Active Learning Design Choices for NER with Transformers

Robert Vacareanu, Enrique Noriega-Atala, Gus Hahn-Powell, Marco A. Valenzuela-Escárcega, Mihai Surdeanu

Computational Language Understanding Lab at the University of Arizona

1. Introduction

Summary

We explore multiple important choices that have not been analyzed in conjunction regarding active learning for token classification using transformer networks. These choices are: (i) how to select what to annotate, (ii) decide whether to annotate entire sentences or smaller sentence fragments, (iii) how to train with incomplete annotations at token-level, and (iv) how to select the initial seed dataset. We explore whether annotating at sub-sentence level can translate to an improved downstream performance by considering two different sub-sentence annotation strategies: (i) entity-level, and (ii) token-level. These approaches result in some sentences being only partially annotated. To address this issue, we introduce and evaluate multiple strategies to deal with partially-annotated sentences during the training process. We show that annotating at the sub-sentence level achieves comparable or better performance than sentence-level annotations with a smaller number of annotated tokens. We then explore the extent to which the performance gap remains once accounting for the annotation time and found that both annotation schemes perform similarly.

Experimental Setup

- **Datasets:** CoNLL-2003, OntoNotes;
  1% of training data serves as validation
  25 Active Learning Iterations
- **Metrics:** F1
- **Model:** DistilBERT

Experiments

- **(1)** Which uncertainty function to use to select data for labeling?
- **(2)** Annotate at sentence level or at sub-sentence level?
- **(3)** How to select the initial dataset?

2. Experiment: Which uncertainty function to use?

Which Uncertainty Function To Use?

We compare between four functions.

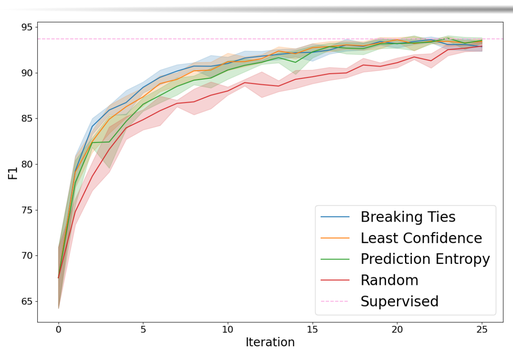
Breaking Ties: $\underset{x_i}{argmin}\left[P(y_i = l_1|x_i) - P(y_i = l_2|x_i)\right]$

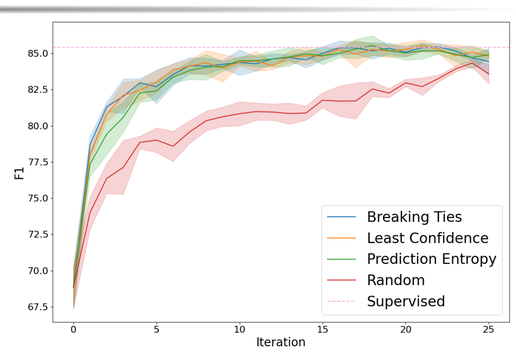
Least Confidence: $\underset{x_i}{argmax}\left[1 - P(y_i = l_1|x_i)\right]$

Prediction Entropy: $\underset{x_i}{armin}\left[\sum_{l=1}^{c} P(y_i = l|x_i) log P(y_i = l|x_i)\right]$

Random: Select random (uniformly) examples for annotation

Results





Conclusion

Breaking Ties performs best or comparable with the best.

3. Experiment: Annotate at sentence level or at sub-sentence level?

Training with Partial Annotations

We experiment with four ways to handle partially annotated sentences.

Mask Unknowns: Ignore unannotated tokens when computing the loss

Dropping All Unknowns: Drop the unannotated tokens

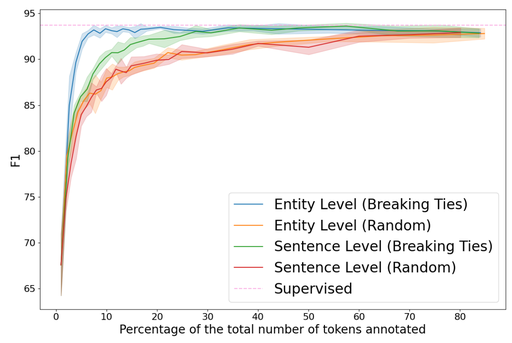
Masking unknown tokens that look like entities: Heuristically assign "O" to all the tokens that are unannotated and are not NNP; Mask all the others

Dropping all unknowns tokens that look like entities: Heuristically assign "O" to all the tokens that are unannotated and are not NNP; Drop all the others

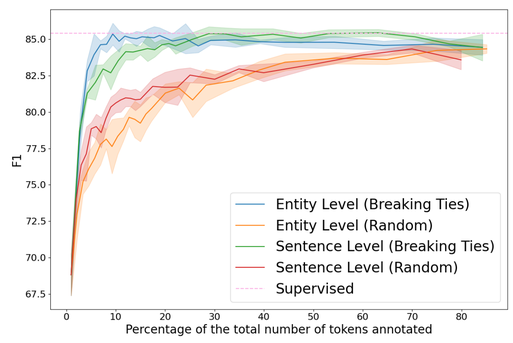
We found that **Mask Unknowns** performs the best (plots omitted due to space constraints; please see the paper)

Sentence vs Sub-Sentence

We compare between the performance of sentence level annotations vs sub-sentence level annotations, as a function of the total number of tokens annotated



(a) CoNLL-2003

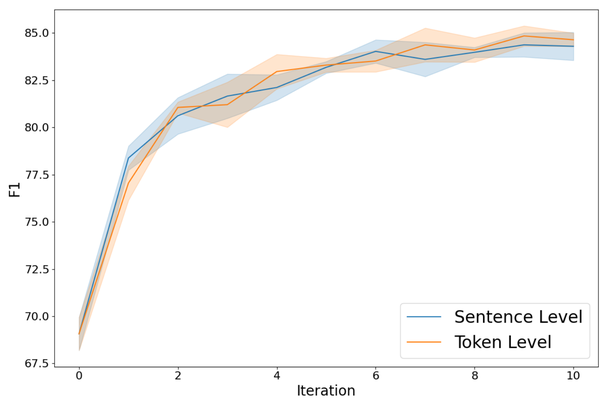


(b) OntoNotes

When viewing the performance as a function of the total number of tokens annotated, sub-sentence level annotations are (much) more efficient.

Feasability of Sub-Sentence Level Annotations

We compare the times between annotating the full sentence and annotating parts of the sentence, estimating: (i) time needed to annotate a sentence, and (ii) time needed to annotate a token



Conclusion

Both annotation schemes perform similarly once we account for annotation time, suggesting that even though in the sentence-level annotation setting the model receives annotations for more tokens including tokens it was not confused about, they are overall meaningfully contributing. We leave investigating better sub-sentence annotations to future work.

4. Experiment: Initial Dataset Selection

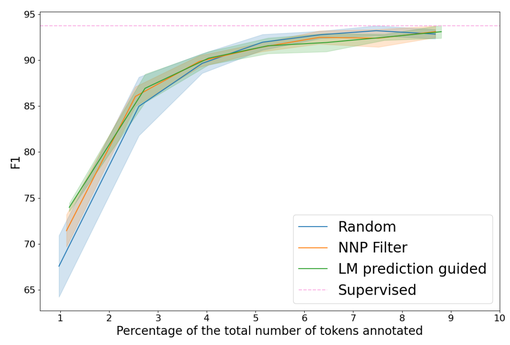
Initial Dataset Selection

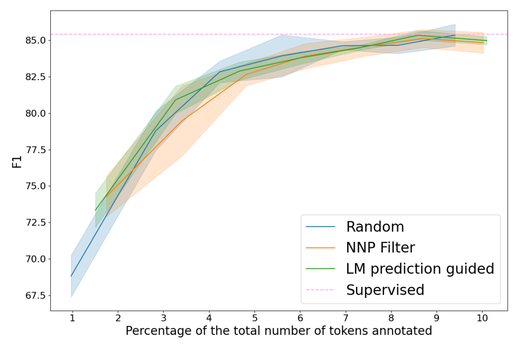
Can we do better than random selection? We investigate two selection procedures, in addition to the standard random selection.

Favor sentences with NNPs: because named entities tend to be NNPs

LM-guided selection: select sentences where it have the most difficulties in predicting the correct next token

Results





Conclusion

Using the language model to the selection improves the final performance

5. Conclusion

We investigated Active Learning Design Choices for NER with Transformers

- Which uncertainty query function to use? Breaking Ties performs well;
- Sub-sentence level annotations: Mask Unknowns works well; Once accounting for annotations time, both annotations level perform similarly;
- Selecting the initial dataset; Using the LM works well;
- Our code is available `https://github.com/clulab/releases/tree/master/lrec-coling2024-active-learning-design-choices`

Disclosure

This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under the ASKEM program and by the National Science Foundation (NSF) under grant #2006583. Mihai Surdeanu and Gus Hahn-Powell declare a financial interest in `lum.ai`. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.