Is supervised NER solved?

* Need to use state of the art model (Rob recommends mLUKE language model which masks spans instead of subwords + CRF layer (presented in previous lectures), but it depends on the data)
  + Check remaining errors qualitatively and quantitatively, to check which errors are still remaining and need to be solved
  + Needs large training data size
* Detailed analysis of remaining errors qualitative + quantitative
* Like the paper in the repo does with part of speech tagging, but for NER instead
* Basically, with the current levels of accuracy, what are still the open challenges with the field?

**Pros:**

* Primarily discussion and analysis
* No real answer, good argumentation can give good result

**Cons:**

* Have to implement SOTA, combining different algorithms and hoping it is good enough for our implementation to make sense
* Have to be really sure about arguments and analysis and discussion

Few-shot NER

* What can we do, if we don’t have a lot of training data? We can never obtain a lot of data for all language types. So how can we get a reasonable tagger, with few samples (<10)?
* “aims at training machine learning models with extremely limited available data”, “The need for a NER system able to be trained with few-annotated examples comes in all its urgency in domains where the annotation process requires time, knowledge and expertise (e.g., healthcare, finance, legal), and in low-resource languages”
* N-way K-shot: N entity types, K examples per entity type. That is all you provide to the model.
* Has become popular fast. With stronger models, few-shot becomes more feasible.
* For inspiration: see papers on Arxiv, but beware that they are non-reviewed (and therefore no quality check). May also have code-repositories
* Uses discriminative language models, but maybe we can use generative and prompting? Maybe compare…
* With only few examples, training data will be biased. For this reason, the few-NERD dataset divides into further sub-group entity types.
* The dataset few-NERD presents entity grouping and sub-grouping, where similarity of sub-groups is clearly seen.
* The usual analysis structure is also hierarchical – first find the entities, then label coarse grained, then label fine grained.
* A good overview: <https://dl.acm.org/doi/10.1145/3609483>
* Settings: (note that they use IO-labeling in the original paper, meaning you can’t always separate 2 annotations from each other)
  + Sup: supervised
  + Intra: use coarse labels to separate train-dev-test (have larger distinction between types of entities you see)
  + Inter: use fine labels to separate (then entities you get are a bit closer, but still not seen) ??
* Methods compared in the few-NERD paper:
  + protoBERT
  + NNShot and Structshot
* Ideas:
  + Cross-lingual few-shot span labelling (xSID)
  + From how many samples is standard supervised learning better?
  + In context learning versus previous few-shot methods

**Pros:**

* Rob seems to really like this topic

**Cons:**

* Difficult to implement, difficult to get something useful out of, just generally very high-level.
* Rob is not all that updated on the topic, and therefore can’t give in-depth help and guidance.

Unsupervised NER

* Instead of few-shot, 0-shot
* Remove the assumption of training data. No annotated training data
* Start with easy cases (actually relatively simple to identify some type of entities – capitalized often), once we have easy cases, we can start to train a model, get examples from raw data, add to training pool, bootstrap from there to build more complex.
* Capitalization or embeddings spaces can be used for example

**Pros:**

* Would be revolutionary if it works

**Cons:**

* Rob did not give a lot of information on it
* Highly language dependent – may be easier with languages like german where nouns are capitalized?
* Requires deep understanding of the code
* Different entities for different fields, how to create general rules for the model?