Crossroads, Buildings and Neighborhoods: A Dataset for Fine-grained Location Recognition

Pei Chen[1] **Haotian Xu**[1] **Cheng Zhang**[2] **Ruihong Huang**[1]
[1] Department of Computer Science and Engineering, Texas A&M University
[2] Purdue University Northwest
{chenpei, hx105, huangrh}@tamu.edu
zhan4168@pnw.edu

Abstract

General domain Named Entity Recognition (NER) datasets like CoNLL-2003 mostly annotate coarse-grained *location* entities such as a country or a city. But many applications require identifying fine-grained locations from texts and mapping them precisely to geographic sites, e.g., a crossroad, an apartment building, or a grocery store. In this paper, we introduce a new 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. We built strong baseline models using Curriculum Learning and experimented with different heuristic curricula to better recognize difficult location mentions. Experimental results show that the simple curricula can improve the system's performance on hard cases and its overall performance, and outperform several other baseline systems. The dataset and the baseline models can be found at https://github.com/brickee/HarveyNER.

1 Introduction

The 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 annotated in many NER datasets, including CoNLL-2003 (Tjong Kim Sang, 2002) and OntoNotes 5.0 (Pradhan et al., 2013). However, these datasets contain mostly coarse-grained entities such as a continent (e.g., Europe), a country (e.g., the U.S.), or a city (e.g., London).

Many downstream applications require identifying fine-grained location entities from texts, such as an apartment building (e.g., Bayou Oaks) or a specific store (e.g., the HEB on Montrose), in order to locate the geographic places on a map, which is vital to identify actionable information (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.

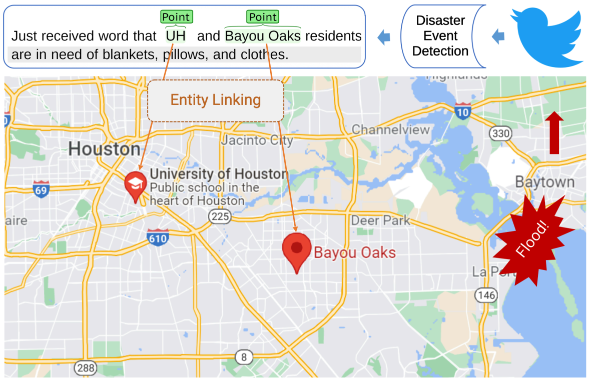
Just received word that UH and Bayou Oaks residents are in need of blankets, pillows, and clothes.

Figure 1: An example of a disaster response system.

Several previous works have attempted to create crisis-related datasets with fine-grained location mentions, either automatically (Middleton et al., 2013) or by manual (Khanal et al., 2021) annotations. However, these prior datasets contain many incomplete descriptions of locations that otherwise can be precisely projected to a map with certain geo-coordinates. 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* on the map, but previous work regard it as two *Road* mentions *"Richey St"* and *"W Harris Ave in Pasadena"*, and such incomplete location mentions will affect uses in many applications. We introduce

HarveyNER, the first dataset that annotates such coordinate-oriented location mentions.

We use the Harvey disaster in Houston as an example to demonstrate how to annotate such location mentions. Specifically, we consider tweets about Hurricane Harvey affecting the Houston metropolitan area in 2017 and annotate mentions of locations that exist in this city. Compared with the location mentions in prior NER datasets, HarveyNER focuses on the location mentions that can link to specific sites on a map. We carefully constructed the annotation guidelines and trained annotators to obtain high-quality annotations. We also built strong baselines over the dataset for future reference.

The unique characteristics of HarveyNER present challenges for NER systems. First, many location entities in this dataset are long and complex to precisely describe a place. E.g., the above example of a *Point* entity contains up to 11 words, and it could be wrongly recognized as two road entities by a NER system. Second, as an instant social medium, tweets contain many informal contents, local conventions, and even grammatical errors, which create many out-of-vocabulary (OOV) words that cannot be found in pretrained word embedding such as Glove (Pennington et al., 2014) or BERT (Devlin et al., 2019).

To improve the performance on these hard location mentions and build strong baselines for the HarveyNER dataset, we adopt Curriculum Learning (CL) (Bengio et al., 2009) to better learn difficulty samples by ordering examples during training based on their difficulty. We design two heuristic curricula based on entity length and word complexity considering that many long and complex entities in HarveyNER are naturally difficult (as shown in Figure 3, the performance of baseline systems is worse on these hard cases). We further assume that the difficulty to learn may not only depend on the inherent difficulty of a type of case but also depend on how commonly seen or how well represented such cases are in the dataset. Therefore, we propose a novel curriculum with a difficulty scoring function that comprehensively considers the two heuristic difficulty metrics as instance frequencies. Empirical results show that all of the curricula can outperform several other baseline systems, and our novel curriculum performs the best.

We also find that different NER-based systems benefit from different curriculum scheduling strategies. In our experiments, the normal curriculum (training with easier samples first) is suitable for training the neural network-based model NCRF++ without pretrained language models, while the anti-curriculum (training with harder samples first) facilitates fine-tuning of the pretrained language model BERT.

2 Related Work

NER research has a long history and many NER datasets have been created with certain pre-defined entity categories. General domain datasets such as CoNLL-2003 (Tjong Kim Sang, 2002) and OntoNotes 5.0 (Pradhan et al., 2013) attend to certain common entity types including *Location*. Location mentions in these datasets are mostly coarse-grained, e.g., the *U.S.* (a country) or *London* (a city). 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 contains incomplete location mentions and is of limited use for a disaster response system. In contrast, our dataset HarveyNER emphasizes 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) use Neural Network models like BiLSTM-CNN-CRF (Ma and Hovy, 2016) and contextual embeddings like BERT (Devlin et al., 2019), and 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 that HarveyNER contains many long and complex location mentions, we design corresponding curricula to learn them.

Data Split	Train	Valid	Test	Total
# of 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
# of Entity Mentions	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 the HarveyNER Dataset.

3 The HarveyNER Dataset

3.1 Data Preparation

Data Collection We used the Twitter PowerTrack API to retrieve the tweets posted during the time of peak disruption caused by Hurricane Harvey in the Houston area, specifically from 5:00 a.m. August 25 to 4:59 a.m. August 31, 2017. In total, we collected 1,121,363 tweets excluding retweets and replies.

Data Cleaning We applied several strategies to filter out irrelevant tweets. First, we only keep the tweets that are related to the Houston area, i.e., the geo-coordinates of the tweets or the authors' profile locations are within the bounding of Houston. Second, we applied our weakly-supervised event detection system (Yao et al., 2020) to identify tweets on disaster-related topics; these tweets are likely to be related to Hurricane Harvey during the specified period. We also manually filtered out remaining irrelevant tweets (such as non-English and repeated tweets) during the annotation process. In total, 6,571 tweets were selected for this study, as shown in Table 1.

3.2 Location Entity Annotation

Location 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.

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

kappa κ	A1 & A2	A1 & A3	A2 & A3	Average
All	85.64	82.17	83.12	83.64
Annotated	66.54	60.49	62.09	63.04

Table 2: Inter-Annotator Agreement (%) at token-level. *All* for all the tokens and *Annotated* for annotated tokens only. There are three annotators A1, A2 and A3.

Annotation Quality To train the annotators to well annotate the fine-grained location mentions, especially to distinguish the *Point* locations, we conduct rounds of initial annotation exercises and receptively update annotation guidelines to reduce ambiguity and subjectivity. The detailed guidelines can be found in Appendix A.1.

We trained three annotators and calculated their Inter-Annotator Agreement (IAA) based on 500 randomly selected tweets they all annotated. We pairwise calculate the Cohen's kappa (κ) scores based on the token-level annotations from each pair of annotators. As suggested by Brandsen et al. (2020), we report two scores: one calculated using all the token annotations and one only using the annotated tokens that exclude non-entity tokens. As shown in Table 2, we observe a high average κ score of 83.64% for all tokens and an average κ score of 63.04% for annotated tokens only. After that, the three annotators annotated the remaining tweets independently.

3.3 Dataset Analysis

We randomly split the annotated tweets into training, validation, and test sets for experiments with a ratio of 6:2:2. Table 1 shows some basic data statistics. We can see that 27.48% of the tweets contain at least one location entity mention, while the remaining tweets do not mention any location. As for location types, *Point* and *Area* are two dominant entity types covering 38.36% and 44.66% of entity mentions respectively, while *Road* and *River* only make up 10.22% and 6.76% of entity mentions respectively.

Comparisons with CoNLL-2003 We compare HarveyNER with CoNLL-2003, a general NER dataset annotated with coarse-grained locations in news articles, and Table 3 shows the comparisons in several aspects.

First of all, entities in HarveyNER are longer

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

Table 3: HarveyNER v.s. CoNLL-2003.

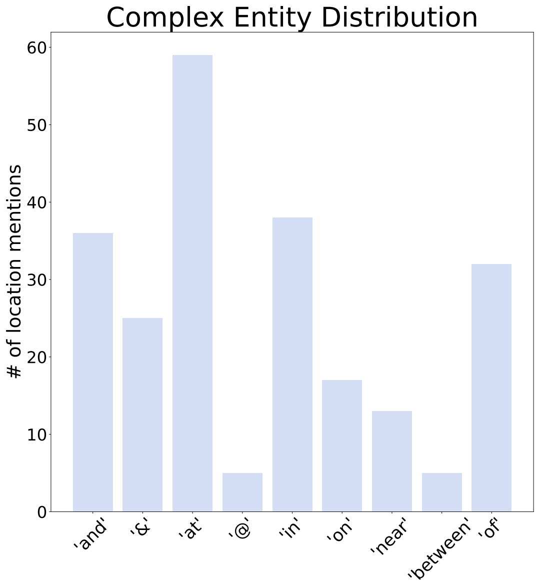

Figure 2: Number of Location Mentions with Each Complexity Indicator Word.

Indicators	Examples
"and"	the corner of Richey St **and** W Harris Ave in Pasadena (Point)
"&"	Beltway 8 **&** Tidwell (Point)
"at"	Brazos River **at** Richmond (River)
"@"	Copperfield Church **@** 8350 hwy 6 north (Point)
"in"	Constellation Field **in** Sugar Land (Point)
"on"	Chimney Rock **on** I-10 East (Point)
"near"	IH 10 **near** Monmouth (Point)
"between"	249 **between** Cypresswood / Louetta (Point)
"of"	0.25-0.5 north **of** I-10 (Point)

Table 4: Examples of complex entities.

on average at both word level and character level. We observed that many location mentions of the type *Point* and *Area* are especially long to precisely describe a site or area on a map. We manually analyzed long entities in the validation set and observed that many location mentions are complex noun phrases with a conjunction or a prepositional phrase attachment. We noted down two commonly seen conjunctions and seven commonly seen prepositions, and table 4 shows location examples for each. We calculate the percentage of complex entity mentions with one of these words, and HarveyNER has 11.8% of entity mentions fall in this category while CoNLL-2003 has few such entity mentions (0.19%). Figure 2 shows the number of tweets with each of the words.

Second, considering that the language used in tweets is informal and contains many abbreviations and even grammatical errors, we calculate the out-of-vocabulary (OOV) rates (in Table 3) for both 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. The high OOV rate might degrade the performance of NER systems relying on pretrained word embeddings.

In addition, we compare the average sentence length between the two datasets. To our surprise, HarveyNER has overall longer sentences, based on both word counts and character counts. This is counter-intuitive since the tweet content is strictly constrained to be no more than 140 characters each. One possible reason is that short tweets are less likely to provide useful event information and have been filtered out by the event detection system (Yao et al., 2020) we used.

Lastly, we measure the density of annotated location entities in the two datasets by calculating the average number of location entity mentions per sentence and calculating the percentage of entity words out of all the words in a sentence. We also calculate these two measures for the subset of sentences that contain at least one location mention (non-empty sentences). The last section of Table 3 shows the results. We can see that the two datasets are similar over these annotation density measures.

4 Curriculum Arrangement

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

- **Difficulty Scoring Function**: Given an input sample $\boldsymbol{x}_i$, this function map it to a numerical

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

score, $d(\boldsymbol{x}_i) \in \mathbb{R}$. The score is used to represent the difficulty level of the corresponding sample. 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.

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 $t = 1, 2, \ldots$ 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 $\mathcal{D}^{\text{train}}_{\text{sorted}}$;
```

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 previously, our HarveyNER dataset has longer entity mentions on average compared to CoNLL-2003, and this brings many long and difficult entities that are hard to identify. Intuitively, we can design a corresponding curriculum based on such an entity-level difficulty. Specifically, given an input tweet sample $\boldsymbol{x}_i$ that contains n words, $\boldsymbol{x}_i = \{w_1, w_2, \ldots, w_n\}$, and k ($k \geq 0$) entities, $\{E_1, E_2, \ldots, E_k\}$. $|E_j|$ represents the length of j-*th* entity, specifically, the number of words in the j-*th* entity. Now, we can assign each tweet sample a score using the maximum length[3] of its entities:

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

where, $L_i = \{|E_1|, |E_2|, \ldots |E_k|\}$, the set of entity lengths for the i-th sample $\boldsymbol{x}_i$.

With this scoring function, we need to pay attention to the tweets with zero entity mention (*72.52%* of tweets in HarveyNER as shown in Table 1) since their difficulty scores will all be 0. In this case, the algorithm will provide all these tweets 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 tweet samples. Specifically, when we order our dataset by the difficulty scores, we randomly intersperse those no-entity tweet samples among the ordered samples that have entities.

Complex Entity Rate (Complex): Corresponding to the analysis of the complex entity rate in HarveyNER, we define another difficulty scoring function. Specifically, we define the complexity of an entity $c(E)$ based on whether the entity contains one of the conjunction words or preposition words we identified and which word the entity contains. Heuristically, we assign a weight greater than 1 to these words, specifically, we assign a weight of 3 to each conjunction word and a weight of 2[4] to each preposition word to reflect our intuition that entities with conjunctions can be more difficult cases.

If an entity E contains more than one "complexity" indicator, we choose the one with the highest weight. For example, the entity example E *"the corner of Richey St and W Harris Ave in Pasadena"* contains the conjunction *"and"* and the preposition *"in"*, we deem the complexity of this entity $c(E)$ is 3 instead of 2. Then, one tweet sample x_i can have multiple entities with different complexities $C_i = \{c(E_1), c(E_2), \ldots, c(E_k)\}$, we assign the

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

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

[4] We further lower the weight for the preposition "of" to 1 considering that this is a very general preposition and is often observed in regular location entities as well, e.g., "University of Houston".

maximum entity complexity value as the complexity value of the tweet sample, i.e.,

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

However, if none of the entities in a tweet sample contain complexity indicators, this scoring function will assign 0 as the difficulty score for the tweet sample. A similar issue has been discussed for the previous scoring function, and we use the same remedy as well and randomly interspersed these zero-scored samples among the other ordered samples.

Commonness of Difficulty (Commonness): We further propose a novel difficulty metric that considers both of the prior metrics as well as their support. We exploit the assumption that the difficulty to learn may not only depend on the inherent difficulty of a type of case but also depend on how commonly seen or how well represented such cases are in the dataset.

Specifically, to assign a difficulty score for a tweet sample, we first count the number of training samples that have the same difficulty score as the sample $\boldsymbol{x}_i$ according to one of the prior two metrics and then divide it by the total number of tweet instances N. Then, we take the reciprocal of it and get f_{metric}.

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

Where, d_{metric} are the difficulty metrics d_{max} or d_{complex}. Hence, the larger support of a difficulty level based on one of the two prior heuristics, the lower the f_{metric} value is. We then normalize this value to the range of $[0,\ 1]$.

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

Then, we integrate f_{max} and f_{complex}, and take the *L2*-norm to generate the final difficulty score.

$$d_{common}(\boldsymbol{x}_i) = \sqrt{f_{\text{max}}(\boldsymbol{x}_i)^2 + (\lambda f_{\text{complex}}(\boldsymbol{x}_i))^2} \tag{5}$$

As a result, the more common a sample is concerning its length or complexity, the smaller the *L2*-norm value is, which indicates a lower difficulty based on the new metric. In addition, we add a hyperparameter λ to balance the influence of the two metrics.

Similar to the previous single difficulty-based curricula, the commonness difficulty score is zero when a tweet sample has no entity. We adopt the same remedy and randomly intersperse those no-entity tweet samples among the ordered ones that contain entities.

4.2 Pacing Function

We use the root-based pacing function introduced by Platanios et al. (2019) in all our experiments.

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

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.

5 Experiments

In our experiments, we use two state-of-the-art NER systems as base models and evaluate their performance on the HarveyNER dataset. Then, we test the effectiveness of the designed curricula by applying them to train the base models.

5.1 Baselines

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

5.2 Training Setup

For the **NCRF++** model, we use the *tweet-based* version Glove as word embeddings and keep all the other hyper-parameters as default. For the **BERT** model, we set learning rate as 5e-5 and set batch size as 32. As for the λ hyperparameter in Eq. (5), we used grid search and set it 1 and 0.6 for the NCRF++ model and the BERT model respectively. We train all the NCRF++ models for 100 epochs and train all the BERT models for 50 epochs.

For fair comparisons, we keep all the training parameters the same when conducting curriculum learning. For the **NCRF++** model, we use the normal curriculum setting and feed easier cases

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, Precision / Recall / F1-Score (Percentages)[5]. Since we use the same pacing function, we use the scoring functions to name the curricula. Note that we apply the normal curriculum setting to the NCRF++ model and apply the anti-curriculum setting to the BERT model.

first, while for the **BERT** model, we use the anti-curriculum setting (more explanations provided in Section 5.5). Note that we train all the experiments five times using different random seeds to alleviate random turbulence.

5.3 Results

Table 5 shows the experimental results. We can see that the BERT base model outperforms the NCRF++ base model consistently across the four location categories on this dataset. Curriculum learning yields further performance gains for both the BERT model and the NCRF++ model, this is true for all the curricula paired with both models except the **Max** curriculum when applied to the BERT model, where the average performance almost stays the same. Among the three curricula, he **Commonness** curriculum achieves the best performance for both models.

Conducting curriculum learning have unequal impacts on the four location categories. When using the NCRF++ model, curriculum learning yields a small performance improvement on *Point* and clear improvements on *Area* and *River*, while using the BERT model, curriculum learning yields a relatively larger improvement on *Point* and a clear improvement on *River* as well.

Note that the best-performed BERT model only achieves a micro-average F1-score of 69.67% on this dataset, which is still much lower than recently published BERT-base performance on CoNLL-2003 (e.g., 92.4% as reported in (Devlin et al., 2019)).) Meanwhile, we acknowledge that recognizing fine-grained locations is a trickier task than recognizing coarse-grained locations or many other general types of entities, even for humans as shown by the imperfect inter-annotator agreements.

5.4 How are the Difficult Samples Learned?

In order to understand if the models have better learned the difficult samples after applying curriculum learning, we divide the test set into "easy" and "hard" subsets based on either entity length or entity complexity and report experimental results on each subset. In this analysis, we only consider tweet samples in the test set that contains at least one entity mention since the groupings are determined by the characteristics of the entities. In the first grouping, we divide tweet samples into "short" and "long" groups depending on if the maximum entity length (number of words) in a tweet sample is greater than a threshold, four words in particular ($<= 4$ v.s. > 4). In the second grouping, we divide tweet samples into "simple" and "complex" groups by checking if a tweet sample contains a complex entity mention[6].

Across all the experimental settings, we report the results on two subsets separately under each grouping (Figure 3). We can see that curriculum learning indeed yields noticeable improvements in identifying those hard cases, while the improvements vary when adopting different curricula. Meanwhile, curriculum learning also achieves mild improvements on identifying the remaining relatively easy cases.

5.5 Curriculum v.s Anti-curriculum

We find that applying different curriculum settings (normal curriculum that exposes the easiest examples first or anti-curriculum that exposes the most difficult examples first) results in a large performance difference between the NCRF++ model and the BERT model. As shown in Figure 4, for the NCRF++ model without pretrained language mod-

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

[6] A complex entity mention has one of the conjunctions or prepositions we identified.

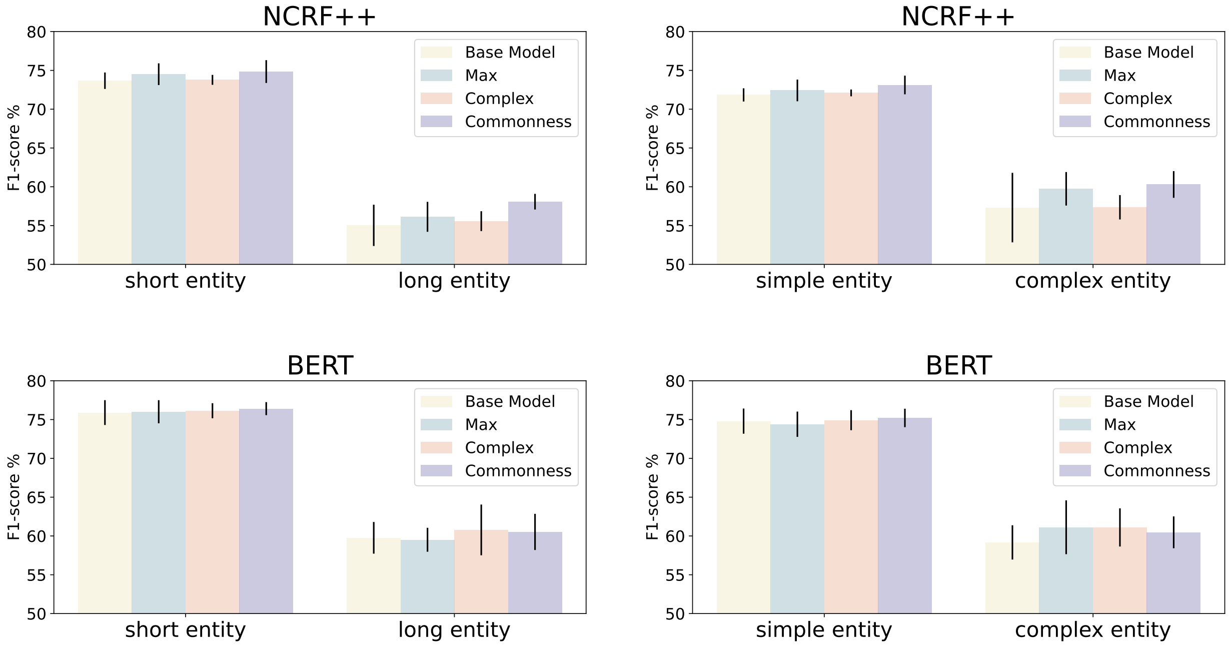

Figure 3: Results on "easy" and "hard" subsets of the test data, F1-score (Percentage).

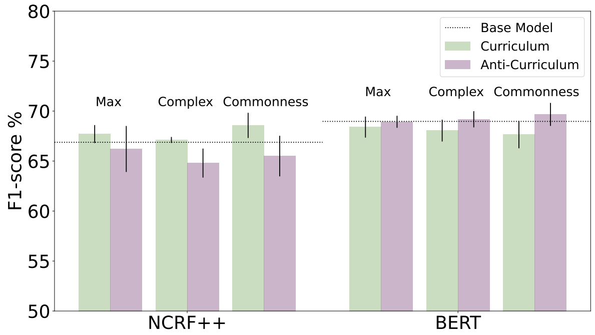

Figure 4: Curriculum v.s. Anti-curriculum, F1-score (Percentage).

els, the normal curriculum setting yields significantly better average F-1 scores across all the three curriculum scoring functions in comparison with the anti-curriculum setting. However, for the pretrained language model BERT, the results are the opposite; using anti-curriculum learning consistently yields better performance than using normal curriculum learning.

One possible explanation is that the volatile gradients resulting from using anti-curriculum learning can lead to better local minima for a well-pretrained model. Specifically, the anti-curriculum learning will feed those “hard” samples to the model first, and the gradients from those long-tailed hard cases will cause relatively large fluctuations compared to those from easy instances. BERT is a pretrained language model and the pretrained parameters might constrain the model to some local regions. The fluctuations produced by the “hard” samples from the anti-curriculum learning can enable the BERT model to reach other better local minima regions.

6 Conclusion

We introduce a fine-grained location recognition dataset, HarveyNER, to enable many downstream applications such as building real-time disaster response systems. This dataset contains many long and complex location mentions that feature interesting internal syntactic and semantic structures and the state-of-the-art NER systems are unable to fully recognize these hard cases. Considering the clear characteristics of difficult cases in this dataset, we experimented with two heuristic curriculum learning strategies and a novel commonness-based curriculum strategy to better recognize the difficult location mentions. Empirical results demonstrate the effectiveness of the curricula, which serve as strong baseline results in this dataset. Future work may consider incorporating external knowledge or innovations on system architectures to better identify fine-grained location mentions.

Acknowledgements

We gratefully acknowledge support from National Science Foundation via the awards IIS-1909252 and IIS-1759537. In addition, this research is partially supported by 2021 PRISE Grant 02-243712. We would also like to thank the anonymous reviewers for their valuable feedback and input.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Alex Brandsen, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. 2020. Creating a dataset for named entity recognition in the archaeology domain. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4573–4577, Marseille, France. European Language Resources Association.

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, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1271–1281. ACM.

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 *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pages 43–52. ACM.

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 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4223–4229. ijcai.org.

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 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, Doha, Qatar. Association for Computational Linguistics.

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. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 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 ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia. 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 on 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. The following locations, *Lake Houston*, *Barker Reservoir*, and *Addick’s Reservoir*, are annotated as areas due to their significant size 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.