Extracting NERs is useful for:

- Entities for medical notes (drugs, dosages, diseases, etc.)
- Entities for journal articles (author, topic, algorithm/method)
- Regex cannot distinguish Madoff as a person rather than a firm (is this really necessary for ML methods? Should we use regex/matching?)

It is difficult because:

- Research progress has been slow
- Many pre-trained NERs are old – language changes
- There exist very few tools (Prodigy being one of them, others?)
- It is knowledge intensive (you need to know the entities!) and need to train on at least 200 examples (manually tagged)
- Mix of easy and hard cases
- Hard to show your ideas are working (more work on challenging cases doesn't always result in much increase in performance whereas putting more care into other labeling may)

How Spacy Prodigy works:

- Start as a state machine with no output attached, all the words in the sentence ahead in the buffer, look at next word, have an action that starts an entity (begin move), fixes the label at the start of the entity (instead of end)
- Use BILUO tagging scheme
- O = outside of an entity
- Block out the actions and build into the framework
- Statistical model for predictions (how NN is structures). Framework:
    - Embed: Come up with the list of words
        - Norm, prefix, suffix, shape
        - No fixed vocabulary due to how they are stored in table (nothing is out of dictionary)
        - 128 vectors
    - Encode: Find them in context
        - Go from context-independent vectors to context-sensitive matrix
        - Uses Convolutional Neural Network, trigram CNN layer takes a window on either side of the word (relearn what this word means by its neighbors and output a new vector)
    - Attend: Summary vector
    - Predict:
        - MLP