Collective Learning From Diverse Datasets for Entity Typing in the Wild

Abhishek
abhishek.abhishek@iitg.ac.in
Indian Institute of Technology Guwahati
Guwahati, Assam, India

Amar Prakash Azad
Balaji Ganesan
amarazad@in.ibm.com
bganesa1@in.ibm.com
IBM Research Lab
India

Ashish Anand
Amit Awekar
anand.ashish@iitg.ac.in
awekar@iitg.ac.in
Indian Institute of Technology Guwahati
Guwahati, Assam, India

ABSTRACT

Entity typing (ET) is the problem of assigning labels to given entity mentions in a sentence. Existing works for ET require knowledge about the domain and target label set for a given test instance. ET in the absence of such knowledge is a novel problem that we address as ET in the wild. We hypothesize that the solution to this problem is to build supervised models that generalize better on the ET task as a whole, rather than a specific dataset. In this direction, we propose a Collective Learning Framework (CLF), which enables learning from diverse datasets in a unified way. The CLF first creates a unified hierarchical label set (UHLS) and a label mapping by aggregating label information from all available datasets. Then it builds a single neural network classifier using UHLS, label mapping and a partial loss function. The single classifier predicts the finest possible label across all available domains even though these labels may not be present in any domain-specific dataset. We also propose a set of evaluation schemes and metrics to evaluate the performance of models in this novel problem. Extensive experimentation on seven diverse real-world datasets demonstrates the efficacy of our CLF.

CCS CONCEPTS

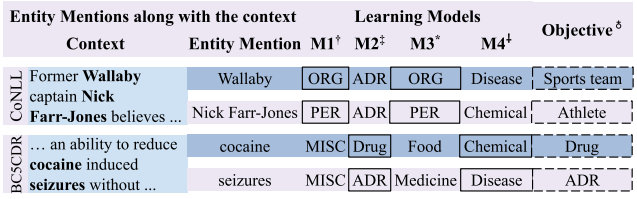
• **Computing methodologies** → **Natural language processing**; *Machine learning*.

KEYWORDS

entity typing, hierarchy creation, learning from multiple datasets

1 INTRODUCTION

Evolution of ET has led to the generation of multiple datasets. These datasets differ from each other in terms of their domain or label set or both. Here, the domain of a dataset represents the data distribution of its sentences. The label set represents the entity types annotated. Existing work for ET requires knowledge of the domain and the target label of a test instance [22]. Figure 1 illustrates this issue where four learning models are typing four entity mentions. We can observe that, in order to make a reasonable prediction (output with a solid border), it is required to assign labels from a model which has been trained on a dataset with similar domain and labels as that of test instances. However, domain and target label information of a test instance is unknown in several NLP applications such as entity ranking for web question answering systems [6] and knowledge base completion [7], where ET models are used.



†Model 1: Training dataset is CoNLL. Labels = {PER, ORG, LOC, MISC}.
‡Model 2: Training dataset is CADEC, Labels = {Drug, Adverse Drug Reaction (ADR), … }.
*Model 3: Training dataset is Wiki, Labels = {PER, Athlete, Sports team, food, medicine, … }.
ǂModel 4: Training dataset is BC5CDR, Labels = {Chemical, Disease}.
δ Objective of this work is to predict the finest-possible label irrespective of the dataset.

Figure 1: The output of four learning models on typing four entity mentions. For example, the model M1 trained on CoNLL dataset assigned the type ORG to the entity mention Wallaby, from the same dataset.

We address ET in the absence of domain and target label set knowledge as ET in the wild problem. As a result, we have to predict the best possible labels for all test instances as illustrated in Figure 1 (output with dashed line border). These labels may not be present in the same domain dataset. For example, the prediction of the label *sports team* for the entity mention Wallaby, when the best possible fine-grained label (*sports team*) is not present in the same domain CoNLL dataset [25]. We hypothesize that the solution to this problem is to build supervised models that generalize better on the ET task as a whole, rather than a specific dataset. This solution requires collective learning from several diverse datasets.

However, collectively learning from diverse datasets is a challenging problem. Figure 2 illustrates the diversity of seven ET datasets. We can observe that every dataset provides some distinct information for the ET task such as domain and labels. For example, CADEC dataset [11] contains informally written sentences from a medical forum, whereas JNLPBA dataset [12] contains formally written sentences from scientific abstracts in life sciences. Moreover, there is an overlap in the label sets as well as a relation between the labels of these datasets. For example, both CoNLL and Wiki [14] datasets have a label *person*. However, only Wiki dataset has a label *athlete*, a subtype of *person*. This means that CoNLL dataset can also contain *athlete* mentions but were only annotated with a coarse label *person*. Thus, learning collectively from these diverse datasets

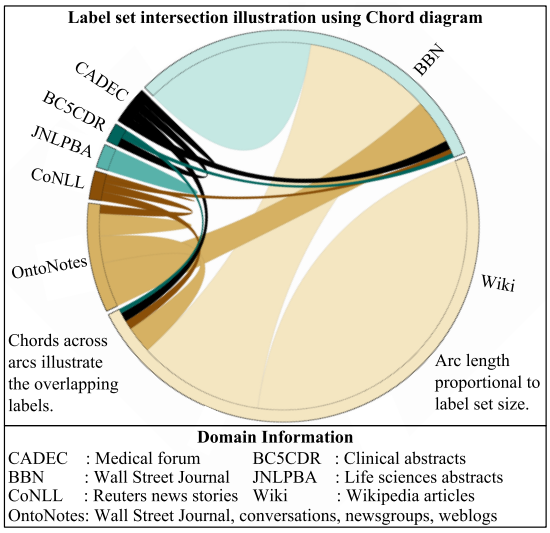


Figure 2: Illustration of the diversity of the seven ET datasets in the label set and domain.

require models to learn a useful feature or representation of the sentences from diverse domains as well as to learn the relation among labels.

This study proposes a collective learning framework for the ET in the wild problem. CLF first builds a unified hierarchical label set (UHLS) and associated label mapping by pooling labels from diverse datasets. Then, a single classifier[1] collectively learns from the pooled dataset using UHLS, label mapping and a partial hierarchy aware loss function.

In the UHLS, the nodes are contributed by different datasets, and a parent-child relation among nodes translate to a coarse-fine label relation. During construction of UHLS, a mapping from every dataset specific label to the UHLS nodes is also constructed. We expect to have one-to-many mappings, as in the case of real-world datasets. For example, a coarse-grained label for a dataset could be mapped to multiple nodes in the UHLS introduced by some other dataset. During the UHLS construction, human judgment is used when comparing two labels. This effort is four orders of magnitude lesser compared to annotating every dataset with fine-grained labels.

Utilizing the UHLS and the mapping, we can view several domain-specific datasets as a collection of a multi-domain dataset having the same label set. On this combined dataset, we use an LSTM [10] based encoder to learn a useful representation of the text followed by a partial hierarchical loss function [29] for label classification. This setup enables a single neural network classifier to predict fine-grained labels across all domains, even though the fine-grained label was not present in any in-domain dataset.

We also propose a set of evaluation schemes and metrics for the ET in the wild problem. In our evaluation schemes, we evaluate learning models performance on a test set which is formed by merging test instances of seven diverse datasets. To excel on this

[1]We used the term single classifier to denote a learning model with a single classification head being trained on multiple datasets with different labels together.

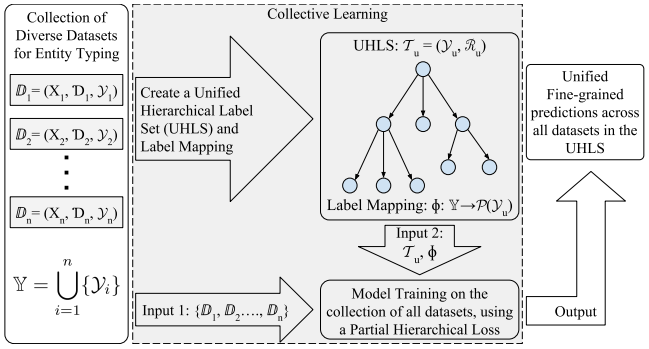


Figure 3: An overview of the proposed collective learning framework.

merged test set, learning models must generalize beyond a single dataset. Our evaluation metrics are designed to measure learning models performance to predict the best possible fine-grained label. We compared a single classifier model trained with our proposed framework with an ensemble of various models. Our model outperforms competitive baselines with a significant margin.

Our contributions can be highlighted as below:

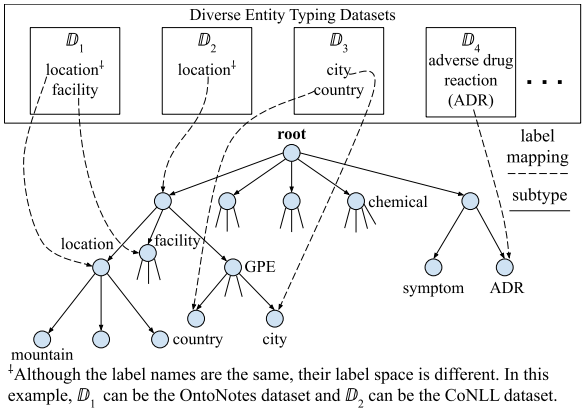
(1) We propose a novel problem of ET in the wild with the objective of building better generalizable ET models (§ 2).
(2) We propose a novel collective learning framework which makes it possible to train a single classifier on an amalgam of diverse ET datasets, enabling fine-grained prediction across all the datasets, i.e., a generalized model for ET task as a whole (§ 3).
(3) We propose evaluation schemes and evaluation metrics to compare learning models for the ET in the wild problem setting (§ 4.5, 4.6).

2 TERMINOLOGIES AND PROBLEM DEFINITION

In this section, we formally define the ET in the wild problem and related terminologies.

Dataset: A dataset, $\mathbb{D}$, is a collection of $(X, \mathcal{D}, \mathcal{Y})$. Here, X corresponds to a corpus of sentences with entity boundaries annotated, $\mathcal{D}$ corresponds to the domain and $\mathcal{Y} = \{y_1, \ldots y_n\}$ is the set of labels used to annotate each entity mention in the X. It is possible that two datasets share domain but differ in their label sets or vice versa. Here the domain means the data characteristics such as writing style and vocabulary. For example, sentences in the CoNLL dataset are sampled from Reuters news stories around 1999, whereas, sentences in the CADEC dataset are from medical forum posts around 2015. Thus, these datasets have different domains.

Label space: A label space $\mathcal{L}$ for a particular label y, is defined as a set of entities that can be assigned a label y. For example, the label space for a label *car* includes mentions of all cars including that of label space of differet car types such as *hatchback*, *SUV* etc. For different datasets, even if two labels with the same name exist, their label space can be different. The label space information is defined in the annotation guidelines used to create datasets.



†Although the label names are the same, their label space is different. In this example, $\mathbb{D}_1$ can be the OntoNotes dataset and $\mathbb{D}_2$ can be the CoNLL dataset.

Figure 4: A simplified illustration of the UHLS and the label mapping from individual datasets.

Type Hierarchy: A type or label hierarchy, $\mathcal{T}$, is a natural way to organize label set in a hierarchy. It is formally defined as $(\mathcal{Y}, \mathcal{R})$, where $\mathcal{Y}$ is the type set and $\mathcal{R} = \{(y_i, y_j) \mid y_i, y_j \in \mathcal{Y} \;\&\; i \neq j \;\&\; \mathcal{L}(y_i) \prec \mathcal{L}(y_j)\}$ is the relation set, in which (y_i, y_j) means that y_i is a subtype of y_j or in other words the label space of y_i is subsumed within the label space of y_j.

ET in the Wild problem definition Given n datasets, $\mathbb{D}_1, \ldots, \mathbb{D}_n$, each having its own domain and label set, $\mathcal{D}_i$ and $\mathcal{Y}_i$ respectively, the objective is to predict the best possible fine-grained label from the set of all labels, $\mathbb{Y} = \bigcup_{i=1}^{n} \{\mathcal{Y}_i\}$, for a test entity mention. The fine-grained label might not be present in any in-domain dataset.

3 COLLECTIVE LEARNING FRAMEWORK (CLF)

Figure 3 provides a complete overview of the CLF, which is based on the following key observations and ideas:

(1) From the set of all available labels $\mathbb{Y}$, it is possible to construct a type hierarchy $\mathcal{T}_u = (\mathcal{Y}_u, \mathcal{R}_u)$ where $\mathcal{Y}_u \subseteq \mathbb{Y}$ (§ 3.1).
(2) We can map each $y \in \mathbb{Y}$, to one or more than one node in $\mathcal{T}_u$, such that the $\mathcal{L}(y)$ is same as the label space of the union of the mapped nodes (§ 3.1).
(3) Using the above hierarchy and mapping, now even if for some datasets we only have the coarse labels, i.e., the labels which are mapped to non-leaf nodes, a learning model with a partial hierarchy aware loss function can predict fine labels (§ 3.2.2, 3.2.3).

3.1 Unified Hierarchy Label Set and Label Mapping

The labels of entity mentions can be arranged in a hierarchy. For example, the label space of *airports* is subsumed in the label space of *facilities*. In literature, several hierarchies, such as WordNet [16] and ConceptNet [15] exists. Even two ET datasets, BBN [27] and Wiki organize labels in a hierarchy. However, none of these hierarchies can be directly used as discussed next.

Data: $\mathbb{Y} = \bigcup_{i=1}^{n} \mathcal{Y}_i$
Result: Unified Hierarchical Label Set (UHLS), $\mathcal{T}_u = (\mathcal{Y}_u, \mathcal{R}_u)$ and label mapping, ϕ.

1 Initialize: $\mathcal{Y}_u = \{root\}, \mathcal{R}_u = \{\}$
2 **for** $y \in \mathbb{Y}$ **do**
3 **if** $\exists \mathcal{S} \subseteq \mathcal{Y}_u \; s.t. \; \mathcal{L}(y) == \mathcal{L}(\mathcal{S})$ **then** // Case 2
4 $\phi(y) \mapsto \mathcal{S}$
5 **else** // Case 1
6 $v = \underset{size(\mathcal{L}(v))}{\arg\min} \{v \mid v \in \mathcal{Y}_u \;\&\; \mathcal{L}(y) \prec \mathcal{L}(v)\}$
7 $\mathcal{Y}_u = \mathcal{Y}_u \cup \{y\}$
8 $\mathcal{R}_u = \mathcal{R}_u \cup \{(y, v)\}$
9 $\phi(y) \mapsto y$
10 **for** $(x, v) \in \mathcal{R}_u$ **do** // Update existing nodes
11 **if** $x \neq y \;\&\; \mathcal{L}(x) \prec \mathcal{L}(y)$ **then**
12 $\mathcal{R}_u = \mathcal{R}_u - \{(x, v)\}$
13 $\mathcal{R}_u = \mathcal{R}_u \cup \{(x, y)\}$
14 **for** $\hat{v} \in \mathcal{Y}_u$ **do** // Restrict to tree hierarchy
15 **if** $\mathcal{L}(\hat{v}) \prec \mathcal{L}(y) \;\&\; \hat{v} \notin subtree(y)$ **then**
16 $\phi(y) \mapsto \hat{v}$

Algorithm 1: UHLS and label mapping creation algorithm.

Our analysis of the labels of several ET datasets suggests that the presence of the same label name in the two or more datasets may not necessarily imply that their label spaces are same. For example, in the CoNLL dataset, the label space for the label *location* includes facilities, whereas in the OntoNotes dataset [28] the *location* label space excludes facilities. These differences are because these datasets were created by different organizations, at different times and for a different objective. Figure 4 illustrates this label space interaction. Additionally, some of these labels are very specific to the domains, and not all of them are present in any publicly available hierarchies such as WordNet, ConceptNet or even knowledge bases (Freebase [2] or WikiData [26]).

Thus, to construct UHLS, we analyzed the annotation guidelines of several datasets and came up with an algorithm formally described in Algorithm 1 and explained below.

Given the set of all labels, $\mathbb{Y}$, the goal is to construct a type hierarchy, $\mathcal{T}_u = (\mathcal{Y}_u, \mathcal{R}_u)$ and a label mapping $\phi : \mathbb{Y} \mapsto \mathcal{P}(\mathcal{Y}_u)$. Here, $\mathcal{Y}_u$ is the set of labels present in the hierarchy, $\mathcal{R}_u$ is the relation set and $\mathcal{P}(\mathcal{Y}_u)$ is the power set of the label set. To construct $\mathcal{T}_u$, we start with an initial type hierarchy, which can be $\mathcal{Y}_u = \{root\}, \mathcal{R}_u = \{\}$ or initialized by any existing hierarchy. We keep on processing each label $y \in \mathbb{Y}$ and decide if there is a need to update $\mathcal{T}_u$ and update the mapping ϕ. For each label y there are only two possible cases, either $\mathcal{T}_u$ is updated or not.

Case 1, $\mathcal{T}_u$ is updated: In this case y is added to a child of an existing node in the $\mathcal{T}_u$, say v. While updating $\mathcal{T}_u$ it is ensured that $v = \underset{size(\mathcal{L}(v))}{\arg\min} \{v \mid v \in \mathcal{Y}_u \;\&\; \mathcal{L}(y) \prec \mathcal{L}(v)\}$, i.e., $\mathcal{L}(v)$ is the smallest possible label space that completely subsumes the label space of y (lines 6-8). After the update, if there are existing subtrees rooted at v, then if the label space of y subsumes any of the subtree

space, then y becomes the root of those subtrees (lines 10-13). In this case the label mapping is updated as $\phi(y) \mapsto y$, i.e., the label in an individual dataset is mapped to a same label name in UHLS. Additionally, if there exist any other nodes, $\hat{v} \in \mathcal{Y}_u \; s.t. \; \mathcal{L}(\hat{v}) \prec \mathcal{L}(y)$ & $\hat{v} \notin subtree(y)$, we add $\phi(y) \mapsto \hat{v}$ for all such nodes (lines 14-16). This additional condition ensures that even in the cases where the actual hierarchy will be a directed acyclic graph, we restrict it to a tree hierarchy by adding additional mappings.

Case 2, $\mathcal{T}_u$ is not updated: In this case, $\exists \mathcal{S} \subseteq \mathcal{Y} \; s.t. \; \mathcal{L}(y) == \mathcal{L}(\mathcal{S})$, i.e, there exists a subset of nodes whose union of label space is equal to the label space of y. If $|\mathcal{S}| > 1$, intuitively this means that the label space of y is a mixed space, and from some other datasets labels with finer label spaces were added to $\mathcal{Y}_u$. If $|\mathcal{S}| = 1$, this means that some other dataset added a label which has the same label space. In this case we will only update the label mapping as $\phi(y) \mapsto \mathcal{S}$ (lines 3-4).

In Algorithm 1 whenever a decision has to be made related to a comparison between two label spaces, we refer a domain expert. The expert makes the decision based on the annotation guidelines for the queried labels and using existing organization of the queried label space in WordNet or Freebase if the queried labels are present in these resources. We argue that since the overall size of $\mathbb{Y}$ is several order of magnitude less than the size of annotated instances ($\approx 250 << \approx 3 \times 10^6$), having a human in the loop preserves the overall semantic property of the tree, which will be exploited by a partial loss function to enable fine-grained prediction across domains. An illustration of UHLS and label mapping is provided in Figure 4.

In the next section, we will describe how the UHLS and the label mapping will be used by a learning model to make finest possible predictions across datasets.

3.2 Learning Model

Our learning model can be decomposed into two parts: (1) Neural Mention and Context Encoders to encode the entity mention and its surrounding context into a feature vector; (2) Unified Type Predictor to infer entity types in the UHLS.

3.2.1 Neural Mention and Context Encoder. The input to our model is a sentence with the start and end index of entity mentions. Following the work of [1, 24, 29] we use Bi-directional LSTMs [8] to encode left and right context surrounding the entity mention and use a character level LSTM to encode the entity mention. After this we concatenate the output of the three encoders, to generate a single representation (R) for the input.

3.2.2 Unified Type Predictor. Given the input representation, R, the objective of the predictor is to assign a type from the unified label set $\mathcal{Y}_u$. Thus, during model training, using the mapping function $\phi : \mathbb{Y} \mapsto \mathcal{P}(\mathcal{Y}_u)$ we convert individual dataset specific labels to the unified label set, $\mathcal{Y}_u$. Due to one to many mapping, now there are multiple positive labels available for each individual input label y. Lets call the mapped label set for an input label y as $\mathcal{Y}_m$. Now, if any of the mapped label $\hat{y} \in \mathcal{Y}_m$ has descendants, then the descendants are also added to $\mathcal{Y}_m$[2]. For example, if the label *GPE* from the OntoNotes dataset, is mapped to the label *GPE* in the UHLS, then *GPE* as well as all descendants of *GPE* are possible candidates. This is because, even though the original example in OntoNotes is a name of a city, the annotation guidelines restrict the fine-labeling. Thus the mapped set would be updated to {*GPE*, *City*, *Country*, *County*, ...}. Additionally, some label have a one-to-many mapping, for example, for the label *MISC* in CoNLL dataset, the candidate labels could be {*product*, *event*, ...}.

From the set of mapped candidate labels, a partial label loss function will select the best candidate label. Due to the inherent design of the UHLS and label mapping, there will always be examples available that will be mapped only at a single leaf node. Thus allowing fine labels in the candidate set for actual coarse labels, will encourage model to predict finer labels across datasets.

3.2.3 Partial Hierarchical Label Loss. A partial label loss deals with the situation where training example have a set of candidate labels and among which only a subset is correct for that given example [4, 18, 30].

In our case, this situation arises because of the mapping of the individual dataset labels to the UHLS. We use a hierarchy aware partial loss function as proposed in [29]. We first compute the probability distribution for the labels available in $\mathcal{Y}_u$ as described in equation 1. Here W is a weight matrix of size $|R| \times |\mathcal{Y}_u|$ and x is the input entity mention along with its context.

$$p(y|x) = softmax(RW + b) \tag{1}$$

Then we compute $\hat{p}(y|x)$, a distribution adjusted to include a weighted sum of the ancestors probability for each label as defined in equation 2. Here $\mathcal{A}_t$ is the set of ancestors of the label y in $\mathcal{R}_u$ and β is a hyperparameter.

$$\hat{p}(y|x) = p(y|x) + \beta * \sum_{t \in \mathcal{A}_t} p(t|x) \tag{2}$$

Then we normalize $\hat{p}(y|x)$. From this normalized distribution, we select a label which has the highest probability and is also a member of the mapped labels $\mathcal{Y}_m$. We assumed the selected label to be correct and propagate the log-likelihood loss. The intuition behind this is that given the design of the ULHS and label mapping; there will always be examples where $\mathcal{Y}_m$ will contain only one element, in that case, the model gets trained for that label. In the case where there are multiple labels, the model has already built a belief about the fine label suitable for that example because of simultaneously training with inputs having a single mapped label. Restricting that belief to the mapped labels encourages correct fine-predictions for these coarsely labeled examples.

4 EXPERIMENTS AND ANALYSIS

In this section, we describe the datasets used, details of experiments related to UHLS creation, baseline models, model training, evaluation schemes and result analysis.

4.1 Datasets

Table 1 describes the seven datasets used in this work. These datasets are diverse, as they span several domains, none of them have an identical label set and some datasets capture fine-grained labels while others only have coarse labels. Also, the Wiki [14] dataset is

[2]This is exempted when the annotated label is a coarse label and a fine label from the same dataset exist in the subtree.

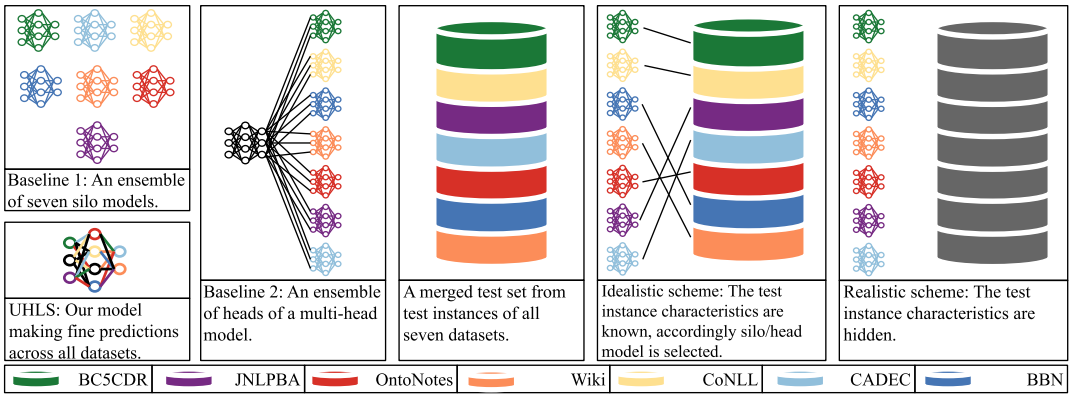


Figure 5: A pictorial illustration of the complete experimental setup.

Dataset	Domain	No. of Labels	Mention count	Fine labels
BC5CDR [13]	Clinical abstracts	2	9,385	No
CoNLL [25]	Reuters news stories	4	23,499	No
JNLPBA [12]	Life sciences abstracts	5	46,750	Yes
CADEC [11]	Medical forum	5	5,807	Yes
OntoNotes [28]	Newswire, conversations, newsgroups, weblogs	18	1,16,465	No
BBN [27]	Wall Street Journal text	73	86,921	Yes
Wiki [14]	Wikipedia	116	20,00,000	Yes

Table 1: Description of the seven ET datasets.

automatically generated using distant supervision process [5] and has multiple labels per entity mention in its label set. The other remaining datasets have a single label per entity mention.

4.2 UHLS and Label Mapping

We followed the Algorithm 1 to create the UHLS and the label mapping. To reduce the load on domain experts for verification of the label spaces, we initialized the UHLS with the BBN dataset hierarchy. We keep on updating the initial hierarchy until all the labels from the seven datasets were processed. There were total 223 labels in $\mathbb{Y}$ and in the end $\mathcal{Y}_u$ had 168 labels. This difference in label count is due to the mapping of several labels to one or multiple existing nodes, without the creation of a new node. This corresponds to case 2 of the UHLS creation process (lines 3-4, Algorithm 1). Also, this indicates the overlapping nature of the seven datasets. The label set overlap is illustrated in Figure 2. The *MISC* label from CoNLL dataset has the highest ten number of mappings to the UHLS nodes. Wiki and BBN datasets were the largest contributor towards fine labels with 96 and 57 labels at the leaf of UHLS. However, only 25 fine-grained labels were shared by these two datasets. This indicates that even though these are the fine-grained datasets with one of the largest label sets, each of them has complementary labels.

4.3 Baselines

We compared our learning model with two baseline models. The first baseline is an ensemble of seven learning models, where each model is trained on one of the seven datasets. We name this model a silo ensemble model[3]. In this ensemble model, each silo model has the same mention and context encoder structure described in Section 3.2.1. However, the loss function is different. For single-label datasets, we use a standard softmax based cross-entropy loss. For multi-label datasets, we use a sigmoid based cross-entropy loss.

The second baseline is a learning model trained using a classic hard parameter sharing multi-task learning framework [3]. In this baseline, all the seven datasets are fed through a common mention and context encoder. For each dataset, there is a separate classifier head with the output labels same as that was available in the respective original dataset. We name this baseline as a multi-head ensemble baseline[4]. Similar to the silo models, the appropriate loss function is selected for each head. The only difference between the silo and multi-head model is the way mention and context representations are learned. In the multi-head model, the representations are shared across datasets. In silo models, the representations are learned separately for each dataset.

4.4 Model Training

For each of the seven datasets, we use the standard train, validation and testing split. If the standard splits are not available, we randomly split the available data into 70%, 15%, and 15%, and use them as train, validation, and testing set respectively. In the case of the silo model, for each dataset, we train a model on its training split and select the best model using its validation split. In the case of the multi-head and our proposed model, we train the model on the training splits of all seven datasets together and select the best model using the combined validation split.[5].

4.5 Experimental Setup

Figure 5 illustrates the complete experimental setup along with the learning models compared. In this setup, the objective is to measure the learning model's generalizability for the ET task as a whole, rather than on any specific dataset. To achieve this, we

[3]Here unlike traditional ensemble models, in silo ensemble, the learning models are trained on different datasets.
[4]Here since the "task" is the same, i.e., entity typing, we use the term multi-head instead of multi-task for the baseline.
[5]The source code and the implementation details are available at: https://github.com/abhipec/ET_in_the_wild

merged the test instances from the seven datasets listed in Table 1 to form a combined test corpus. On this test set, we compared the performance of the baseline models with the learning model trained via our proposed framework. We compare these models performance using the following evaluation schemes.

4.5.1 Evaluation schemes. **Idealistic scheme:** Given a test instance, this scheme picks a silo model from the silo ensemble model (or head of the multi-head ensemble model) which has been trained on a training dataset with the same domain and target labels set as the test instance. This scheme gives an advantage to the ensemble baselines and compares the models in the traditional ways.

Realistic scheme: In this scheme, all of the test instances are indistinguishable in their domain and candidate label set. In other words, given a test instance, learning models do not have information about its domain and target labels. This is a challenging evaluation scheme and close to real-world setting, where once learning models are deployed, it cannot be guaranteed that the user submitted test instances will be from the same domain. In this scheme, the silo ensemble and multi-head ensemble models assign a label to a test instance based on the following criteria:

Highest confidence label (HCL): The label which has the highest confidence score among the different models/heads of an ensemble model. For example, let there be two models/heads, MA and MB, in a silo/multi-head ensemble model. For a test instance, MA assigns the score of 0.1, 0.2 and 0.7 for the labels l_1, l_2 and l_3 respectively. For the same test instance, MB assigns the score of 0.05 and 0.95 for the labels l_4 and l_5 respectively. Then the final label will be the label l_5 which has a confidence score of 0.95.

Relative highest confidence label (RHCL): The label which has the highest normalized confidence score among the different models/heads from an ensemble model. Continuing with the example mentioned above for MA and MB, in this criteria, we normalize the confidence score for each model based on the number of labels the model is predicting. In this example, MA is predicting three labels and MB is predicting two labels. Here the normalized scores for MA will be 0.3, 0.6 and 2.1 for the label l_1, l_2, and l_3 respectively. Similarly, the normalized scores for MB will be 0.1 and 1.9 for the label l_4 and l_5. Then the final label will be the label l_3 with the confidence score of 2.1.

Recall that the experimental setup includes multiple models, each having a different label set. The existing classifier integration strategies [31], such as sum rule or majority voting are not suitable in this setup. For these evaluation schemes, we use the evaluation metrics described in the following section.

4.6 Evaluation metrics

In the evaluation schemes, there are cases where the predicted label is not part of the gold dataset label set. For example, our proposed model or the ensemble model might predict a label *city* for a test instance which has a gold label annotated as a *geopolitical entity*. Here, the models are predicting a fine-grained label, however, the dataset from where the test instance came only had annotations at the coarse level. Thus, without manually verifying, it is not possible to know whether the model's prediction was correct or not. To overcome this issue, we propose two evaluation metrics, which allows us to compare learning models making predictions in different label sets with minimum re-annotation effort.

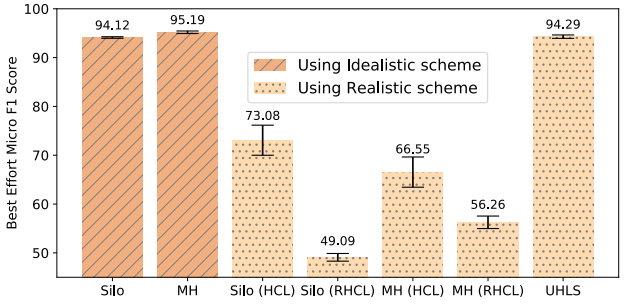


Figure 6: Comparison of learning models in the idealistic and realistic schemes.

In the first metric, we compute an aggregate micro-averaged F1 score on best effort basis. It is based on the intuition that if the labels are only annotated at a coarse level in the gold test annotations, then even if a model predicts a fine-label within that coarse label, this metric should not penalize such cases[6]. To find the fine-coarse subtype information, we use the UHLS and the label mapping. We map both prediction and gold label to the UHLS and evaluate in that space. We compute this metric both in an idealistic and realistic scheme. By design, this metric will not capture errors made at a finer level, which the next metric will capture.

In the second metric, we measure how good are the fine-grained predictions on examples where the gold dataset has only coarse labels. We re-annotate a representative sample of a coarse-grained dataset and evaluate the model's performance on this sample.

4.7 Result and Analysis

4.7.1 Analysis of the idealistic scheme results. In Figure 6, we can observe that the multi-head ensemble model outperforms the silo ensemble model (95.19% vs. 94.12%). The primary reason could be that the multi-head model has learned better representations using the multi-task framework as well as has an independent head for each dataset to learn dataset specific idiosyncrasy. The performance of our single model (UHLS) is between the silo ensemble model and multi-head ensemble model. Note that this performance comparison is in a setting which is the best possible case for ensemble models where the ensemble models know complete information about the test instance domain and label set. Despite this, UHLS model which does not require any information about test instance domain and candidate labels performs competitive (94.29%), even better than the silo ensemble model. Moreover, the ensemble models do not always predict the finest possible label, whereas UHLS can (§ 4.7.3).

4.7.2 Analysis of the realistic scheme results. In Figure 6, we can observe that both silo ensemble and multi-head ensemble model performs poorly in this scheme. The best result for ensemble models (73.08%) is obtained by the silo ensemble model when the labels were assigned using the HCL criteria. We analyzed some of the outputs of ensemble models and found that there were several cases

[6]Exception is where the source dataset also has fine-grained labels.

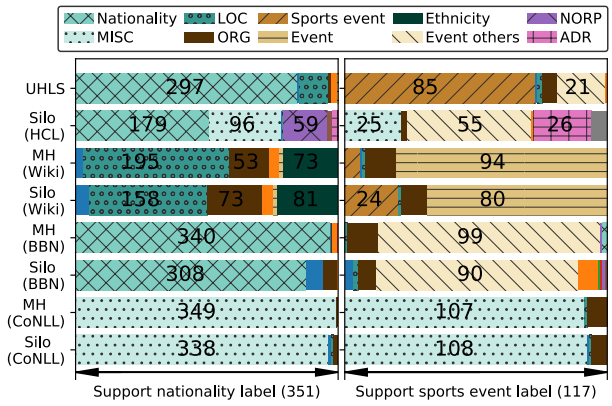


Figure 7: Analysis of Fine-grained label predictions. The two columns specify results for nationality and sports event label. Each row represents a model used for prediction. The results can be interpreted as, out of 351 entity mentions with type nationality, model Silo (CoNLL) predicted 338 as MISC type and the remaining as other types illustrated.

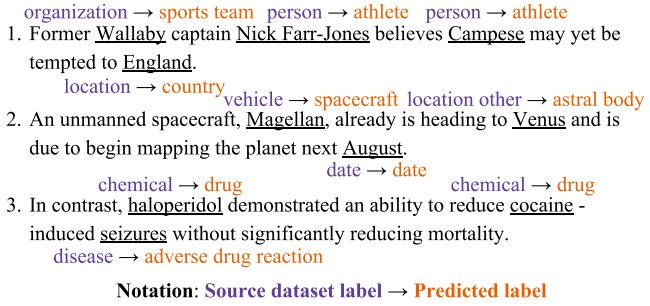


Figure 8: Example output of our proposed approach. Sentence 1, 2, 3 are from the CoNLL, BBN and BC5CDR dataset respectively.

where a narrowly focused model predicts with very high confidence (0.99 probability or above) out-of-scope labels. For example, prediction of label ADR with confidence 0.999 by a silo model trained on the CADEC dataset for a *sports event* test instance of Wiki domain. The performance of our UHLS model is 94.29%, which is an absolute improvement of 21.21% compared to the next best model Silo (HCL) model in the realistic scheme of evaluation.

4.7.3 Analysis of the fine-grained predictions. For this analysis, we re-annotate the examples of type *MISC* from the CoNLL test set into *nationality* (support of 351), *sports event* (support of 117) and others (support 234). We analyzed the prediction of different models for the *nationality* and *sports event* labels. Note that this is an interesting evaluation where the test instances domain is Reuters News, and the in-domain dataset does not have labels *nationality* and *sports event*. The *nationality* label is contributed by the BBN dataset whose domain is Wall Street Journal. The *sports event* label is contributed by the Wiki dataset whose domain is Wikipedia. The results (Figure 7) are categorized into three parts as described below:

In-domain results: The bottom two rows, Silo (CoNLL) and MH (CoNLL) represent these results. We can observe that in this case, since train and test dataset are from the same domain, these models can predict accurately the label *MISC* for both the *nationality* and *sports event* instances. However, *MISC* is not a fine-grained label. These results are from the idealistic scheme where it is known about the test instance characteristics.

Out of domain but with known candidate label: The middle four rows, Silo (BBN), MH (BBN), Silo (Wiki) and MH (Wiki) represent these results. In this case, we assume that the candidate labels are known, and pick the models which can predict that label. However, there is not a single silo/head model in the ensemble models which can predict both *nationality* and *sports event* labels. For example, model/head with the BBN label set can predict the label *nationality* but not the *sports event* label. For *sports event* instances, it assigns a coarse label *events other*, which also subsumes other events such as *elections*. Similarly, model/head with the Wiki label set can predict the label *sports event* but not the label *nationality*. For *nationality* instances, it assigns completely out of scope labels such as *location* and *organizations*. The out of scope predictions are due to the domain mismatch.

No information about domain or candidate label: The top two rows, Silo (HCL) and UHLS represent these results. The Silo (HCL) is a silo ensemble model with the realistic evaluation scheme. We can observe that this model makes out of scope predictions such as predicting *ADR* for *sports event* instances. The UHLS model is trained using our proposed framework. It can predict fine-grained labels in both *nationality* and *sports event* test instances, even though two different datasets contributed these labels. Also, it does not use any information about the test instance domain or candidate labels.

4.7.4 Example output on different datasets. In Figure 8, we show the labels assigned by the model trained using the proposed framework on the sentences from the CoNLL, BBN and BC5CDR datasets. We can observe that, even though the BBN dataset is fine-grained, it has complementary labels compared with the Wiki dataset. For example, for the entity mention `Magellan`, a label *spacecraft* is assigned. *Spacecraft* label is only present in the Wiki dataset. Additionally, even in sentences from clinical abstracts, the proposed approach is assigning fine-types, which came from a dataset with the medical forum domain. For example, *ADR* label is only present in the CADEC dataset with the domain of medical forum. The proposed approach can aggregate fine-labels across datasets and makes unified fine-grained predictions.

4.7.5 Result and analysis summary. Collective learning framework allows a limitation of one dataset being covered by some other dataset(s). Our results convey that a model trained using CLF on an amalgam of diverse datasets generalizes better for the ET task as a whole. Thus, the framework is suitable for the ET in the wild problem.

5 RELATED WORK

To the best of our knowledge, the work of [21] in the visual object recognition task is closet to our work. They consider two datasets. First a coarse-grained and second, a fine-grained. Label set of the first dataset is assumed to be subsumed by the label set of the second

dataset. Thus coarse-grained labels can be mapped to fine-grained dataset labels in a one-to-one mapping. Additionally, they did not propagate the coarse labels to the finer labels. As demonstrated by our experiments, when several real-world datasets are merged, one to one mapping is not possible. In our work, we provide a principled approach where multiple datasets can contribute to fine-grained labels. In our framework, a partial loss function enables fine-label propagation on datasets with coarse labels.

In the area of cross-lingual syntactic parsing, there is a notation of universal POS tagset [20]. This tagset is a collection of coarse tags that exist in similar form across languages. Utilizing this tagset and a mapping from language-specific fine-tags, it becomes possible to train a single model in a cross-lingual setting. In this case, the mapping is many-to-one, i.e., a fine-category to a coarse category, thus the models are limited to predict a coarse-grained label.

Related to the use of partial label loss function in the context of the ET problem, there exist other notable works including [22] and [1]. In our work, we use the current state-of-the-art hierarchical partial loss function proposed in [29].

6 CONCLUSION

In this paper, we propose building learning models that generalize better on the ET as a whole, rather than on a specific dataset. We comprehensively studied ET in the wild task which includes problem definition, collective learning framework, and evaluation setup. We demonstrated that by using in conjunction a UHLS, one-to-many label mappings, and a partial hierarchical loss function; we can train a single classifier on several diverse datasets together. The single classifier collectively learns from diverse datasets and predicts the best possible fine-grained label across all datasets, outperforming an ensemble of narrowly focused models in their best possible case. Also, during collective learning there is a multi-directional knowledge flow, i.e., there is no one source or target dataset. This knowledge flow is different from the well studied multi-task and transfer learning approaches [19] where the prime objective is to transfer knowledge from a source dataset to a target dataset.

In NLP there are several tasks such as entity linking [23], relation classification [9], and named entity recognition [17], where the current focus in on excelling at a particular dataset, not on a particular task. We expect that collective learning approaches will open up a new research direction for each of these tasks.

REFERENCES

[1] Abhishek Abhishek, Ashish Anand, and Amit Awekar. 2017. Fine-grained entity type classification by jointly learning representations and label embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 797–807.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, 1247–1250.

[3] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[4] Timothee Cour, Ben Sapp, and Ben Taskar. 2011. Learning from partial labels. *Journal of Machine Learning Research* 12, May (2011), 1501–1536.

[5] Mark Craven and Johan Kumlien. 1999. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 77–86.

[6] Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[7] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 601–610.

[8] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.

[9] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Association for Computational Linguistics, 94–99.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics* 55 (2015), 73–81.

[12] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*. Association for Computational Linguistics, 70–75.

[13] 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* 2016 (2016).

[14] Xiao Ling and Daniel S Weld. 2012. Fine-Grained Entity Recognition.. In *AAAI*, Vol. 12. 94–100.

[15] Hugo Liu and Push Singh. 2004. ConceptNetâĂŤa practical commonsense reasoning tool-kit. *BT technology journal* 22, 4 (2004), 211–226.

[16] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[17] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.

[18] Nam Nguyen and Rich Caruana. 2008. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 551–559.

[19] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[20] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

[21] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6517–6525.

[22] Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1369–1378.

[23] Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27, 2 (2014), 443–460.

[24] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural Architectures for Fine-grained Entity Type Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 1271–1280.

[25] 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-Volume 4*. Association for Computational Linguistics, 142–147.

[26] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[27] Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia* 112 (2005).

[28] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA* (2013).

[29] Peng Xu and Denilson Barbosa. 2018. Neural Fine-Grained Entity Type Classification with Hierarchy-Aware Loss. *arXiv preprint arXiv:1803.03378* (2018).

[30] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. 2017. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017), 2155–2167.

[31] Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.