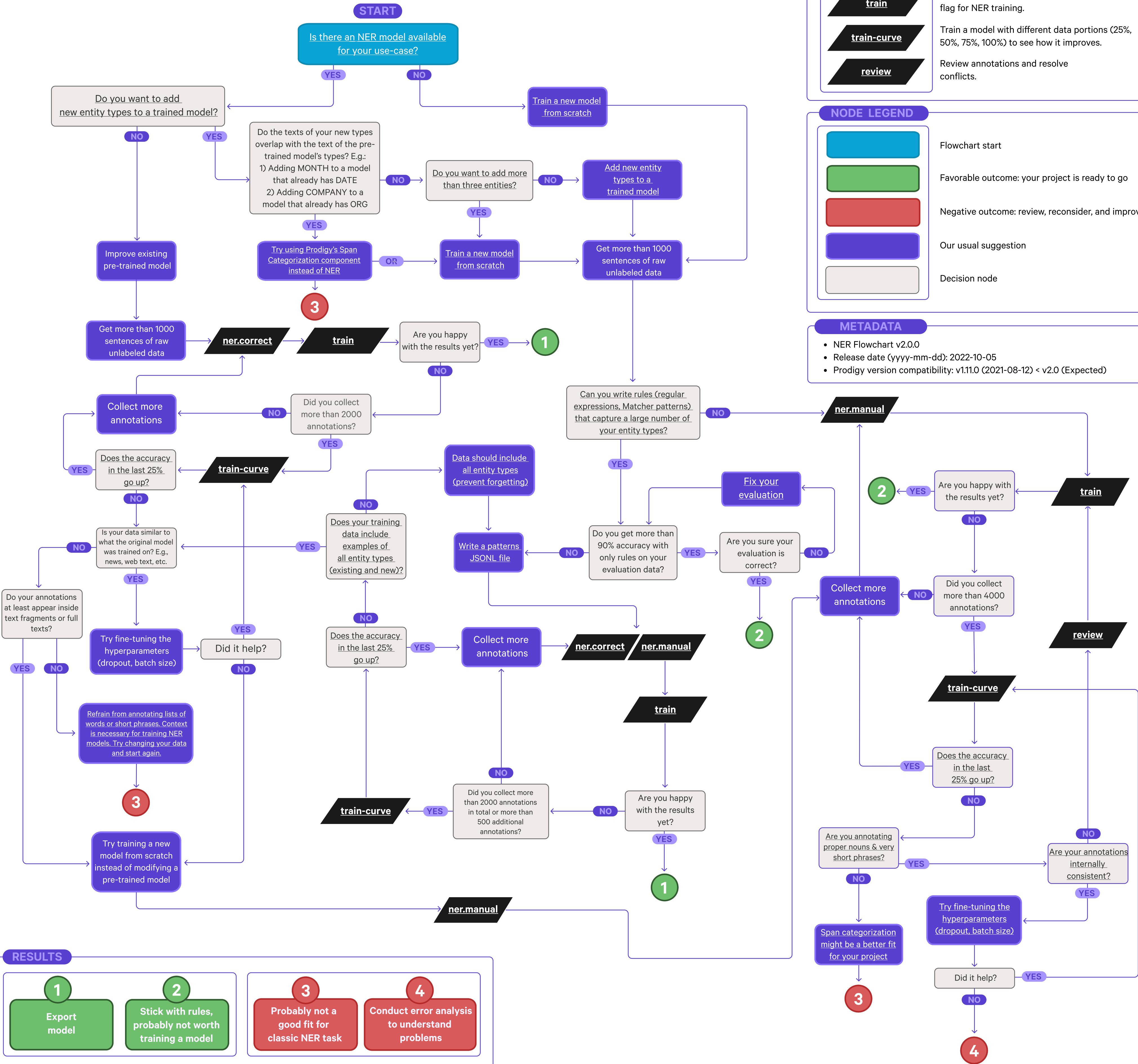
<b>ner.correct</b>	Stream in suggestions from the model and accept them, correct them or reject them.
<b>ner.manual</b>	Manually highlight spans of text and add optional pattern matches.
<b>train</b>	Train a model with annotations. Add the <code>--ner` flag for NER training.</code>
<b>train-curve</b>	Train a model with different data portions (25%, 50%, 75%, 100%) to see how it improves.
<b>review</b>	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)



## 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