Active Learning Design Choices for NER with Transformers

Robert Vacareanu[1,2], **Enrique Noriega-Atala**[1], **Gus Hahn-Powell**[1],
Marco A. Valenzuela-Escárcega[3], **Mihai Surdeanu**[1]
[1]University of Arizona, Tucson, AZ, USA
[2]Technical University of Cluj-Napoca, Cluj-Napoca, Romania
[3]Lex Machina
{rvacareanu, enoriega, hahnpowell, msurdeanu}@arizona.edu
mvalenzuela@lexmachina.com

Abstract

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.

Keywords: active learning, named entity recognition, transformers

1. Introduction

One approach to mitigate the time and cost requirements of data annotation for machine learning is active learning (Atlas et al., 1989; Balcan et al., 2006), where the machine learning algorithm is actively involved in deciding which examples are worth annotating. In the field of natural language processing (NLP), most research on active learning (Thompson et al., 1999; Zhang et al., 2022) has been concentrated on sequence classification (Tong and Koller, 2002; Zhang et al., 2016; Schröder et al., 2022), for tasks such as text classification (Zhang et al., 2015; Zhang and Plank, 2021), sentiment classification (Socher et al., 2013; Margatina et al., 2021), question answering (Fisch et al., 2019; Longpre et al., 2022), question classification (Li and Roth, 2002; Ein-Dor et al., 2020). In comparison, active learning (AL) for token classification (TC) has received little attention, and most of the previous work has concentrated either on pre-transformers methods with sub-sentence level annotations (Lowell et al., 2018; Radmard et al., 2021; Tsvigun et al., 2022) or on transformer-based models but with full-sentence annotations.

Exploring the use of transformers (Vaswani et al., 2017) for active learning is important for two reasons. First, it is unclear if transformers will exhibit the same behavior as other architectures in the active learning setting. As such, it is uncertain how much of the previous insights from other architectures are applicable to transformer networks, especially in the context of pretraining. Second, applying active learning to token classification tasks, as opposed to sequence classification tasks, is an understudied topic that presents unique challenges. For example, there are open questions on how to select what to annotate, annotating complete sentences or only a few tokens (see Figure 1), or how to select the initial dataset. In this work, we analyze these design choices with transformer-based backbones. We experiment on the named entity recognition (NER) task, as one of the most common token classification tasks, and two datasets. Figure 2 shows an example sentence for the NER task.

We make the following contributions: **(1)** We analyze which of the uncertainty-based query functions performs the best with transformer networks. We observe that *Breaking Ties* (Scheffer et al., 2001; Luo et al., 2004) performs either the best or comparably with the best; **(2)** We investigate three levels of annotation schemes: (i) complete sentences (sentence-level), (ii) entities (entity-level), (iii) individual tokens (token-level). We find that annotating at sub-sentence level (i.e., token-level or entity-level) achieves comparable or better performance with a lower number of tokens annotated, but this difference disappears once accounting for annotation time; **(3)** We explore 4 different approaches to enabling training with partially-annotated data. Masking the unannotated tokens during backpropagation performs the best; **(4)** We explore if the initial dataset can be selected in a better way than random sampling. Using the pre-trained language model for the initial selection performs better or comparable, at a reduced standard deviation.

2. Related Work

Active learning in NLP (and other fields) is a popular choice for reducing annotation burden (Zhang et al., 2022). In this work, we focus on active learning for token classification, which is relatively underexplored in comparison to active learning for sentence classification (Zhang et al., 2016; Schröder et al., 2022).

Our work is similar to that of Schröder et al. (2021), as we both investigate how the traditional uncertainty-based query strategies behave with the newer pre-trained transformer networks. In contrast, we work on active learning for the token classification task, something that has been under-explored. Additionally, we investigate querying smaller sentence fragments, the representation of partially-annotated data, and selecting an initial dataset beyond randomly sampling it.

Working with smaller sentence fragments for named entity recognition has been explored before (Jie et al., 2019; Mayhew et al., 2019; Strobl et al., 2022). For example, Mayhew et al. (2019) learns a classifier to reduce the weight of potential false negative `O` tags. Li et al. (2020) uses negative-sampling to keep the probability of training with unlabeled entities low. Effland and Collins (2021a) makes no assumptions about the unannotated tokens and treats them as latent variables. Different from these approaches, we work with partial annotations in the context of active learning.

In the context of active learning, most of the work focuses on using full sentence annotations (Sapci et al., 2021; Tsvigun et al., 2022; Nguyen et al., 2022; Moniz et al., 2022). Transformer-based networks are explored in Tsvigun et al. (2022), but only for sentence-level annotations. Additionally, they use a different model for acquisition than for training. Moniz et al. (2022) shows that doing active learning for multiple languages can improve the efficiency.

Similar to our approach, but for the part-of-speech tagging task, Chaudhary et al. (2020) explores token-level annotations. They use CVT (Clark et al., 2018) to handle unannotated tokens, which is more complicated than our proposed approach for training with partially-annotated data. Shelmanov et al. (2021) uses transformer-based networks for NER, but they work with fully-annotated sentences and analyze a different set of query strategies than us. Jafarpour et al. (2021) combines AL with curriculum learning for named entity recognition. Brantley et al. (2020) explores a heuristic that provides noisy guidance for annotations. They postpone querying the expert until a second classifier predicts that the expert is likely to disagree with the heuristic.

The work of Radmard et al. (2021) is close to our approach. However, they ignore transformers (Vaswani et al., 2017; Devlin et al., 2019) which induced a paradigm shift in the field and it is unclear if the same conclusions hold after using a pre-trained transformer-based network as the starting point. Their approach involves propagating the labels to the same spans in different sentences to serve as candidate labels. On the other hand, our proposed approach for handling unannotated tokens is much simpler: we ignore them when computing the loss.

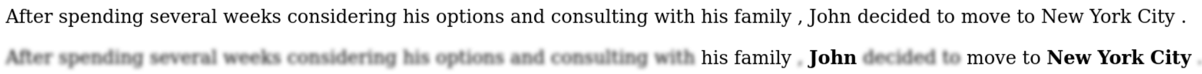
Positive-Unlabeled Learning (Peng et al., 2019; Zhou et al., 2022) is another relevant line of research, where, similar to our setting, there is a need to handle unannotated data. The main difference, however, is that we operate in the active learning framework, where the model is allowed to make queries. Additionally, we have no restrictions on which type of tokens are unannotated, allowing having annotations even for tokens that are labeled as `O`.

3. Method

We analyze the contribution of three different design choices that are critical for AL for TC: (i) how to select what to annotate, i.e., which query strategy is optimal for TC?, (ii) at what level should annotations be performed for TC, i.e., should one annotate complete sentences or sentence fragments?, and (iii) if one annotates fragments of sentences, how should these incomplete annotations be represented? We describe the various options in greater detail below.

3.1. Uncertainty-Based Query Strategies

We investigate four commonly used uncertainty-based query strategies. Note that all these strategies were designed for individual data points. However, for TC we must aggregate multiple data points (i.e., when full sentences are to be selected for annotation). To apply these query strategies to complete sentences, we aggregate the strategy scores of each token in the sentence to obtain an overall score for the entire sentence. The specific aggregation function used (either `min` or `max`) depends on the query strategy and are detailed below. To apply the query strategies at sub-sentence level, we use the scores of each individual token to determine which token should be selected for annotation. Because the selection is context-specific, once a token t from a particular sentence s is selected for annotation, that token is annotated only in sentence s. At selection time, for each query strategy, we sort the resulting list in ascending order and take



After spending several weeks considering his options and consulting with his family , John decided to move to New York City .

After spending several weeks considering his options and consulting with his family , **John** decided to move to **New York City** .

Figure 1: An example of a sentence with two named entities of interest: *John*, and *New York City*. The top text shows the complete text, while the bottom text shows just the named entities (in bold) and the local context sufficient for annotating the entity labels. The figure highlights that decisions for NER are driven mostly by local context, i.e., most of the sentence text can be ignored during AL annotations.

John	flew	to	New	York	to	watch	the	Super	Bowl	final	with	his	friends
B-PER	O	O	B-LOC	I-LOC	O	O	O	B-MISC	I-MISC	O	O	O	O

Figure 2: Example of a sentence (top) together with its corresponding annotations (bottom). We use the same labels and IOB annotation scheme as the CoNLL-2003 dataset. The sentence contains three named entities: *John*, labeled as PER, *New York*, labeled as LOC, and *Super Bowl*, labeled as MISC. Everything else is labeled as O. Each word within a named entity is further distinguished by whether it is the first word of the entity (B-) or not (I-).

the pre-defined number of elements to annotate. We detail each query strategy below.

Breaking Ties: Based on the model's predictions, we select token examples where the difference between the top two predictions is the smallest (Scheffer et al., 2001; Luo et al., 2004). At sentence-level, we consider the score of the token with the smallest difference to correspond to the score of the entire sentence (i.e., min); Formally, we select tokens x_i using:

$$\underset{x_i}{\operatorname{argmin}}\,[P(y_i = l_1|x_i) - P(y_i = l_2|x_i)]$$

where l_1 and l_2 is the most likely label and the second most likely label, respectively, according to the current model.

Least Confidence: Based on the model's predictions, we select tokens with the smallest probability for the most confident prediction (Culotta and McCallum, 2005). At sentence-level, we consider the score of the token with the smallest prediction confidence to correspond to the score of the entire sentence (i.e., min). Formally, we select tokens x_i based on:

$$\underset{x_i}{\operatorname{argmax}}\,[1 - P(y_i = l_1|x_i)]$$

where l_1 is the most likely label according to the model.

Prediction Entropy: Based on the model's predictions, we select tokens with the highest entropy for label probability distribution (Roy and McCallum, 2001). At sentence-level, we consider the score of the token with the highest prediction entropy to correspond to the score of the entire sentence (i.e., max).

$$\underset{x_i}{\operatorname{armin}}\left[\sum_{l=1}^{c} P(y_i = l|x_i) logP(y_i = l|x_i)\right]$$

Random: We select random (uniformly) examples for annotation, irrespective of the model's prediction. This query strategy does not need to aggregate any score. At the sentence-level, we simply select a given number of sentences and fully annotate them. At sub-sentence level we only annotate the selected tokens for the given sentences.

3.2. Annotation Level

We explore whether annotating at a lower granularity than sentence-level is (i) feasible from a learning perspective, and (ii) practical. We question the efficiency of sentence-level annotations for TC because they necessitate annotating a larger number of tokens, even those that the model is already confident about. For example, the class O is the most prevalent class in a typical named entity recognition (NER) dataset and there may be little benefit in extensively annotating such words. Figure 1 illustrates the motivation behind this intuition with a simple example.

To investigate this, we explore annotations at sub-sentence level and compare the corresponding models against models resulting from sentence-level annotations. Importantly, annotating at sub-sentence level means that we have sentences in the training data that are only partially annotated. To address this, we investigate different ways to train the model with partially annotated sentences. We detail them below.

3.2.1. Annotation Level

We describe below the three annotation levels we experimented with: (i) sentence-level, (ii) entity-level, and (iii) token-level.

Sentence-level: At this level we select *complete* sentences to annotate. Particular to the TC task is that we need to aggregate the score of each token to obtain a global score for the sentence. This global score is used to determine whether the sentence is selected for annotation or not. The specific aggregation function used (either min or max) depends on the query strategy. As discussed before, we use min for *Least Confidence*, and *Breaking Ties*, and max for *Prediction Entropy*, respectively. For example, when using *Least Confidence*, the score for a sentence is given by the score of the token with the lowest (i.e., min) *Least Confidence* value. We acknowledge that other aggregation strategies, such average, are possible. However, regardless of the aggregation strategy, the classifier receives annotations for both certain and uncertain tokens.

Sub-sentence-level: At this level we select and annotate *individual* tokens, rather than complete sentences. We investigate two distinct sub-sentence levels: (i) entity-level and (ii) token-level.

Entity-level: If the selected token is part of a named entity, we fully annotate that entity. Otherwise, we only annotate that particular token. For example, for the sentence John Doe flew to New York City to watch the Super Bowl final with his friends (see Figure 2), if the selected token is *York*, we will annotate the full entity *New York* as B-LOC I-LOC. The motivation behind this direction is that entity labels can often be determined using local context (Chieu and Ng, 2003; Agarwal et al., 2021), which should reduce the annotation effort (see Figure 1).

Token-level: With this strategy, we only annotate the individual tokens, regardless if they are part of an entity or not. For example, for the sentence above, if the selected token is *York*, we only annotate *York* as I-LOC, without annotating *New*. If the selected token is *flew*, we only annotate *flew* as O.

We acknowledge that the latter, token-level annotations are complex in practice, as annotating entity fragments may not be trivial. Consider, for example, the token *of*. It can be part of an entity, for example, *They work for Bank of America*, or it can be outside of an entity, for example, *They are coworkers of mine*. Nevertheless, we use it to investigate the limits of learning with sub-sentence annotations. Moreover, in practice, token-level annotations are complex because of different annotations schemes such as IOB vs. IOBES (Ramshaw and Marcus, 1995; Ratinov and Roth, 2009).

3.2.2. Data Representation

When we annotate at sub-sentence level, the dataset will contain partially-annotated sentences. For example, for the sentence in Figure 2, we might have the following tokens annotated: *{flew, to, New, York}* with the following annotations: {O, O, B-LOC, I-LOC}. In order to train the model with this type of partially-annotated data, we examine the following strategies:

(i) Masking all unknowns: We feed the model the full sentence, regardless if it is fully annotated or not. Then, we calculate the loss using only the annotated tokens, ignoring (i.e., *masking*) the predictions for the unannotated tokens. For the sentence in Figure 2 and given the aforementioned annotated tokens, the loss is calculated using only the predictions for the tokens *flew*, *to*, *New*, and *York*. Nevertheless, the representation of annotated tokens is influenced by the unannotated tokens due to the self-attention mechanism in transformers.[1]

(ii) Dropping all unknowns: Before feeding the sentence to the model, we drop the unannotated tokens. This process may result in an ungrammatical sentence. For example, for the sentence in Figure 2, we will feed only *flew to New York* to the model. The training then proceeds as usual, since all the tokens in the sentence are annotated.

(iii) Masking unknown tokens that look like entities: In this strategy we use a heuristic commonly employed for NER tasks: the part-of-speech (POS) tag of named entities constituents tend to be NNP. Based on this observation, we assign a label of O to every unannotated token that is not an NNP. Then we use the same strategy as in (i). That is, we feed the model the full sentence and we only calculate the loss using the tokens that were either gold annotated or annotated according to our heuristic. For the sentence in Figure 2 with the aforementioned annotations, we assign a label of O to the following unannotated tokens {*to, watch, the, final, with, his, friends*} because their POS tag is not NNP. The tokens {*John, Super, Bowl*} are left unannotated because their POS tag is NNP. The training then proceeds as in (i), ignoring only the tokens {*John, Super, Bowl*}.

(iv) Dropping all unknowns tokens that look like entities: We employ the same heuristic as in (iii) and we combine it with the dropping all unknowns strategy. More concretely, we assign a label of O to every unannotated token that is not an NNP. Then we use the same strategy as in (ii). For example, for the sentence in Figure 2 and given the aforementioned annotated tokens, we assign a label of O to the following unannotated tokens {*to, watch, the, final, with, his, friends*} because their POS tag is not NNP. The tokens {*John, Super, Bowl*} are unannotated, but their POS tag is NNP, so we drop them, as in (ii). The final sentence that will be fed to the model is *flew to New York to watch the final with his friends*.

We note that although *masking* and *dropping* are conceptually similar, the key difference is that *masking* utilizes all tokens to compute the final representations, while *dropping* does not see unannotated tokens in the text provided to the model.

[1] This also holds true for LSTMs or CNNs, as they also aggregate global context into local representations.

3.3. Initial Dataset Selection

Traditionally, the initial (or seed) dataset is randomly sampled from the dataset. We examine the possibility of improving upon this selection process by prioritizing sentences likely to contain more useful signal, e.g., more named entities in the NER context. The intuition is that it should be beneficial to the model to see more examples of named entities rather than more examples of tokens annotated with `O`. We investigate two novel approaches, that we describe below:

NNP-guided random sampling: In this approach, we employ random sampling but restrict it to sentences with a higher count of proper nouns (NNP part-of-speech tags) Sentences containing fewer than a threshold T number of words tagged as NNP are filtered out. Then, we continue with random sampling over the remaining set. For each sentence s, we use an adaptive threshold that depends on the number of words logarithmically $count\_nnp(s) \geq \alpha \cdot log(len(s))$. The intuition here is that expecting the number of words in a sentence that are tagged as NNP to grow linearly with the sentence length is too strict.

Language model-guided selection: In this approach we explore the possibility of using the pre-trained language model's (LM) already-acquired knowledge to select the most challenging sentences. This is achieved by leveraging the capabilities of the pre-trained LMs to select the examples where the model has the most difficulties in predicting the correct token. Formally, we measure this difficulty by looking at the difference between the two most likely predictions, similar to *Breaking Ties*. A small difference indicates that the model has difficulties in predicting the real token. We hypothesize that sentences with such tokens are more informative for the model because there is no sufficient background knowledge accumulated in the LM to be used for the downstream task.

Random: We randomly select data points from the dataset. We include this approach to serve as a baseline, as it is the most common strategy in the literature (Radmard et al., 2021).

4. Experiments

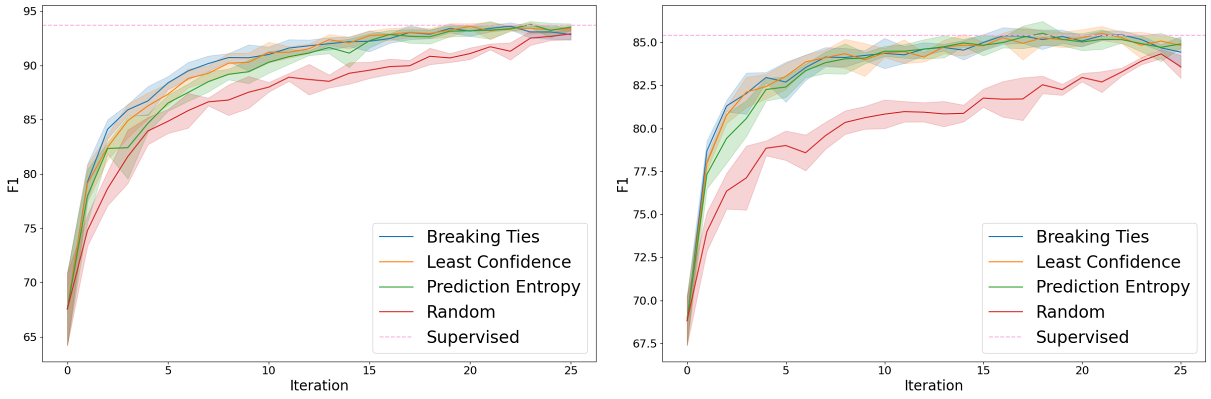
4.1. Experiment Setup

We experiment with two widely used named entity recognition datasets: CoNLL-2003 (Sang and Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2012). We use the same learning rate and weight decay for all our experiments. We use early stopping and choose the best-performing model on the validation partition. Similar to (Radmard et al., 2021), we sample 1% of the training data to serve as validation and use this for the early stopping procedure. The rationale behind using an alternative, small validation partition than the one available with the datasets is that those partitions are usually much larger, making the active learning experiment unrealistic. We then show our exploratory results on the complete development partition of each dataset (i.e., not the randomly sampled 1%), then use these insights and run the best settings on the test partition. Due to computation budgets, we train for only 25 active learning iterations. We begin each experiment with 1% of the total sentences fully annotated, regardless of the annotation level used. The initial dataset is selected through random sampling, unless otherwise specified. We gradually increase the number of examples we annotate at each step to ensure a larger coverage of the full dataset as we progress towards the end of the active learning process. By the final active learning iteration, approximately 80% of the data is annotated. Further details can be found in the supplementary material.

For all experiments, we use the F1 score, as it is standard for the datasets (Sang and Meulder, 2003; Pradhan et al., 2012; Radmard et al., 2021) and include the performance in the fully-supervised case, as a baseline. The F1 is calculated according to the official `conlleval` script (Nakayama, 2018). All experiments were repeated with five random seeds. All experiments follow the same set-up, unless otherwise specified. We use early stopping with a patience of 3 epochs on the F1 score. More details can be found in the supplementary material. For completeness, we include the performance on the test partition in the supplementary material using the insights from our exploration.

4.2. Datasets

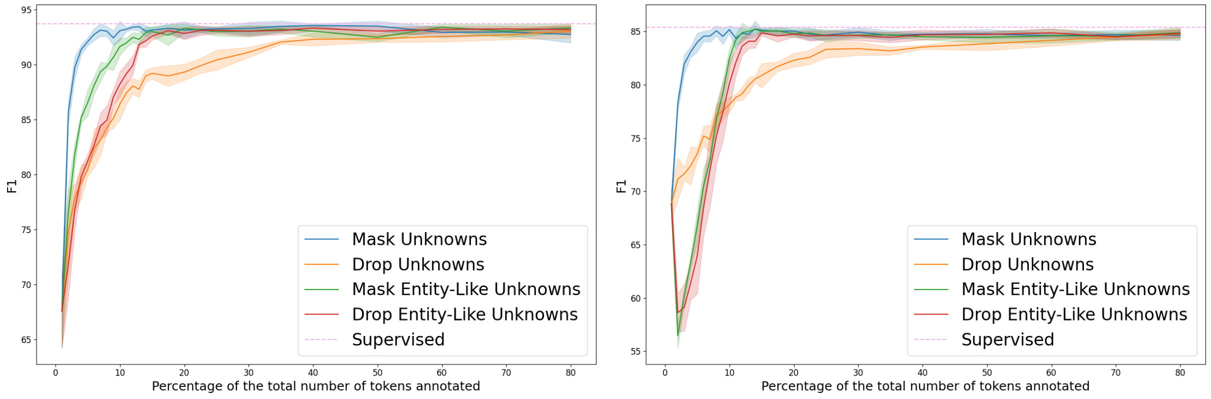
We use CoNLL-2003 (Sang and Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2012). We use the English data from both datasets. The datasets are annotated with the `IOB` annotation scheme (Ramshaw and Marcus, 1995), which means that it differentiates between a token at the beginning of a named entity (`B-`) and a token in the middle of a named entity (`I-`). CoNLL-2003 data is sourced from the Reuters Corpus, while OntoNotes 5.0 contains data from multiple genres such as news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, talk shows. Both datasets come with part-of-speech tag annotations. The named entities annotated in the datasets differ: CoNLL-2003 uses four named entities classes, while OntoNotes 5.0 contains 18 (see Supplementary material).



(a) CoNLL-2003

(b) OntoNotes

Figure 3: F1 Score on the validation partition for the datasets considered, as a function of the active learning iteration. The annotations are at the sentence-level, meaning that all query strategies will have the same number of sentences for training. Breaking Ties consistently performs the best or comparably to the best. Additionally, all query strategies tend to perform similarly towards the end because approximately 80% of the data is annotated at this point. Lastly, all 3 uncertainty-based queries consistently outperform the random query baseline up until saturation.



(a) CoNLL-2003

(b) OntoNotes

Figure 4: F1 Score on the validation partition for the datasets considered, as a function of the percentage of the tokens annotated. We query at token-level and compare between 4 different ways to enable training with partially-annotated sentences. The strategy *Mask unknowns* performs overall the best in all the cases considered.

4.3. Query Strategy

In our analysis, we first evaluate the performance of each query strategy presented in Section 3.1 using sentence-level annotations.

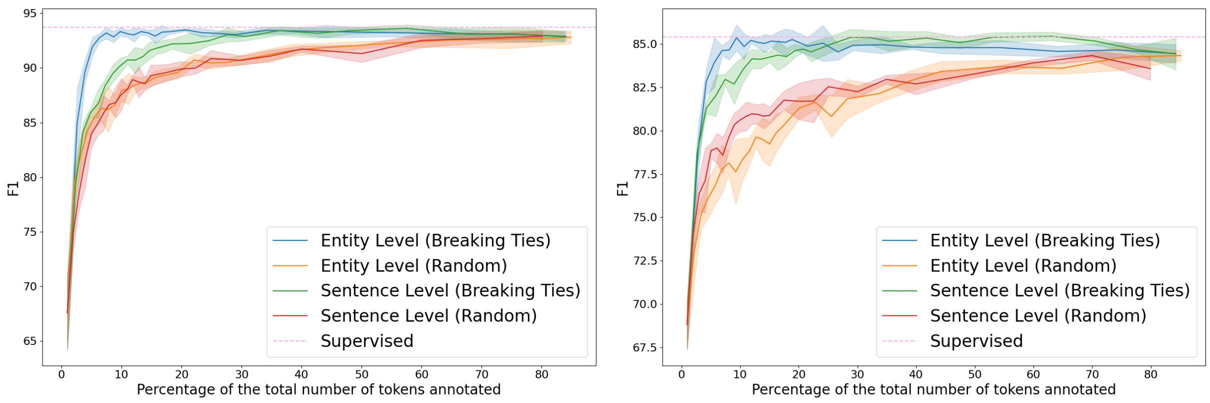
We present our results in Figure 3 and draw the following observations. First, Breaking Ties performs better or at least as well as the other uncertainty queries, despite its simplicity. This is most notable when little training data is used, which is, arguably, the most realistic AL scenario. This aligns with the observation of Schröder et al. (2022) for sentence classification using Transformers. We note, however, that *Least Confidence* is a strong contender as well. Second, all query strategies perform similarly towards the end of the active learning loop, as expected given that approximately 80% of the data is annotated at this point. Third, the three uncertainty-based queries consistently outperform the random query baseline up until saturation.

4.4. Annotation Level

Following the observation in Section 4.3, we use *Breaking Ties* as our default query strategy.

4.4.1. Data Representation

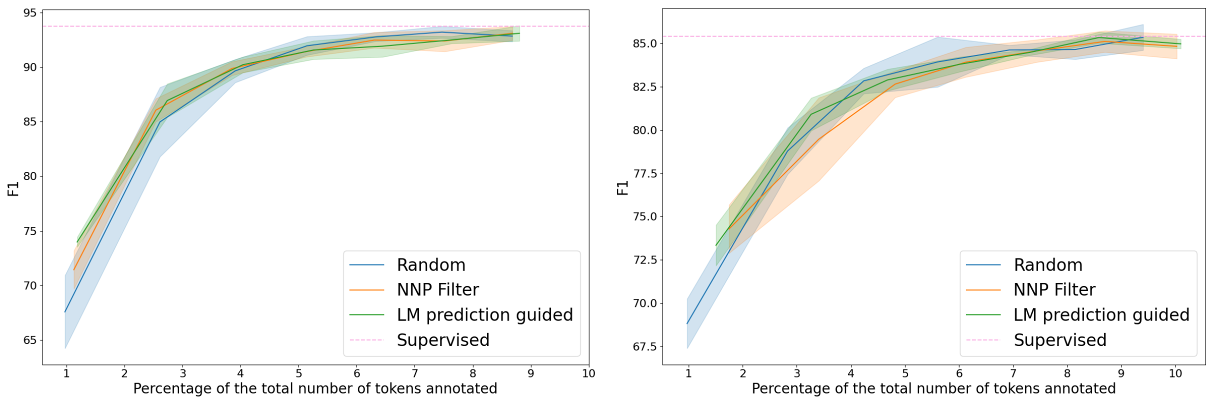
To investigate the feasibility of partial annotations, we first explore different data representation strategies to enable training with a partially-annotated dataset, as described in Section 3.2.2. We query



(a) CoNLL-2003

(b) OntoNotes

Figure 5: F1 Score on the validation partition for the datasets considered. We compare between annotating at entity-level and annotating at sentence level. Annotating at entity-level obtains similar (or better) performance, but at a greatly reduced cost.

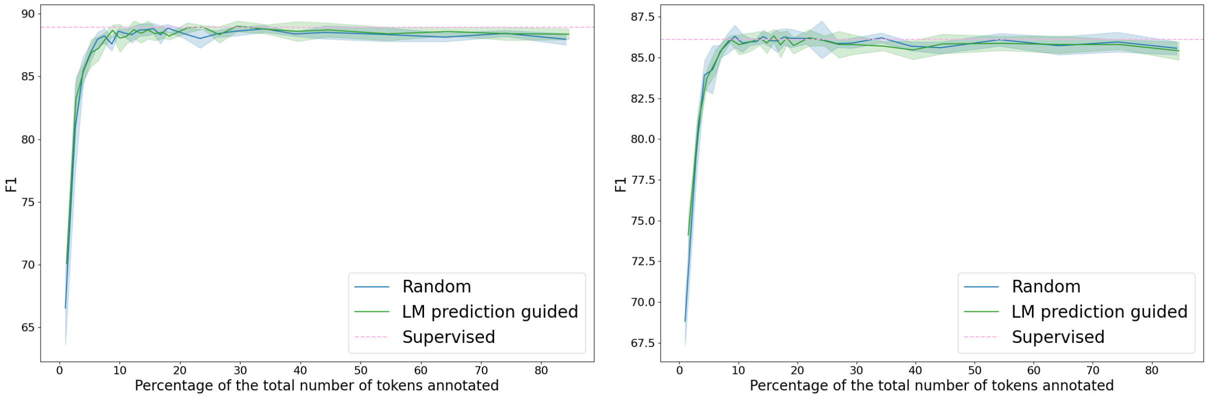


(a) CoNLL-2003

(b) OntoNotes

Figure 6: F1 Score on the validation partition for the datasets considered, as a function of the percentage of the tokens annotated. We query at entity-level with Breaking Ties and compare against 3 different ways of selecting the initial dataset. We remark that using more informed dataset selection strategies offer a boost in performance in the early stage of the active learning loop, boost that decreases as the number of tokens selected increases.

at the token-level using the *Breaking Ties* query strategy. The F1 performance of these strategies is shown in Figure 4 as a function of the percentage of tokens annotated. We can see that the *Mask Unknowns* data representation strategy, despite its simplicity, performs the best. We suspect this is caused by the attention mechanism in transformers: being exposed to tokens even when they are not labeled is helpful, as they are used to construct the representation of the tokens that are annotated. Second, we can see that querying at sub-sentence level achieves a high performance early, saturating at around 15% of the total number of tokens. Third, we observe that dropping the unknowns performs the worst until a large number of tokens have been annotated. We suspect that this is because, in the beginning, the tokens selected for annotation are spread across a large number of sentences. Therefore, in the *Drop unknowns* strategy, the model will be exposed to a large number of short, even ungrammatical sentences which is detrimental. In (Effland and Collins, 2021b) the authors argue that backpropagating only through labeled (and non-O) tokens is detrimental to the model's performance, as it will not learn meaningful representation to predict the O tag. In contrast, we find empirically that using an initial random sample of fully annotated sentences and then only sub-sentence level annotations gives enough initial signal to learn meaningful representations to predict the O label, even though the model predominantly selects non-O tokens for annotations.



(a) CoNLL-2003

(b) OntoNotes

Figure 7: F1 Score on the test partition for the datasets considered, as a function of the percentage of the tokens annotated. We show the mean and standard deviation of 5 runs.

4.4.2. Annotation Efficiency

We compare the performance of the Breaking Ties query strategy at the sentence-level and at the sub-sentence level. Following the observations in Section 4.4.1, for the sub-sentence level, we use the *Mask Unknowns* strategy to enable training with partially-annotated data. We include the performance with a *Random* query strategy to show that both (i) a finer-grained annotation level, and (ii) an informative query strategy are important in order to obtain similar (or better) performance with the same number of annotated tokens.

We show the F1 performance as a function of the percentage of tokens annotated in Figure 5. We remark that querying at sub-sentence level achieves comparable (or better) performance as querying at sentence-level, and it does so at a much faster rate. Entity-level annotations achieves its top performance at around 10% of the training data, while sentence-level annotations does so at around 30%. Lastly, we note that when a non-informative query strategy is used (i.e., *Random*), the performance at entity-level is similar with the performance at the sentence level. This indicates that the gain in performance at entity level comes from having labels for the informative tokens.

For example, for both CoNLL-2003 and OntoNotes the percentage of tokens annotated with O is over 83%. In comparison, the number of entities together with a window of 2 tokens (which is an estimate of the local context necessary to annotate the NER labels) constitutes only 33% of the dataset for CoNLL-2003 and 18% for OntoNotes, respectively.

4.5. Initial Dataset Selection

In the following, we investigate whether it is possible to improve upon the traditional method of randomly sampling the initial dataset. We explore two new strategies: (i) *NNP Filter*, which is a baseline that randomly selects sentences after filtering them based on the number of NNPs (based on the heuristic that sentences containing more proper nouns are more informative for NER), and (ii) *LM prediction guided*, which uses the underlying pre-trained LM to select sentences based on its word-prediction scores.

We show the F1 performance as a function of the percentage of tokens annotated in Figure 6. We remark that both *NNP Filter* and *LM prediction guided* outperform the classical random dataset selection algorithm until approximately 5% of the total number of tokens. As the number of tokens annotated increases, the difference decreases. This is expected because *Breaking Ties* is able to identify and select relevant tokens for annotations, therefore the ratio of relevant tokens increases. As it reaches and surpasses 5% (i.e., ~ 1% selected initially, ~ 4% during the active learning iteration loop), the advantages of a more informed initial dataset diminish. Nevertheless, we remark that the best performance achieved with a more informed dataset selection strategy is higher than random selection, although the difference is small. More importantly, the standard deviation of the model's F1 score across 5 runs for the more informed selection strategies is at less than half compared to the random selection strategy, suggesting that a better selection strategy can make the training more stable. We find the LM-guided selection results exciting considering that this strategy is agnostic to the actual sequence modeling task.

4.6. Test Performance

Following the observations regarding various design choices for Active Learning (AL) for Token Classification (TC) from the previous sections, we apply our resulting model on test. To facilitate comparison with the classical methods, we include the performance with the sentence-level annotations as well as the performance of the model in the fully-supervised case. We show our results in Figure 7. We remark that our proposed method obtains a similar or better performance than the fully supervised model at around 10%, after only 6 active learning annotation rounds. We acknowledge the difference in performance between our model and the state-of-the-art. The difference in performance comes from different model choices. Due to computation budgets, we use DistilBERT, which is much lighter than the ones used for the state-of-the-art. For example, DistilBERT-base has $65M$ parameters, while BERT-base has $107M$ and BERT-large has $345M$. Furthermore, our goal was not to obtain the best performance, as that would have required a large computational budget, but instead to perform an exploration of how transformer-based networks behave in the context of active learning for sequence tagging at sentence and sub-sentence level.

4.7. Feasibility of Token-Level Annotations

While § 4.4.1 indicates that for the same number of annotated tokens, partial annotations perform better, that experiment ignored the time cost of annotating these tokens in the different scenarios. In this section, we delve into the practicality of token-level annotations, focusing on the annotation workload aspect. Specifically, we investigate whether opting for token-level annotations, as opposed to sentence-level annotations, leads to enhanced model performance while considering the *time investment required for annotation*. To this end, we compare: (a) a traditional NER annotation task for sentence-level annotations, and (b) a token-level annotation task, in which we leverage the model's predictions and ask the annotator to decide which (if any) of the top 5 NE predictions for the given token is correct. Based on this experiment on OntoNotes, we approximate the annotation time for a random sentence to $\sim 27.5s$ and for a random token to $\sim 3.2s$. Based on this, at each active learning iteration, we select data amounting to 6h of annotation effort. We show our results in Figure 8. Importantly, the X-axis in this figure uses the *same annotation time per iteration* rather than the number of tokens annotated. Overall, 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.

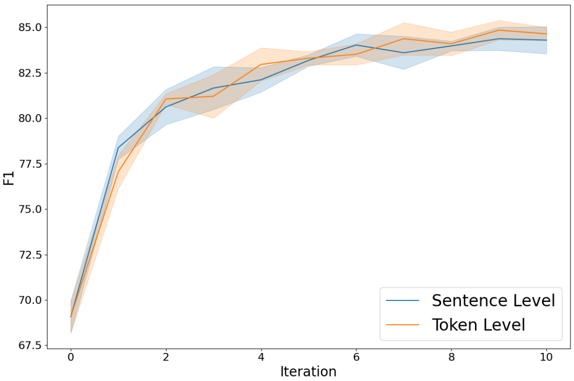


Figure 8: Comparing the F1 score on OntoNotes between sentence- and token-level annotations once accounting for annotation time.

5. Conclusion

We analyzed multiple important choices that have not been analyzed jointly for active learning for token classification using transformer networks. We investigated the following choices: (i) how to select what to annotate, (ii) decide whether to annotate complete sentences or smaller sentence fragments, (iii) if we annotate smaller sentence fragments, how to train with incomplete sentence annotations, and (iv) how to select the initial dataset, beyond random sampling. Our experiments showed that: (i) Breaking Ties performs better than other methods, (ii) annotating smaller sentence fragments can achieve similar (or better) performance as annotating the full sentence for a similar number of tokens annotated, but this difference advantage vanishes once accounting for the annotation time, (iii) in order to enable training with incomplete annotations, masking the tokens with unknown annotation when computing the loss performed the best out of the strategies analyzed, and (iv) using the pre-trained language model for the initial dataset selection can increase the performance when little data is annotated. Our code is publicly available.[2]

6. Acknowledgements

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.

[2] https://github.com/clulab/releases/tree/master/lrec-coling2024-active-learning-design-choices

7. Limitations

Despite being evaluated on multiple datasets and multiple genres, our work focuses on one language, English, and one task, named entity recognition. In particular, it is unclear if the effectiveness of *Mask Unknown* for training with partially-annotated data translates to other languages and how dependent it is on the perplexity of the underlying language model. Future work will complement our exploration to other languages and other tasks.

Our proposed method's scalability to long text depends on the underlying model's scalability properties. In this paper, we used transformers with quadratic attention which scales poorly to longer texts. However, typically, named entity recognition is performed at the sentence level.

8. Ethical Considerations

We use pre-trained language models, therefore this work shares many of the same ethical issues such as social biases or perpetuating stereotypes (Weidinger et al., 2021). In this work we did not pretrain any new language model. We do not envision any additional negative societal impact resulting from this work.

9. Bibliographical References

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2021. Interpretability analysis for named entity recognition to understand system predictions and how they can improve. *Computational Linguistics*, 47(1):117–140.

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Les E. Atlas, David A. Cohn, and Richard E. Ladner. 1989. Training connectionist networks with queries and selective sampling. In *NIPS*.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. 2006. Agnostic active learning. *Proceedings of the 23rd international conference on Machine learning*.

Su Lin Blodgett, Solon Barocas, Hal Daum'e, and Hanna M. Wallach. 2020. Language (technology) is power: A critical survey of "bias" in nlp. In *Annual Meeting of the Association for Computational Linguistics*.

Kianté Brantley, Amr Sharaf, and Hal Daum'e. 2020. Active imitation learning with noisy guidance. In *Annual Meeting of the Association for Computational Linguistics*.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Aditi Chaudhary, Antonios Anastasopoulos, Zaid A. W. Sheikh, and Graham Neubig. 2020. Reducing confusion in active learning for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 9:1–16.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

Hai Leong Chieu and Hwee Tou Ng. 2003. Named entity recognition with a maximum entropy approach. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 160–163.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Conference on Empirical Methods in Natural Language Processing*.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Thomas Effland and Michael Collins. 2021a. Partially supervised named entity recognition via the expected entity ratio loss. *Transactions of the Association for Computational Linguistics*, 9:1320–1335.

Thomas Effland and Michael Collins. 2021b. Partially supervised named entity recognition via the expected entity ratio loss. *Transactions of the Association for Computational Linguistics*, 9:1320–1335.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active learning for bert: An empirical study. In *Conference on Empirical Methods in Natural Language Processing*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *ArXiv*, abs/1910.09753.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active curriculum learning. *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *North American Chapter of the Association for Computational Linguistics*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *International Conference on Computational Linguistics*.

Yangming Li, Lemao Liu, and Shuming Shi. 2020. Empirical analysis of unlabeled entity problem in named entity recognition. *ArXiv*, abs/2012.05426.

S. Longpre, Julia Reisler, Edward Greg Huang, Yi Lu, Andrew J. Frank, Nikhil Ramesh, and Chris DuBois. 2022. Active learning over multiple domains in natural language tasks. *ArXiv*, abs/2202.00254.

David Lowell, Zachary Chase Lipton, and Byron C. Wallace. 2018. Practical obstacles to deploying active learning. In *Conference on Empirical Methods in Natural Language Processing*.

Tong Luo, Kurt Kramer, Dmitry B. Goldgof, Lawrence O. Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. 2004. Active learning to recognize multiple types of plankton. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 3:478–481 Vol.3.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Conference on Empirical Methods in Natural Language Processing*.

Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. Named entity recognition with partially annotated training data. In *CoNLL*.

Joel Ruben Antony Moniz, Barun Patra, and Matthew R. Gormley. 2022. On efficiently acquiring annotations for multilingual models. In *ACL*.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Minh Le Nguyen, Nghia Trung Ngo, Bonan Min, and Thien Huu Nguyen. 2022. Famie: A fast active learning framework for multilingual information extraction. *ArXiv*, abs/2202.08316.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419, Florence, Italy. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *EMNLP-CoNLL Shared Task*.

Puria Radmard, Yassir Fathullah, and Aldo Lipani. 2021. Subsequence based deep active learning for named entity recognition. In *ACL*.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *ArXiv*, cmp-lg/9505040.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Conference on Computational Natural Language Learning*.

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *ArXiv*, cs.CL/0306050.

Ali Osman Berk Sapci, Öznur Taştan, and Reyyan Yeniterzi. 2021. Focusing on possible named entities in active named entity label acquisition. *ArXiv*, abs/2111.03837.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2021. Revisiting uncertainty-based query strategies for active learning with transformers. *ArXiv*, abs/2107.05687.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis I. Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, E. Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. Active learning for sequence tagging with deep pre-trained models and bayesian uncertainty estimates. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*.

Michael Strobl, Amine Trabelsi, and Osmar R Zaiane. 2022. Named entity recognition for partially annotated datasets. In *NLDB*.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *International Conference on Machine Learning*.

Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66.

Akim Tsvigun, Artem Shelmanov, Gleb A. Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian, and Leonid Zhukov. 2022. Towards computationally feasible deep active learning. *ArXiv*, abs/2205.03598.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Laura Weidinger, John F. J. Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zachary Kenton, Sande Minnich Brown, William T. Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *ArXiv*, abs/2112.04359.

Mike Zhang and Barbara Plank. 2021. Cartography active learning. In *Conference on Empirical Methods in Natural Language Processing*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *ArXiv*, abs/1509.01626.

Ye Zhang, Matthew Lease, and Byron C. Wallace. 2016. Active discriminative text representation learning. In *AAAI Conference on Artificial Intelligence*.

Zhisong Zhang, Emma Strubell, and Eduard H. Hovy. 2022. A survey of active learning for natural language processing. *ArXiv*, abs/2210.10109.

Kang Zhou, Yuepei Li, and Qi Li. 2022. Distantly supervised named entity recognition via confidence-based multi-class positive and unlabeled learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7198–7211, Dublin, Ireland. Association for Computational Linguistics.

10. Language Resource References

Sameer Pradhan and Alessandro Moschitti and Nianwen Xue and Olga Uryupina and Yuchen Zhang. 2012. *CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*.

A. Experimental Set-Up

A.1. Number of Examples for Annotation

Due to computation budgets, we gradually increase the number of examples we annotate at each step to ensure a larger coverage of the complete dataset towards the end without the need of performing a large number of active learning iteration. In the beginning, we annotate 1% to allow for a finer-grained view of the performance as the number of annotated tokens increase. More precisely, the annotation scheme is: {1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 1%, 2.5%, 2.5%, 2.5%, 2.5%, 5%, 5%, 5%, 10%, 10%, 10%, 10% }, such that at the last iteration we will have around 80% of the full dataset annotated.

A.2. Datasets

The datasets we use have different named entities. CoNLL2003 has *person*, *location*, *organization*, and *miscellaneous*.

And OntoNotes 5.0 has *person*, *norp*, *fac*, *org*, *gpe*, *loc*, *product*, *date*, *time*, *percent*, *money*, *quantity*, *ordinal*, *cardinal*, *event*, *work_of_art*, *law*, *language*.

A.3. Hardware

We ran all our experiments on a system with Tesla V100 SXM2 32 GB GPUs. We used `distillbert`, a lighter transformer-based model. From a computation perspective, one single experiment takes approximately 1 hour for CoNNL-2003 and 4 hours for OntoNotes on a single GPU.

A.4. Hyperparameters

We use a learning rate of 5e-5 and train for at most 20 epochs for each active learning iteration. We use early stopping with a patience of 3 epochs.