Impact of Training Instance Selection on Domain-Specific Entity Extraction using BERT

Eileen Salhofer
Know-Center GmbH
esalhofer@know-center.at

Xinglan Liu
Know-Center GmbH
lliu@know-center.at

Roman Kern
Graz University of Technology
Know-Center GmbH
rkern@know-center.at

Abstract

State of the art performances for entity extraction tasks are achieved by supervised learning, specifically, by fine-tuning pretrained language models such as BERT. As a result, annotating application specific data is the first step in many use cases. However, no practical guidelines are available for annotation requirements. This work supports practitioners by empirically answering the frequently asked questions (1) how many training samples to annotate? (2) which examples to annotate? We found that BERT achieves up to 80% F1 when fine-tuned on only 70 training examples, especially on biomedical domain. The key features for guiding the selection of high performing training instances are identified to be pseudo-perplexity and sentence-length. The best training dataset constructed using our proposed selection strategy shows F1 score that is equivalent to a random selection with twice the sample size. The requirement of only a small number of training data implies cheaper implementations and opens door to wider range of applications.

1 Introduction

Information extraction (IE) is the process of turning unstructured texts into structured data (Jurafsky and Martin, 2021), and is one of the most widely used natural language processing (NLP) tasks in industrial applications. Named entity recognition (NER) is an IE task of tagging entities in text with their corresponding types. Most existing NER methods require either handcrafted features, and/or a large number of annotated examples (Jurafsky and Martin, 2021), both of which are labor intensive.

Recent advances in transformers (Vaswani et al., 2017) and BERT (Devlin et al., 2019) changed the landscape for many NLP tasks. Significant performance gain can be achieved by fine-tuning language models on a small number of training examples due to transfer learning. As a result, the pipeline of annotating – fine-tuning becomes common practice. Following this pipeline, the first step for each use case is to annotate application specific data. It is therefore beneficial to estimate in advance how many training samples need to be annotated, as well as which samples to annotate.

This work answers these two frequently asked questions through empirical studies on the NER task. Specifically, we repeatedly down-sample benchmark datasets and fine-tune BERT models for the downstream task of token classification. Two benchmark datasets (1) general domain Conll2003 (F. and De Meulder, 2003) and (2) biomedical domain BC5CDR (Li et al., 2016) are used in this study.

In summary, our main contributions are:

- Empirically identified the relation between sample size and model performance on the entity extraction task for corpora of different domains.
- Proposing key measures for selecting training examples that yield high performances in our evaluation, which can serve as a promising starting point for many other application scenarios.

2 Experimental Setting

The goal of the experiments is to answer beforementioned questions on how many and which training samples to annotate for the named entity extraction task.

We repeatedly down-sample benchmark NER datasets and compared model performances fine-tuned on different number of training examples and different samples. Two datasets of different domains, and two BERT models pretrained on different datasets are used in this study.

2.1 Fine-Tuning Language Models

As recommended in Devlin et al. (2019), the NER task is formulated as a token-level classification

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pages 83 - 88

	CoNLL2003 *(news)*						BC5CDR *(PubMed, PMC)*			
split	n-sentence	n-token *mean*	n-LOC *mean*	n-MISC *mean*	n-ORG *mean*	n-PER *mean*	n-sentence	n-token *mean*	n-Disease *mean*	n-Chemical *mean*
train	14042	14.50	0.51	0.24	0.45	0.47	4612	24.95	0.91	1.13
validation	3251	15.80	0.57	0.28	0.41	0.57	4607	24.81	0.92	1.16
test	3454	13.44	0.48	0.20	0.48	0.47	4819	25.02	0.92	1.12

Table 1: Number of sentences, the mean of the number of tokens and entities for CoNLL2003 and BC5CDR datasets. On average, sentences in BC5CDR are nearly twice as long as those in CoNLL2003.

task. Namely, a pretrained BERT model is stacked with a linear layer on top of the hidden-states output, before fine-tuned on training examples. The transformers library from Hugging Face (Wolf et al., 2020) is used for fine-tuning. Two BERT models are compared: (1) BERT[1] pretrained on BooksCorpus (Zhu et al., 2015) and Wikipedia, which represent general domain. (2) BioBERT[2] (Lee et al., 2020) where also PubMed abstracts and PMC articles are added to the pretraining data. As a result, the pretraining data for BioBERT also covers the biomedical domain. For both pretrained models, we choose the base setting with 12 transformer layers and 768 hidden embedding sizes. Following recommendations from both Devlin et al. (2019) and Lee et al. (2020), the cased vocabulary is used for the NER task.

2.2 Datasets

Two NER datasets with different domains were used and statistics for both graphs are provided in Table 1.

CoNLL2003 (English) dataset (F. and De Meulder, 2003)) is one of the most commonly used NER datasets. The corpus consists of 1.4K news articles with four types of entities (LOCations, ORGanizations, PERsons, and MISCellaneous) being annotated.

BC5CDR dataset (Li et al., 2016) consists of 1.5K PubMed articles, where two types of entities (chemical and disease) are annotated.

2.3 Down-Sample

To study the relation between model performance and training sample size, we uniformly draw N ($N \in \{50, 150, 500, 1000, 2000\}$) sentences at random from the training split, with the constraint that at least one instance from each IOB (Inside–Outside–Beginning) class is present in the sample.

[1] https://huggingface.co/bert-base-cased

[2] https://huggingface.co/dmis-lab/biobert-v1.1

3 Results

We first establish a baseline using the full dataset, which also serves as an upper bound. Next, we compare the F1 scores for each dataset for different random sample-sizes and for the training subset selected using our proposed method. Finally, we conclude the analysis with a recommended workflow for training instance selection.

3.1 Corpora Domain for Pretraining and Fine-Tuning

We first select a pretrained BERT model for each dataset. Table 2 shows the best F1 score on the test data for CoNLL2003 and BC5CDR datasets, using pretrained BioBERT and BERT.

	CoNLL2003	BC5CDR
BERT	**91.4**	84.9
BioBERT	89.1	**88.2**

Table 2: F1 score on test data for CoNLL2003 and BC5CDR datasets, using different pretrained models BioBERT and BERT. Best performance is observed when the domain for pretraining matches that of the downstream task.

Similar to previous work (Lee et al., 2020; Gururangan et al., 2020), best performance is observed when the domain for language model pretraining matches that of the downstream task. For further experiments, we choose pretrained BERT model for CoNLL2003 dataset and BioBERT for BC5CDR dataset.

3.2 Effect of Sample Size

Next, we fine-tune a BERT language model on the randomly down-sampled datasets of different size, and the F1 performance in entity extraction on the test split is summarized in Figure 1. For sample size below 200 sentences, the model performance

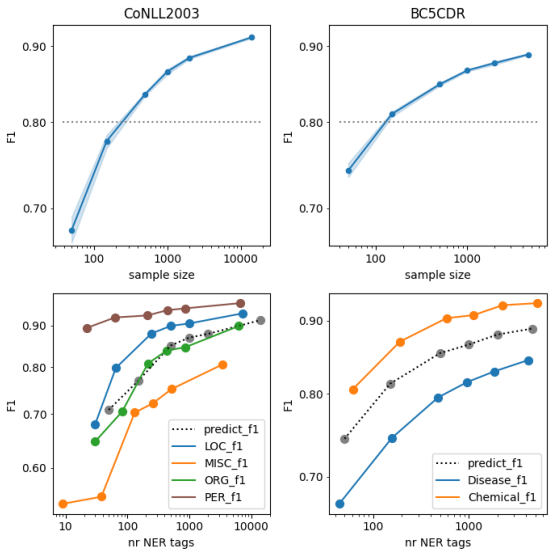

Figure 1: *Top*: performance (micro F1) in entity extraction on the test split for random selection of the training data subset of different sample sizes (number of sentences). The shading represent 95% confidence interval over 8 different runs using the same data and same training parameter. To reach F1 score of 80%, only 150 and 300 sentences are needed for BC5CDR and CoNLL2003 dataset, respectively. *Bottom*: F1 per NER class as a function of the number of tokens tagged per class. We observe a difference in performance between the NER classes, which cannot be explained by the number of respective tokens in the training set.

increases very fast . Above 200 sentences, the increase in F1 score slows down when more training examples become available.

Different fine-tuning runs show very low variance (shown as shaded band in Figure 1). The variance, however, increases as the sample size decreases, as expected.

Within each sample, the number of observations for each entity class may be different from each other. Would the same scaling hold for each entity class? In other words, can the differences in F1 score per class be explained by the differences in the number of observations? Figure 1 plots F1 score per class as a function of number of tokens tagged with that class. We observe that although NER classes with less observations show lower F1 score than those with a larger number of observations, the curves per class do not fall on the same line. This suggests that the difference in the number of observations is not the only reason for the differences in F1 per NER class.

Furthermore, for CoNLL2003, the F1 score for MISC entities shows the lowest value for all sample sizes. The MISC class has the lowest number of observations (see also Table 1), which causes the lower F1-MISC, which in turn reduces the overall F1 score.

3.3 Effect of Sample Seed

In this experiment, we empirically investigate if fine-tuning on different training samples results in similar performance.

10 different random samples of size 50 are generated, following Section 2.3, and F1 performances of the BERT models fine-tuned on the different samples are reported in Table 3. 7 to 8 points difference in F1 score observed between best and worst random samples, which is much higher than the variations between different runs of the same sample. The difference is the highest for the lowest sample size, suggesting the importance of sampling optimization, especially when annotated data is limited.

3.4 Training Instance Selection

The large difference in model performance between different random training samples raises the pos-

	BC5CDR	
sample size	50	150
variation runs std	1.3	0.5
variation runs min-max	3.4	1.4
worst random	66.6	79.3
best random	74.4	81.3
best kernel density	78.5	83.4
	CoNLL2003	
sample size	50	150
variation runs std	2.0	1.3
variation runs min-max	5.3	3.5
worst random	61.6	73.5
best random	70.8	78.2
best kernel density	71.6	75.6

Table 3: F1 score on test split for CoNLL2003 and BC5CDR datasets, finetuned on different training samples (random or selected via our proposed method) of size 50 and 150. The best random sample shows up to 8 points higher F1 than the worst random sample, which is much higher than the variations between 8 different runs of the same sample. The sample guided by kernel density (see section 3.4) improves further over the best random sample.

sibility to improve training instance selection. In order to identify the key features that differentiate a "good" random sample from a "bad" one, we first investigate several potential features to characterise the different random samples, before selecting the two most differential features. Finally, we propose a sampling strategy guided by the identified key features.

Identifying Key Features

Since the goal is to select training instances before annotation, we only include features that can be computed without labeled data. Three types of features are investigated for characterising the training examples. (1) Descriptive statistics including sentence-length and coverage over different documents. (2) "Fluency" measures include perplexity and pseudo-perplexity (Salazar et al., 2020) for masked language model like BERT, which are computed by masking tokens one by one. (3) Diversity measures as recommended in Mccarthy and Jarvis (2010).

The most differentiating features turn out to be sentence length (number of tokens) and pseudo-perplexity, while all three diversity measures are very similar across different samples. Thus we omitted diversity measures in this study and leave it to future research.

Figure 2 top row shows median sentence-length per random sample vs median pseudo-perplexity, where the coloring represents F1 score on the evaluation split when model is trained on this random sample. Fine-tuning model on samples on the periphery tend to result in higher F1 score than those in the center.

Training Instance Selection

2-dimensional kernel density estimation is used to capture the observed relation between sentence length, pseudo-perplexity and F1 score (Figure 2 *bottom*). We then proceed to generate training instances based on the kernel density profile[3]. Different sampling ratios are tested, and the best performing setting is to sample 85% of the training instances from the 15% sentences at the lowest density. The results on the improved training sample can be found in Table 3.

For the BC5CDR dataset, the best sampling achieves F1 of 78.5 and 83.4 for sample size of 50 and 150, respectively. Using the relation in

[3]We release all code for future studies at https://github.com/tugraz-isds/kd

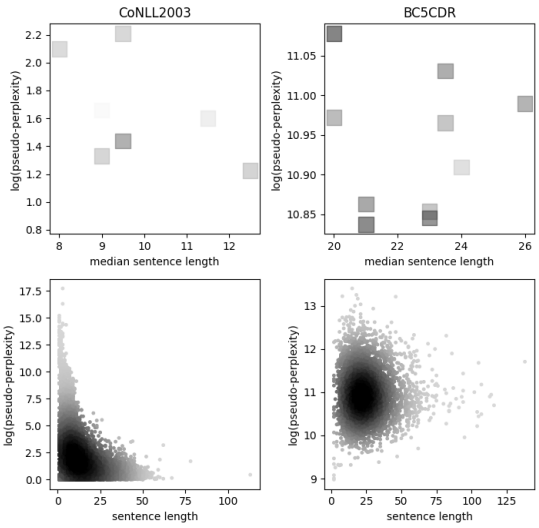

Figure 2: *Top*: Median sentence-length per random sample vs median pseudo-perplexity, where the coloring represents F1 score on the evaluation split when model is trained on this random sample. Fine-tuning model on samples on the periphery result in higher F1 score than those in the center. *Bottom*: Sentence-length vs pseudo-perplexity for all training samples, colored by kernel-density. The random samples with higher F1 scores have median pseudo-perplexity and median sentence length values that are located around the periphery, i.e. the lower density area, especially for the BC5CDR dataset.

Figure 1, this level of F1 is equivalent to the performance of a random sample with size 120 and 400, respectively. In other words, a smart sampling is worth more than twice as many training examples.

For the CoNLL2003 dataset, the F1 score of the optimized sample does not consistently outperform random sampling. Possibly because the Gaussian kernel density estimator does not fit very well to the map with pseudo-perplexity vs sentence length. In addition, the CoNLL2003 dataset shows larger variation over different finetuning runs, and contains sentences that are not as "clean" as those in the BC5CDR dataset. For instance, sentences like *"4-6 7-6 (7-4)"* or *"————————"*.

Compared to the full training set, our best sample with sample size 150 is only 5 points lower in F1, albeit with less than 4% of training data size.

The optimised sampling can also be intuitively understood: (1) longer sentences have higher chance to contain more NER tagged tokens; (2) instances with higher perplexities offer more "learnings" for the pretrained model; (3) samples that weigh more on rare instances are apparently more enabling for BERT language models.

We notice that although our best sampling leads

to 2 - 4 points improvement in F1 over the best random samples, our empirical way for sample selection is possibly only at a local maximum.

Training instance selection work flow

Based on this result, our recommended workflow for training instance selection is summarized in Figure 3.

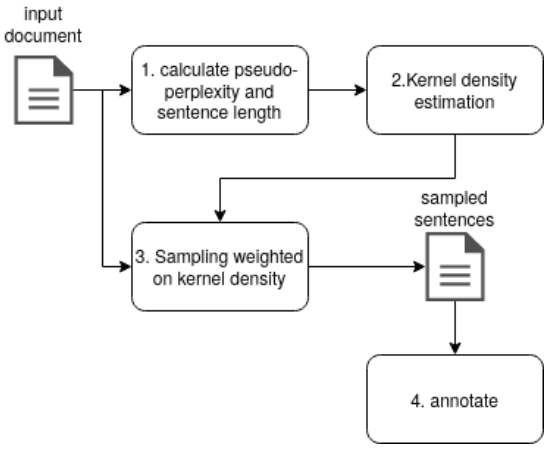

Figure 3: Recommended workflow for annotating customised dataset.

To select the best sample for annotation, first of all, pseudo-perplexity and sentence length should be calculated for all unlabelled text. A kernel density estimator can then be used to fit the relation. Finally, the optimised samples can be drawn weighing on kernel density, before being annotated.

We notice that the proposed workflow differs from typical active learning (Olsson, 2009) approaches, in the sense that no active feedback or interaction with oracle is included. It is thereby a complementary simpler approach for training instance selection.

4 Conclusions

It can be shown that domain-specific pre-trained BERT performs well even when fine-tuned only on small amounts of training samples. Initial increase in amount of data leads to large performance gain before saturating at around 200 training examples. For small data sizes, the F1 scores of different random samples vary greatly.

A sampling strategy is proposed in this work which uses kernel density estimate to balance the instance selection between pseudo-perplexity and sentence length.

The F1 scores of BERT models fine-tuned on training sets constructed using our method are equivalent to the same model fine-tuned on a random sample using twice as many training examples.

This work provides practical guidelines for annotation requirements, namely, data size and sampling strategy. Given the reduced number of training instances needed due to sampling optimisation, data annotation becomes less expensive and can be achievable in more use cases.

Acknowledgements

The research was conducted under the framework of the ECSEL AI4DI "Artificial Intelligence for Digitising Industry" project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The Know-Center is funded within the Austrian COMET Program—Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG. We acknowledge useful comments and assistance from our colleagues at Know-Center and at Infineon.

References

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186.

Tjong Kim Sang Erik F. and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Dan Jurafsky and James H. Martin. 2021. *Speech and Language Processing*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*.

Philip Mccarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42:381–92.

Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. pages 2699–2712.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.