Fine-grained Location Extraction via Curriculum Learning

Anonymous ACL submission

Abstract

Named Entity Recognition (NER) seeks to extract entity mentions from texts with predefined categories such as *Person*, *Location*. General domain NER datasets like CoNLL-2003 mostly annotate *Location* coarse-grained entities (e.g., a country or a city). However, many applications require to identify fine-grained locations from texts and map them precisely to geographic sites (e.g., a crossroad or a store). Therefore, we propose a new NER dataset HarveyNER with fine-grained locations annotated in tweets. This dataset presents unique challenges and characterizes many complex and long location mentions in informal descriptions. Considering Curriculum Learning can help a system better learn the hard samples, we adopt it and first design two heuristic curricula based on the characteristic difficulties of HarveyNER, and then propose a novel curriculum that takes the commonness of sample difficulty into consideration. Our curricula are simple yet effective and experimental results show that our methods can improve both the hard case and overall performance in HarveyNER over strong baselines without extra cost.

1 Introduction

Named Entity Recognition (NER) task aims to locate and classify textual phrases as entity mentions that belong to predefined entity categories. *Location* is one of the general entity categories and has been included in many NER datasets, including CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2013). However, the scope of the location defined in these datasets is vague, and they contain coarse-grained entities such as a continent (e.g., Europe), a country (e.g., the U.S.), or a city (e.g., London). In practical applications, many systems require identifying fine-grained location entities such as an apartment (e.g., Bayou Oaks) or a specific store (e.g., the HEB on Montrose) from texts to locate the geographic places on a map, which is vital to identify actionable information from situational awareness (Khanal and Caragea, 2021). For example, in Figure 1, a flood disaster happened in the Houston area and then someone tweeted the shortage of necessities in two locations. If a disaster response system can detect the disaster-related tweets, identify the two location mentions from the text, and link them to location entities on the map, necessary help can be directly delivered to the people living in disaster-affected places. Accurately identifying the fine-grained location mentions plays a critical role in such a system.

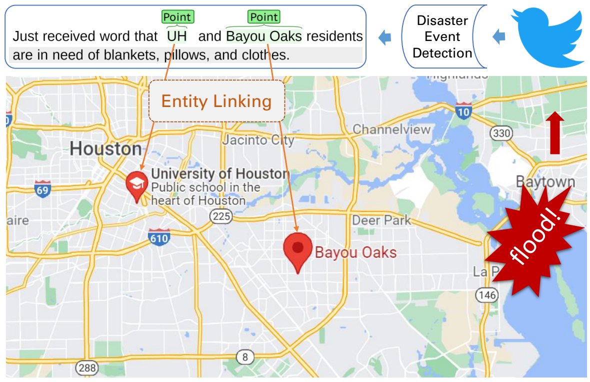

Figure 1: An example of a disaster response system.

Considering the necessity of suitable datasets, some previous work tried to either automatically (Middleton et al., 2013) or manually (Khanal et al., 2021) annotated crisis-related location extraction datasets. However, they ignore that these location mentions are meant to precisely map to geo-coordinates and their annotation quality is limited for such applications. We closely obey the practical needs and propose a dataset HarveyNER that annotates such coordinate-oriented location mentions from tweets. Specifically, we select the tweets about Hurricane Harvey affecting the Houston metropolitan area in 2017 and then annotate the location mentions located in this city from the tweets. During the annotation, we carefully construct the guidelines and train annotators to control the quality. Compared with the location mentions in previous NER datasets, HarveyNER focuses on the location mentions that can link to specific sites on a map. For example, *"the corner of Richey St and W Harris Ave in Pasadena"* is an intersection of two roads and we annotate it as a *Point*, but previous work regard it as two *Road* mentions *"Richey St"* and *"W Harris Ave in Pasadena"* that are not as helpful in applications. This is the first dataset that contains such coordinate-oriented location annotations meriting applicational values. We use the Harvey disaster in Houston as an example to demonstrate how to annotate such location mentions and how to improve the NER performance on such datasets. We do not expect the dataset can generalize to other applications.

However, the unique characteristics of HarveyNER bring challenges for existing systems. For one thing, many entities are long and complex to precisely point to a place. E.g., the previous *Point* entity contains up to 11 words, and it could be wrongly recognized as two roads entities by a NER system; for another, as an instant social medium, tweets contain many informal contents, local conventions, and even grammatical errors, making the HarveyNER even more ambiguous. For example, the abbreviations in the previously mentioned location (*"UH", "St", "Ave", etc.*) bring many out-of-vocabulary (OOV) words that cannot fully utilize pre-trained word embedding such as Glove (Pennington et al., 2014) or BERT (Devlin et al., 2019).

In order to improve the performance on these hard location mentions, we propose to adopt Curriculum Learning (CL) (Bengio et al., 2009) that can learn difficulty samples better when ordering examples during training based on their difficulty. One big precondition to utilize CL for training is to distinguish between easy and hard samples. Considering that there are many long and complex entities in HarveyNER that are naturally difficult (as in Figure 3, the performance of baselines are saliently worse on these hard cases), we directly design two corresponding heuristic curricula. We further assume that easy cases are not necessarily the shortest or least complex entities, but could be the most common ones with abundant training examples. Then we propose a novel curriculum with a difficulty scoring function that comprehensively considers the commonness of the two heuristic difficulty metrics. Empirical results show that all of the heuristic curricula can improve both the hard case and overall NER performance over strong baselines and our novel curriculum performs best.

We also find that different NER systems may need different curriculum scheduling strategies, and the normal curriculum (training easier samples first) is better for the neural network-based model and the anti-curriculum (training harder samples first) performs better for the language model-based system.

2 Related Work

NER research has a long history and many NER datasets have been proposed based on different applications with different entity categories. General domain datasets such as CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2013) attend to certain common entity types including *Location*. The location mentions in these datasets such as a country (e.g., the U.S.) or a city (e.g., London) are coarse-grained. Li and Sun (2014); Ji et al. (2016) focus on identifying fine-grained points-of-interest for location-based services, and their dataset is automatically constructed by mapping location inventory to tweets. Khanal and Caragea (2021); Khanal et al. (2021) try to identify crisis-related location mentions but their dataset quality is limited for a disaster response system. Our proposed dataset HarveyNER closely follows applicational needs and focuses on fine-grained locations that can map to coordinates on a map.

Recent approaches (Yang and Zhang, 2018; Li et al., 2020; Chen et al., 2021) using Neural Network models like BiLSTM-CNN-CRF (Ma and Hovy, 2016) and contextual embeddings like BERT (Devlin et al., 2019) have greatly improved the NER performance. However, none of these approaches consider the difficulty of different NER cases in their model training. Bengio et al. (2009) pointed out that using a curriculum strategy enables the model to learn from easy examples to complex ones and leads to generalization improvement. Many Natural Language Processing tasks such as machine translation (Platanios et al., 2019; Liu et al., 2020; Zhang et al., 2021), natural language understanding (Xu et al., 2020), text generation (Liu et al., 2018, 2021) and dialogue systems (Su et al., 2021) benefit from such curriculum learning strategies. Considering the characteristics

Data Split	Train	Valid	Test	Total
All Tweets	3,967	1,301	1,303	6,571
Tweets w/ Entity	1,087	366	353	1,806
Tweets w/o Entity	2,880	935	950	4,765
All Entity Type	1,581	523	500	2,604
Point	591	206	202	999
Area	715	236	212	1,163
Road	158	51	57	266
River	117	30	29	176

Table 1: Statistics of HarveyNER.

of HarveyNER containing many complex cases, we design corresponding curricula to learn them.

3 The HarveyNER Dataset

3.1 Data Preparation

Data Collection Considering the immediacy requirement of a disaster response system, we choose texts from instant social media Twitter. Specifically, we used the Twitter PowerTrack API to retrieve the tweets posted between 5:00 a.m., August 25, and 4:59 a.m., August 31, 2017. This was the time range of peak disruption caused by Hurricane Harvey in the Houston area. In total, we collect 1,121,363 tweets, excluding retweets and replies.

Data Cleaning In order to filter irrelevant tweets, we apply several strategies. First, we only keep the tweets that are related to the Houston area, i.e., the geo-coordinates of the tweets or the profile location of the authors within the bounding of Houston. Second, we adopt a weakly supervised event detection algorithm (Yao et al., 2020) to identify tweets on disaster-related topics; these tweets have a high probability relating to Hurricane Harvey at this time range. We also manually filter the remaining irrelevant tweets (like non-English and repeated ones) during the annotation process. In total, 6,571 tweets are selected for this study, as in Table 1.

3.2 Location Entity Annotation

Annotation Types HarveyNER focuses on the coordinate-oriented locations so we mainly annotate *Point* that can be precisely pinned to a map and *Area* that occupies a small polygon of a map. Considering that some disasters can affect line-like objects (e.g., a flood can affect the neighbors of a whole river), we also include *Road* and *River* types.

	A1 & A2	A1 & A3	A2 & A3	Average
κ (%)	85.64	82.17	83.12	83.64

Table 2: Inter-Annotator Agreement. A# represents No.# annotator.

- **Points**: denote an exact location that a geo-coordinate can be assigned. E.g., a uniquely named building, intersections of roads or rivers;
- **Areas**: denote geographical entities such as city subdivisions, neighborhoods, etc;
- **Roads**: denote a road or a section of a road;
- **Rivers**: denote a river or a section of a river.

Quality Control In order to guide the annotators to correctly annotate the fine-grained location mentions, especially to distinguish the *Point* locations, we take several measurements to control data quality. We make some initial annotation exercises and receptively update annotation guidelines to reduce ambiguity and subjectivity. The detailed guidelines can be found in Appendix A.1.

With the guidelines, we train 3 annotators and test their Inter-Annotator Agreement (IAA) on 500 randomly selected tweets. We pairwise calculate the Cohen's kappa (κ) scores based on the token-level BIO (Beginning, Inside and Outside a entity) annotations from each pair of annotators. As in Table 2, we observe a high average κ score of *83.64%*. After that, the 3 annotators start annotating the remaining tweets independently. Exampels of the annotation disagreement can be found in Appendix A.2.

3.3 Dataset Analysis

Datasets	HarveyNER	CoNLL-2003 (Loc-only)
Avg. Ent. Len. (word)	**2.68**	1.15
Avg. Ent. Len. (char)	**13.91**	7.24
Complex Ent. Rate (%)	**11.8**	0.19
OOV Rate (%)	**14.47**	2.33
Avg. Sent. Len. (word)	20.07	14.53
Avg. Sent. Len. (char)	117.03	76.89
Avg. Ent. Count	0.40	0.51
– non-empty	1.44	1.38
Avg. Ent. Ratio (%)	5.33	7.23
– non-empty (%)	19.39	19.43

Table 3: HarveyNER v.s. CoNLL-2003. "non-empty" excludes the sentences without location mentions.

General Statistics We quantitatively analyze the HarveyNER dataset, and the resulting statistics are

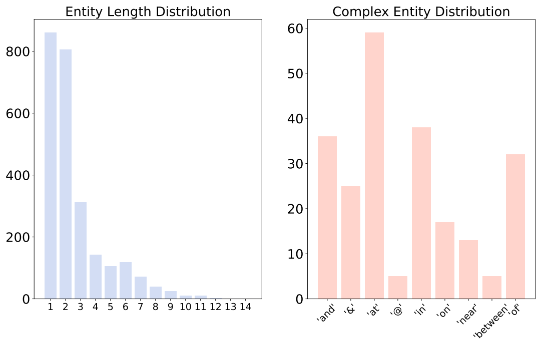

Figure 2: Distributions of the difficult examples.

Indicators	Examples
"and"	the corner of Richey St **and** W Harris Ave in Pasadena
"&"	Beltway 8 **&** Tidwell
"at"	Brazos River **at** Richmond
"@"	Copperfield Church **@** 8350 hwy 6 north
"in"	Constellation Field **in** Sugar Land
"on"	Chimney Rock **on** I-10 East
"near"	IH 10 **near** Monmouth.
"between"	249 **between** Cypresswood / Louetta
"of"	University **of** Houston

Table 4: Examples of complex entity.

shown in Table 1. Among the 6,571 annotated tweets, we can see that about *27.48%* of them contain at least one location entity and the remaining do not mention any target location. We randomly split the annotated tweets into training (3,967), validation (1,301), and test(1,303) sets for experiments with a ratio of 6:2:2. As for location types, *Point* and *Area* entities occupy the majority as *38.36%* and *44.66%* respectively, while *Road* and *River* only make up *10.22%* and *6.76%* respectively.

Comparison with CoNLL-2003 Different from general NER datasets that annotate coarse-grained locations from news articles, our HarveyNER dataset is characterized with fine-grained annotations from informal Twitter texts. As presented in Table 3, we compare our HarveyNER dataset with CoNLL-2003 on a range of aspects to demonstrate its characteristics.

First comes the entity length comparison. It is salient that entities in HarveyNER are longer on average (*133.04%* longer at word-level and *92.13%* longer at character-level). This is in line with our intuition because HarveyNER contains many precisely described locations in order to locate them on a map. The entity length distribution is shown in Figure 2.

To better analyze these long entities in detail, we use some heuristic rules to probe what types of complex entities and how many of them exist in the dataset. Specifically, after our manual analysis on the validation set, we selected 9 tokens (*"and","&", "at", "@", "in", "on", "near", "between", "of"*) as complex entity clues. If an entity contains any of these tokens, we regard it as a complex one. As in Table 3, the HarveyNER contains about *14.47%* complex entities, while such entities barely exist in the CoNLL-2003 (*0.19%*). The detailed distribution of these complex entities with different indicators can be found in Figure 2. We also list some examples of these complex entities in Table 4 with these indicators. We can see that these entities are indeed complex, and even we human beings need to make efforts to resolve them.

As we mentioned before, the language used in tweets is informal and contains many abbreviations and even grammatical errors. In order to quantitatively analyze the informal texts, we calculate the out-of-vocabulary (OOV) rates for the datasets by counting words that are absent from the pretrained Glove[1] (Pennington et al., 2014) word lists. We can see that the HarveyNER has a much higher OOV rate than CoNLL-2003 (*14.47%* vs. *2.33%*). The high OOV rate could degrade the performance of NER systems relying on pre-trained word embeddings like Glove or language models like BERT (Devlin et al., 2019).

Apart from the difficult aspects of HarveyNER, we also compare some other metrics of interest. To our surprise, the average sentence length of the HarveyNER is about *38.13%* and *52.20%* longer than that of CoNLL-2003 at word-level and character-level, respectively. This phenomenon is counter-intuitive since the tweet content is strictly constrained to be no more than 140 characters each. One possible reason could be that the short tweets are usually irrelevant to the Hurricane and have been filtered by the disaster detection system we used (Yao et al., 2020).

As for the average location entity count for each sentence of the two datasets, the results show that there is no big difference between the HarveyNER and CoNLL-2003, either for all the texts (0.40 vs. 0.51) or for those sentences containing at least one entity (1.44 vs. 1.38). A similar phenomenon also exists in the average entity ratios of the two datasets. The entity ratio is the proportion of entity words in a sentence and we calculate the average across

[1]For fair comparison, we use glove.twitter.27B for HarveyNER and glove.6B for CoNLL-2003.

sentences. It turns out that the two datasets have similar entity ratios (*5.33%* vs. *7.23%* for all sentences and *19.39%* vs. *19.43%* for all non-empty ones). The reason may be that even though HarveyNER has longer entities, it also has larger sentence lengths. From these aspects, HarveyNER shares the same level of difficulty with CoNLL-2003.

4 Curriculum Arrangement

In consideration of the characteristic difficulties of HarveyNER, we employ curriculum arrangements to help learn these hard cases. There are many different approaches to implementing a curriculum. We follow the curriculum designing approach introduced by Bengio et al. (2009), which mainly requires to specify two functions:

- **Difficulty Scoring Function**: Given an input sample $\boldsymbol{x}_i$, this function map it to a numerical score, $d(\boldsymbol{x}_i) \in \mathbb{R}$. The score is used to represent the difficulty level of the corresponding sample and usually the higher the score, the more difficult the sample is.
- **Pacing Function**: The pacing function $p(t) \in (0, 1]$ specifies the input training data size at time or step t. Normally we use $p(t)$ the lowest difficulty-scored samples for training at time t, but in the ***anti-curriculum*** setting, we use $p(t)$ the highest difficulty-scored samples. Given such a subset of the dataset containing the easiest or hardest ones, we sample training batches uniformly from it for training.

The curriculum learning procedure using the two functions is described in Algorithm 1.

4.1 Three Difficulty Scoring Functions

We first design two dataset-specific heuristic curricula, based on maximum entity length and entity complexity[2], inspired by the dataset analysis in Section 3.3. Then, we introduce a new metric that integrates the two heuristic metrics.

Maximum Entity Length (Max): As mentioned before, our HarveyNER dataset has longer entity length than CoNLL-2003 on average, and this brings many long and difficult entities that are hard to identify. Intuitively, we can design a corresponding curriculum based on such entity-level difficulty. Specifically, given an input sample $\boldsymbol{x}_i$ contains n words: $\boldsymbol{x}_i = \{w_1, w_2, \dots, w_n\}$, the

[2]We tried using the OOV rate as the difficulty score in our experiment, but the performance is not as good.

Algorithm 1 Curriculum Learning with Scoring and Pacing Functions

Input:
- The training Data, $\mathcal{D}^{\text{train}} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, including N samples;
- A model $\mathcal{M}$ that takes batches of data for training at each step t;
- A difficulty scoring function d;
- A pacing function $p(t)$.

Output: A model $\mathcal{M}^{\text{trained}}$ trained with the curriculum.

Compute the difficulty score $d(\boldsymbol{x}_i)$ for each sample;
Sort $\mathcal{D}^{\text{train}}$ ascendingly or descendingly based on $d(\boldsymbol{x}_i)$ and obtain $\mathcal{D}^{\text{train}}_{\text{sorted}}$;
Initialize the pacing function $p(0)$;
Generate the initial curriculum $\mathcal{D}_0$ using the top $p(0)$ samples in $\mathcal{D}^{\text{train}}_{\text{sorted}}$;
for training epoch $t = 1, 2, \dots$ **do**
    Uniformly sample batches from the current curriculum $\mathcal{D}_{t-1}$ for model training;
    Update the pacing function $p(t)$ based on equation Eq. (6);
    Generate the next curriculum $\mathcal{D}_t$ using the top $p(t)$ samples in $\mathcal{D}^{\text{train}}_{\text{sorted}}$;

sample can have $k \geq 0$ entities, $\{E_1, E_2, \dots, E_k\}$. Each E_j is a subset of $\boldsymbol{x}_i$ ($\forall\, 0 < j \leq k : E_j \subseteq \boldsymbol{x}_i$). $|E_j|$ represents the number of words that *j-th* entity contains or the length of *j-th* entity. Now, we can assign each sample that has entity or entities in it $\boldsymbol{x}_i$ a score using the longest entity length[3] it has:

$$d_{\text{max}}(\boldsymbol{x}_i) = \max(L_i) \qquad (1)$$

L_i is the set of entity length for the i-th sample $\boldsymbol{x}_i$, i.e. $L_i = \{|E_1|, |E_2|, \dots |E_k|\}$. With such a scoring function, we need to pay attention to the samples without any entity mentioned (about *72.52%* as in Table 1) since their difficulty scores will all be 0. In this case, the algorithm will put all these samples in one step to the curriculum, which will mislead the model to a local minimum and learn that no entity exists in the data. We propose a remedy to this issue by randomly feeding the empty samples. When we order our dataset by the difficulty scores, those non-entity samples will be randomly interspersed among the ordered samples

[3]We also tried using the average entity length as the difficulty score in our experiment but the performance is not as good.

which have entities.

Complex Entity Rate (Complex): Corresponding to the analysis about the complex entity rate in HarveyNER, we define another difficulty scoring function. Specifically, we define the complexity of entity $c(E)$ as whether the entity contains words or symbols such as *"and", "&", "at"*, etc and what symbols the entity contains. We set up a complexity dictionary based on the heuristic analysis with these complex entities, i.e., $\{$*"and"* : 3, *"&"* : 3, *"at"* : 2, *"@"* : 2, *"in"* : 2, *"on"* : 2, *"near"* : 2, *"between"* : 2, *"of"* : 1$\}$. The larger value implies the more complex the entity is. Because each entity E can contains many "complexity" indicators, we choose the largest one. For example, a aforementioned entity E *"the corner of Richey St and W Harris Ave in Pasadena"* contains *"of"*, *"and"* and *"in"* indicators, we say the complexity value of this entity is $c(E) = 3$, because of 3 > 2 > 1. Besides, one sample x_i may have multiple entities with different complex rates $C_i = \{c(E_1), c(E_2), \ldots, c(E_k)\}$, we also choose the maximum complexity value to determine the complexity value for the sample, i.e.,

$$d_{\text{complex}}(\boldsymbol{x}_i) = \max(C_i) \tag{2}$$

However, if the sample's entities do not have those complex clues at all, the complex entity rate for that sample will be simply 0, which we regard as a simple data point. Such a scoring function based on the entity mentioned will encounter the same issue as with the **Max** scoring function because if a sentence does not contain any entity, calculating the complexity value of that sample will be meaningless and unreasonable. We use the same remedy as well and randomly interspersed these non-scored samples among the ordered samples.

Commonness of Difficulty (Commonness): In addition to these heuristic-based scoring functions, we propose a comprehensive metric that incorporates both of these two difficulties. We assume that easy cases are not necessarily to be the samples with shortest entities or lowest complex entity rates but should be the most common cases with abundant training examples. Thus, we need to answer a question: what are the most common cases? We use the previously mentioned two metrics (the *Maximum Entity Length* and the *Complex Entity Rate*) as the two dimensions for representing the commonness, i.e., the commonness of difficulty level evaluated by the two metrics. This means that if a sample has the most common maximum entity length and the most common complex entity rate, it should be the easiest.

We propose a new difficulty score to represent the commonness. As in Eq. (3), we first count the number of training samples have the same difficulty score with the sample $\boldsymbol{x}_i$, and then divide it by the total number of instances N. Because we expect the smaller values indicating more commonness or easiness, we take the reciprocal of it and get f_{metric}. Here d_{metric} are the difficulty metrics $d_{\max}$ or d_{complex}.

$$f_{\text{metric}}(\boldsymbol{x}_i) = \frac{1}{count(d_{\text{metric}}(\boldsymbol{x}_i))/N} \tag{3}$$

After having commonness values for maximum entity length $f_{\max}$ and complex entity rate f_{complex}, we re-scale them to the same range of of $[0,\ 1]$ as in Eq. (4).

$$f_{\text{metric}}(\boldsymbol{x}_i) = \frac{f_{\text{metric}}(\boldsymbol{x}_i) - \min(f_{\text{metric}})}{\max(f_{\text{metric}}) - \min(f_{\text{metric}})} \tag{4}$$

Then we integrate the two metrics and take the *L2*-norm of the to generate the final difficulty score as in Eq. (5). As a result, the more common for a sample, the smaller the *L2*-norm value, and the easier it is. Besides, we add a hyperparameter λ to balance the influence of the two metrics.

$$d_{common}(\boldsymbol{x}_i) = \left\| < f_{\max}(\boldsymbol{x}_i),\ \lambda f_{\text{complex}}(\boldsymbol{x}_i) > \right\|_2 \tag{5}$$

Similar to the previous single difficulty-based curricula, the commonness difficulty score only exists when there are some entities mentioned in the sample. We adopt the same remedy and randomly intersperse those non-entity samples among the ordered ones which contain entities.

4.2 Pacing Function

As for the pacing function, we use the root-based pacing function introduced by Platanios et al. (2019) in all our experiments, as in Eq. (6).

$$p(t) = \sqrt{t \cdot \frac{1 - p(0)^2}{T} + p(0)^2} \tag{6}$$

Here $p(0)$ defines the proportion of samples we feed our model at the very beginning; T is the number of epochs that we apply curriculum learning to our model.

Models	Entity Type in HarveyNER				
	Point	**Area**	**Road**	**River**	**Micro-Average**
NCRF++	71.43 / 72.26 / 71.85	66.00 / 61.68 / 63.77	**77.39 / 77.93 / 77.66**	61.40 / 44.56 / 51.64	68.69 / 65.16 / 66.88
+ Max	**72.55** / 71.51 / **72.03**	65.90 / 65.54 / 65.72	75.30 / **77.93** / 76.59	62.42 / 44.56 / 52.00	69.06 / 66.40 / 67.70
+ Complex	70.47 / 72.08 / 71.26	66.07 / 64.16 / 65.10	74.67 / 75.17 / 74.92	63.50 / 44.56 / 52.37	68.34 / 65.92 / 67.11
+ Commonness	71.40 / **72.64** / 72.02	**68.27 / 65.84 / 67.03**	77.23 / 77.24 / 77.24	**66.68 / 45.96 / 54.42**	**70.09 / 67.12 / 68.57**
BERT	71.55 / 73.11 / 72.32	62.04 / **72.87** / 67.02	76.42 / 82.07 / 79.15	62.11 / 55.09 / 58.39	66.62 / 71.48 / 68.97
+ Max	72.14 / 72.74 / 72.44	62.49 / 72.67 / **67.20**	77.83 / 80.69 / 79.23	57.92 / 56.14 / 57.02	66.73 / 71.28 / 68.93
+ Complex	70.41 / **75.47** / 72.85	62.32 / 72.87 / 67.19	76.12 / **82.76 / 79.30**	59.92 / 55.09 / 57.40	66.13 / **72.52** / 69.18
+ Commonness	**72.98** / 73.87 / **73.42**	**62.53** / 71.98 / 66.92	**79.20** / 78.62 / 78.91	**63.55 / 60.00 / 61.72**	**67.66** / 71.80 / **69.67**

Table 5: Evaluation on the test set, P / R / F1 (Precision / Recall / F1-Score,%)[4]. Since we use the same pacing function, we use the scoring function names as the curriculum names. We apply the normal curriculum setting to the NCRF++ model and the anti-curriculum setting to BERT model.

5 Experiments

In our experiments, we use two state-of-the-art NER systems as baselines and evaluate their performance on the HarveyNER dataset. And then we test the effectiveness of the designed curricula by adding them to the baseline systems.

5.1 Baselines

NCRF++ (Yang and Zhang, 2018) is an open-source Neural Sequence Labelling Toolkit. We use the BiLSTM-CNN-CRF structrue as a baseline.
BERT (Devlin et al., 2019) is a pretrained language model based on Transformer (Vaswani et al., 2017), which has largely improved many NLP tasks including NER. We fine-tune the *base-uncased* version for experiments.

5.2 Training Setup

For the **NCRF++** model, we use the *tweet-based* version Glove as word embeddings and keep all other hyper-parameters as default. For the **BERT** model, we test with some recommended hyper-parameters and use the set-up (learning rate as 5e-5 and batch size as 32) that performs best with the baseline model. As for the λ hyperparameter in Eq. (5), we choose 1 for the NCRF++ model and 0.6 for the BERT model after some searching. We train all the NCRF++ models 100 epochs and all the BERT model 50 epochs.

For a fair comparison, we keep all the training parameters the same when adding the curriculum arrangements. For the **NCRF++** model, we use the normal curriculum setting and feed easier cases first and for the **BERT** model, we use the anti-curriculum setting (more explanations can be found in 5.5). Besides, we train all the experiments five times using different random seeds to alleviate random turbulence.

5.3 Results

The experimental results are shown in Table 5. We can see that the best performed baseline BERT achieves *69.67%* F1 score, which is much lower than the BERT-base performance on CoNLL-2003 (*92.4%* (Devlin et al., 2019)). This illustrates the difficulty of the dataset.

Regarding the effectiveness of the curricula, we can easily see that almost all three curriculum arrangements (except **Max** with BERT) bring performance gains on both of the baselines. Our proposed **Common** curriculum added to both of the models performs the best across all the settings.

Specifically, for the NCRF++ model, the **Common** curriculum performs best and increases the baseline about *1.69%* (*68.57%* vs. *66.88%*) on average. Other proposed **Max** curriculum also performs well and improves the baseline by *0.82%* (*67.70%* vs. *66.88%*). The **Complex** curriculum marginally improves the baseline by *0.23%*.

As for the BERT model, our proposed **Common** curriculum is the most effective one and increases the baseline about *0.7%* F1 score (*69.67%* vs. *68.97%*) on average. Besides, the **Complex** curriculum also improves the baseline by *0.21%*.

5.4 How are the Difficult Samples Learned?

In order to analyze how the models have learned the difficult samples from the curricula, we divide the test set into "easy" and "hard" subsets based on their characteristic difficulties. First, we only keep those entity-contained samples in the test set since the difficulty scores are determined by the entities. For the difficulty caused by entity length, we set threshold values to partition them into the "short" test set and "long" test set; the "short" test set has an entity length range from 1 to 4, and the "long"

[4]All results are the average of 5 system runs.

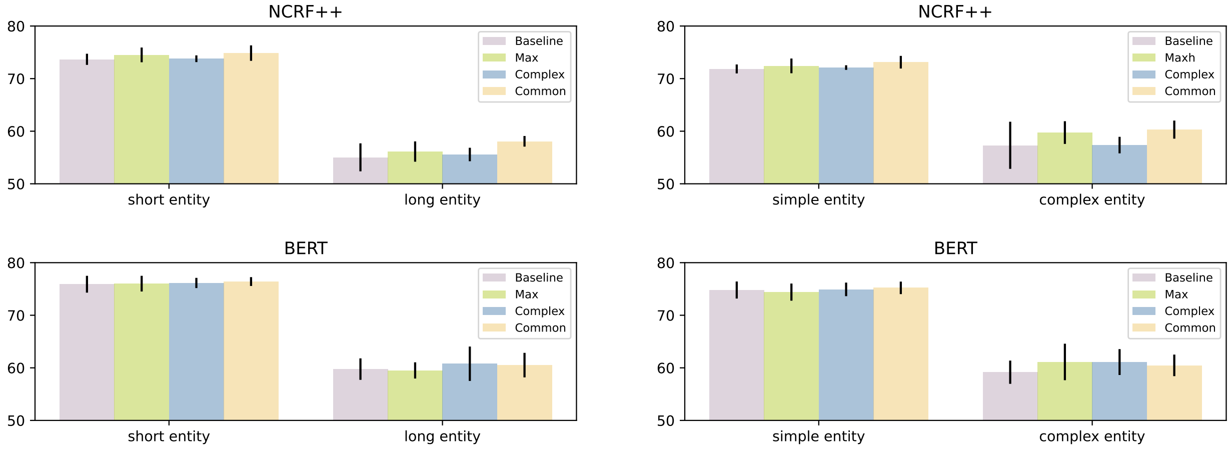

Figure 3: Test results on "easy" and "hard" subsets, F1-score,%.

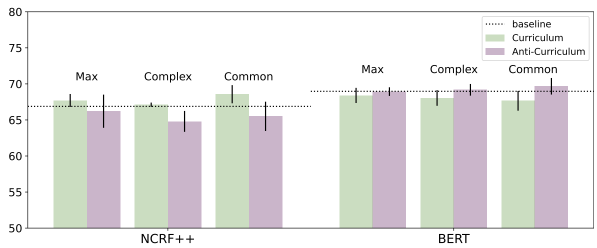

Figure 4: Curriculum v.s. Anti-curriculum, F1-score,%.

test set only contains samples with maximum entity length larger than 4.

As for the difficulty caused by complex entities, we just simply throw the samples into our "complex" entity set if there exists a complex indicator in its entities. The rest of the entity-contained samples are viewed as the "simple" entity set.

We test all our settings on the four subsets. As illustrated in Figure 3, in most cases, adding curricula achieve better performance than the baseline on both the "easy" sets and the "hard" sets for both the NCRF++ and BERT models.

5.5 Curriculum v.s Anti-curriculum

Apart from the different difficulty metrics, we find that applying different curriculum settings (normal curriculum that exposes easier examples early or anti-learning showing the most difficult examples first) will also result in a huge performance difference between the NCRF++ and the BERT models. As shown in Figure 4, for the neural network-based NCRF++ model, the normal curriculum setting has saliently better F-1 scores on average across all the three curriculum scoring functions in comparison with the anti-curriculum setting. But for the pretrained language model based on BERT, the results are the opposite; here using anti-curriculum learning will consistently give better performance than using normal curriculum learning.

One possible reason is that the volatile gradients from the anti-curriculum can lead to better local minima for a well pretrained model. As we know, the anti-curriculum learning will feed those "hard" samples to the model first, and the gradients from those long-tailed hard cases will have a relatively larger degree of fluctuations compared to that of easy instances. BERT is a pretrained language model and the pretrained parameters might constrain the model to some local regions. The fluctuations provided by the "hard" samples from the anti-curriculum learning can enable the BERT model to reach other better local minimal regions.

6 Conclusion

In this work, we propose a fine-grained location extraction dataset HarveyNER for facilitating local disaster response systems. This dataset contains many long and complex location mentions and state-of-the-art NER systems are far from addressing these hard cases. Based on the characteristic difficulty of the dataset, we propose two heuristic curriculum learning strategies and a novel commonness-based curriculum strategy to address the difficult cases. Empirical results demonstrate the effectiveness of our approaches. However, these hard cases are still far from being solved. Future work may consider using external knowledge to better identify the long and complex entities.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Pei Chen, Haibo Ding, Jun Araki, and Ruihong Huang. 2021. Explicitly capturing relations between entity mentions via graph neural networks for domain-specific named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 735–742, Online. Association for Computational Linguistics.

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

Zongcheng Ji, Aixin Sun, Gao Cong, and Jialong Han. 2016. Joint recognition and linking of fine-grained locations from tweets. In *Proceedings of the 25th international conference on world wide web*, pages 1271–1281.

Sarthak Khanal and Doina Caragea. 2021. Multi-task learning to enable location mention identification in the early hours of a crisis event. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4051–4056.

Sarthak Khanal, Maria Traskowsky, and Doina Caragea. 2021. Identification of fine-grained location mentions in crisis tweets. *arXiv preprint arXiv:2111.06334*.

Chenliang Li and Aixin Sun. 2014. Fine-grained location extraction from tweets with temporal awareness. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 43–52.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum learning for natural answer generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4223–4229. International Joint Conferences on Artificial Intelligence Organization.

Fenglin Liu, Shen Ge, and Xian Wu. 2021. Competence-based multimodal curriculum learning for medical report generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3001–3012, Online. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Stuart E Middleton, Lee Middleton, and Stefano Modafferi. 2013. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, 29(2):9–17.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Yixuan Su, Deng Cai, Qingyu Zhou, Zibo Lin, Simon Baker, Yunbo Cao, Shuming Shi, Nigel Collier, and Yan Wang. 2021. Dialogue response selection with hierarchical curriculum learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1740–1751, Online. Association for Computational Linguistics.

Erik F. Tjong Kim Sang 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Wenlin Yao, Cheng Zhang, Shiva Saravanan, Ruihong Huang, and Ali Mostafavi. 2020. Weakly-supervised fine-grained event recognition on social media texts for disaster management. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 532–539. AAAI Press.

Mingliang Zhang, Fandong Meng, Yunhai Tong, and Jie Zhou. 2021. Competence-based curriculum learning for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2481–2493, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 Annotation Guidelines

- 1. Location types can be "Area", "Point", "Road", and "River."
  - "Area" refers to all the named entities of cities, neighborhoods, super neighborhoods, geographic divisions etc.
  - "Point" refers to a location that is a building, a landmark, an intersection of two roads, an intersection of a river with a lake/reservoir/ocean, or a specific address.
  - "Road" refers to a road/avenue/street or a section of a road/avenue/street when the tweet does not provide an exact location among that road.
  - "River" refers to a river or a section of a river when the tweet does not imply there is an intersection between the river and other places.
- 2. A section of a road/river between two detailed/precise locations should be considered as a point. However, if the distance between the two points is very large, it might be considered as a stretch of a road/river.
- 3. A road passing through a small area can be designated as a point. A road intersecting a very large area cannot be a point and must be denoted as a stretch of a road. In some peculiar cases, the road takes a small detour and tangentially brushes off an area – in such specific cases, roads can be annotated as a point.
- 4. For the following locations, *Lake Houston*, *Barker Reservoir*, and *Addick's Reservoir* are annotated as areas while all other lakes/reservoirs are considered as points.
- 5. Ignore generic company/franchise names like HEB, Kroger etc. unless it is accompanied with a precise location, for example, *HEB at Kirkwood Drive*. However, non-franchised small businesses with only one unique location are considered as a point.
- 6. Ignore any locations in the Twitter username, like @HoustonABC. However, if the @ does not refer to a Twitter account name, please recognize the location. For example, *I am @ XXX High School*, "XXX High School" will be considered as a point.
- 7. For abbreviations or vague location names, always look up the tweet's context (or even other tweets' context) to decide if it is a location or not. We will use search engine if it is necessary.
  - Eg: *Coke Ck*; Here, "Ck" refers to a creek. This is understood when multiple such tweets point towards a creek.
- 8. Similarly, for names that can refer to different or multiple locations, like "Bellaire" can either refer to Bellaire St or the Bellaire area, we always look up the tweet's context to decide their location types.
- 9. We annotate the mentioned location as the complete set of phrases that describes the detail of the location including the core noun and all defining relative clauses. If a tweet mentioned the same location multiple times,

they will be annotated as multiple location mentions.

- 10. Ignore the location that **only** contains "Houston", "Harris County", or "Texas"

- 11. Ignore any tweet outside Houston (like London, Dallas, etc) and all non-English tweets.

- 12. We keep the exact words in tweet context as the location name after extracting the entities.

A.2 Examples of Annotation Disagreement

Examples are showed in Table 6.

No.	Tweets Body																	
1	Tropical	Storm	Warning	for	Liberty	,	Harris	,	Chambers	,	Jackson	,	Matagorda	,	Brazoria	and	Galveston	County
	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
	O	O	O	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	I-area
	O	O	O	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	I-area
	O	O	O	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	O	B-area	O	O	O
2	RT	@nyuudle	:	Buffalo	Bayou	(	I-45	and	Memorial	Drive	)	progression	in	Houston				
	O	O	O	B-point	I-point	I-point	I-point	I-point	I-point	I-point	I-point	O	O	O				
	O	O	O	B-river	I-river	O	B-point	I-point	I-point	I-point	O	O	O	O				
	O	O	O	B-river	I-river	O	B-point	I-point	I-point	I-point	O	O	O	O				
	O	O	O	B-river	I-river	O	B-area	I-area	I-area	I-area	O	O	O	O				
3	If	you	need	to	evacuate	from	Conroe	,	take	FM1097	between	I-45	to	149	.	FM2854	is	closed
	O	O	O	O	O	O	B-area	O	O	B-point	I-point	I-point	I-point	I-point	O	B-road	O	O
	O	O	O	O	O	O	B-area	O	O	B-road	O	B-road	O	B-road	O	B-road	O	O
	O	O	O	O	O	O	B-area	O	O	B-point	I-point	I-point	I-point	I-point	O	B-road	O	O
	O	O	O	O	O	O	B-area	O	O	B-road	O	B-road	O	B-road	O	B-road	O	O
4	Our	GF	N	FRWY	&amp	;	GF	Grand	Parkway	locations	are	open	for	those	in	need	.	
	O	O	B-road	I-road	O	O	O	B-road	I-road	O	O	O	O	O	O	O	O	
	O	B-road	I-road	I-road	O	O	B-road	I-road	I-road	O	O	O	O	O	O	O	O	
	O	B-point	I-point	I-point	I-point	I-point	I-point	I-point	I-point	O	O	O	O	O	O	O	O	
	O	B-point	I-point	I-point	O	O	B-point	I-point	I-point	O	O	O	O	O	O	O	O	
5	Fire	Event	-	E031	-	Sikes	-	Sikes	St	-	00:52	-	https://t.co/twmyivTj5Q					
	O	O	O	O	O	B-road	O	B-road	I-road	O	O	O	O					
	O	O	O	O	O	O	O	B-road	I-road	O	O	O	O					
	O	O	O	O	O	B-point	I-point	I-point	I-point	O	O	O	O					
	O	O	O	O	O	O	O	B-road	I-road	O	O	O	O					

Table 6: BIO in is from annotator 1, BIO in is from annotator 2, BIO in is from annotator 3, and BIO in is the final annotation. The error analysis between each annotator shows that annotators are more likely to have a disagreement when the location entities may indicate a point.