Fine-grained Entity Detection and Typing

Thesis submitted in partial fulfilment of the requirements
for the award of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Abhishek

Under the supervision of

Dr. Amit Awekar and Dr. Ashish Anand

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

Guwahati - 781039 Assam India

July, 2020

Dedicated to my parents and my sister for their unconditional love and support.

Acknowledgements

I express my heartfelt gratitude to Dr. Amit Awekar and Dr. Ashish Anand, my Ph.D. advisors. They are excellent mentors and have always provided an atmosphere to pursues research problems that excites me the most. At the same time, their guidance helped in shaping ideas to concrete objectives while aiming at the ultimate goal. They always encouraged to participate in various seminars, tutorials, meetings, workshops, and conferences often by providing travel support, and I am grateful to them. I am also thankful to my doctoral committee members, Dr. Vijaya Saradhi, Dr. Sanasam Ranbir Singh, and Dr. Prithwijit Guha, for providing valuable feedback during the research.

I would like to thank my collaborators, Dr. Amar Prakash Azad and Balaji Ganesan, from IBM research India for their mentorship during an internship at IBM. I will also thank Riddhiman Dasgupta, Srikanth Tamilselvam, from IBM for their valuable suggestions during the internship. I will thank several interns and collaborators: Sanya Bathla Taneja, Garima Malik, Aneesh Barthakur, and Nitin Nair.

I am thankful to organizations such as Google, ACM India-IARCS, AAAI, and the Department of CSE, IITG, for providing travel support to attend international conferences. Also, I thank MHRD, Government of India, for providing financial assistantship throughout the Ph.D. program.

I am thankful to various funding agencies such as BRNS (project no. 2013/13/8-BRNS/10026), Department of Biotechnology (project no. BT/COE/34/SP28408/2018) and Dept. of CSE for providing necessary computing resources used during the work.

I am fortunate to have several wonderful friends who were there to support and celebrate with me during various events of the grad school experience. Especially I would like to thank VA Amarnath, Mirza Galib, Anasua Mitra, Akshay Parakh, Mohit Kumar, Gaurav Pandey, Debika Datta, Sunil Kumar Sahu, Sonia, and Tushar.

July 07, 2020 Abhishek

Declaration

I certify that

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisor(s).
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisor(s) are not in a position to check for any possible instance of plagiarism within this submitted work.

July 07, 2020 Abhishek

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

Dr. Amit Awekar
Assistant Professor
Email : awekar@iitg.ac.in
Phone : +91-361-258-2373

Dr. Ashish Anand
Associate Professor
Email: anand.ashish@iitg.ac.in
Phone: +91-361-258-2374

Certificate

This is to certify that this thesis entitled "**Fine-grained Entity Detection and Typing**" submitted by **Abhishek**, in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, to the Indian Institute of Technology Guwahati, Assam, India, is a record of the bonafide research work carried out by him under my guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India. To the best of my knowledge, no part of the work reported in this thesis has been presented for the award of any degree at any other institution.

Date: July 07, 2020
Place: Guwahati

Dr. Amit Awekar
(Main-supervisor)

Dr. Ashish Anand
(Co-supervisor)

Abstract

Detection and typing of entity mentions present in natural language text are one of the fundamental problems in information extraction paradigm. The dissertation directly focuses on advancing the state-of-the-art of the entity detection and entity classification problems in a setting where entity mentions can belong to a large set of types spanning diverse domains such as biomedical, finance, and sports. Moreover, the entity mentions could be mentioned in several text genres, such as newswire, scientific abstracts, and forums. When the scope of entity mentions is diverse, and several text genres are involved, data scarcity becomes one of the primary issues for these tasks.

The thesis addresses several issues related to the data scarcity, either directly or indirectly. First, we propose a noise-aware learning model for the task of fine-grained entity typing. The proposed model outperforms previous state-of-the-art models, which assumes that the training dataset is noise-free. The noise-aware model addresses the data-scarcity issue indirectly as the majority of datasets for the Fine-ET task are generated automatically using the distant supervision paradigm. The automatically generated datasets have noise, and thus noise-aware models permit efficient and effective learning. We also propose transfer learning approaches in cases where the training dataset size is small.

Second, we propose a collective learning framework for the task of fine-grained entity typing. The proposed framework aggregates different datasets which can have partial overlapping labels and can predict a unique fine-grained label for a given entity mention. The work also addresses the data scarcity issue as often we do not have datasets available with all of the label set annotated, and utilizing different datasets that have partial labels annotated eliminates the need to create new datasets.

Third, we propose a framework to improve the quality of the datasets generated in the distant supervision paradigm for the fine-grained entity detection and fine-grained entity typing task. Using the framework, we created two datasets, each containing more than thirty million sentences annotated with around hundred and thousand entity types, respectively. The work directly addresses the data scarcity issue by sharing new datasets for these tasks with the research community.

Contents

List of Figures

List of Algorithms

List of Tables

List of Abbreviations

CLF collective learning framework. 44

Coarse-ER Coarse-grained Entity Recognition. 62

ED Entity Disambiguation. 18

ED Entity Detection. 2

EL Entity Linking. 17

ER Entity Recognition. 2, 22

ET Entity Typing. 2

Fine-ED Fine-grained Entity Detection. 3

Fine-ET Fine-grained Entity Typing. 3

HAnDS Heuristics Allied with Distant Supervision. 9, 62

KB Knowledge Base. 4, 13

KBC Knowledge Base Construction. 16, 22

KG Knowledge Graph. 14

LP Link Prediction. 18

NDS Naive Distant Supervision. 64

NLP natural language processing. 1, 11, 22

OpenIE Open Information Extraction. 2

QA Question Answering. 22

RE Relation Extraction. 22

SPO (subject, predicate, object). 14, 15

UHLS unified hierarchical label set. 42

1

Introduction

1.1 Overview

One of the primary ways of sharing knowledge among humans is through documents written in natural language. These documents capture several aspects of human knowledge ranging from life-saving discoveries in biomedical sciences to technological advancements in space sciences. Much of this knowledge is currently difficult to access for computer algorithms as the knowledge is not expressed in any structured format (graphs or databases) that computers can easily understand. In the past two decades, there has been a considerable amount of work in the natural language processing (NLP) community to automatically extract important knowledge components from text documents and make them available in an easily accessible structured format such as graphs or databases. Some prominent projects include OpenCyc[1], Freebase [1], Google Knowledge Vault [2], DBpedia [3], YAGO [4], WikiData [5], and NELL [6]. The extracted structured information can then facilitate several applications such as helping virtual assistants/chatbots in answering factual questions and finding components of potential drugs that affect specific diseases.

There are several tasks involved in the process of extracting structured information from unstructured text. These tasks include detection of entity mentions present in the natural language sentences [7], linking or disambiguation of entity mentions to known entities [8], classification or typing of entity mentions into a set of predefined categories [9], and finding relations between entities [10]. The entities act as an essential constituent of the structured knowledge, where they act as nodes/keys in a graph/database. These entities

[1] http://www.opencyc.org

and their mentions are associated with various properties. One such important property is a type or a label or a category of an entity or an entity mention. The focus of this thesis is on the Entity Detection (ED) and Entity Typing (ET) problems.

Entity Detection is the task of detecting entity mentions belonging to predefined categories or types, in natural language sentences[2]. We illustrate the problem description with examples in Table 1.1. In the table, there are two sentences. The first sentence is from a news article, whereas the second sentence is from a medical forum. For the ED task, the input is a sentence, and output is word boundaries that constitute an entity mention from the predefined categories. For example, in the first sentence, four entity mentions belong to the predefined categories, and they are detected (underlined for illustration purpose). Similarly, in the second sentence, there are three entity mentions.

Domain	**Sentences (Input)**	**Predefined categories**	**Sentences with detected entities (Output)**
News	Former Wallaby captain Nick Farr-Jones believes Campese may yet be tempted to England.	*person*, *location*, *organization*	Former Wallaby captain Nick Farr-Jones believes Campese may yet be tempted to England.
Medical Forum	In contrast, haloperidol demonstrated an ability to reduce cocaine - induced seizures.	*drug*, *medical condition*	In contrast, haloperidol demonstrated an ability to reduce cocaine - induced seizures.

Table 1.1: **Entity Detection problem:** The objective is to identify entity mentions present in natural language sentences that belong to predefined categories. The input for an ED system is natural language sentences, and the output is boundaries of entity mentions.

Entity Typing is the task of assigning types or categories to already identified entity mentions present in sentences. We illustrate the problem description with examples in Table 1.2. In the table, the entity mentions are from the same two sentences as that present in Table 1.1. For the ET task, the input is an entity mention along with the context (which can be a full sentence), and output is a type or a category (from predefined categories), which best describes that entity mention.

The ED and ET tasks usually go hand in hand. We have a set of predefined categories and are interested in entity detection and typing based on the predefined categories. When ED and ET are solved together, it is referred to as the Entity Recognition (ER) task. The ER task is one of the fundamental tasks in NLP with over two decades of active research [12–14]. However, a majority of research work has been limited to a handful of predefined categories such as *person*, *location*, and *organization*. With the recent success

[2]In contrast, the Open Information Extraction (OpenIE) paradigm does not have predefined categories [11].

Domain	Entity Mention (underlined) with context	Predefined Categories	Entity type
News	Former Wallaby captain Nick Farr-Jones believes ...	*person*, *location*, *organization*	*organization*
	... believes Campese may yet be tempted to England.		*location*
Medical Forum	In contrast, haloperidol demonstrated an ability to reduce ...	*drug*, *medical condition*	*drug*
	... cocaine - induced seizures.		*medical condition*

Table 1.2: **Entity Typing problem:** The objective is to classify the identified entity mentions present in natural language sentences to predefined categories. The input for an ET system is entity mention along with its context (which can be a full sentence), and the output is a category which best describes that entity mention.

of machine learning techniques in several NLP areas, modern NLP applications require fine-grained information about entities from diverse domains. For example, in addition to *person*, *location*, and *organization*, a chatbot should be able to recognize entities from diverse domains such as entertainment, biomedical, and sports. The diversity can then enable fine-grained predictions, for example, instead of predicting a label to be an *organization*, predicting labels to be a *musical band* or *sports team*, which are subtypes of the label *organization*.

How can we detect and type entity mentions in fine-grained diverse domains setting? What are the limitations of existing work? These are the questions we explored in this dissertation. We describe the problem description in the subsequent section.

1.2 Problem Description

ED and ET in the setting where there is a diverse and large number of predefined categories (from hundreds to thousands of types spanning several domains) are denoted as **Fine-grained Entity Detection** (**Fine-ED**) and **Fine-grained Entity Typing** (**Fine-ET**) task respectively. We illustrate the problem description with examples in Figure 1.1. In the figure, the predefined categories contain *educational degree*, *engineering discipline*, *organism* categories. These categories are from different domains, education and biomedical. Existing works related to ED and ET fail to detect and type entity mentions belonging to these types [12, 13]. Moreover, even for common categories such as *person* and *organization* (mentioned in magenta color), the Fine-ET task assigns context-dependent fine-grained labels such as *biologist* or *university* instead of a coarse label *person* or *organization* respectively.

The contributions of this dissertation are in advancing the state-of-the-art for Fine-ED and Fine-ET tasks.

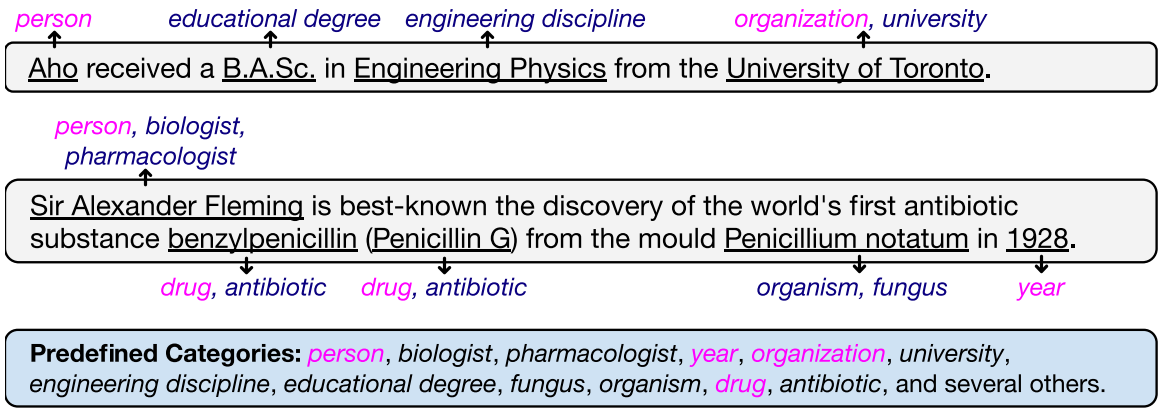


Figure 1.1: **Fine-ED and Fine-ET problem description:** Detection and typing of entity mentions is a setting where there is a diverse and large number of predefined categories. In the above example, the predefined categories contain types from two domains, biomedical and education.

1.3 Overview of Existing Work

Fine-ED and Fine-ET tasks are usually modeled as a supervised learning problem as for a given input; the objective is to detect and type entity mentions from a set of predefined categories. For any supervised classification task, there are two essential components: the data and the learning model. We provide an overview of existing work by analyzing some of the crucial properties of existing datasets and learning models.

1.3.1 Datasets

Datasets play a crucial role in advancing the state-of-art of any machine learning task. Fine-ED and Fine-ET tasks require a dataset where all entity mentions belonging to hundreds to thousands of categories are annotated with the context-dependent fine-grained types. Manually creating a dataset for these tasks is an expensive and time-consuming process as an entity mention could be assigned multiple types from a set of thousands of types. The existing work predominantly uses the distant supervision paradigm [15, 16] to create training datasets for these tasks [9, 17, 18]. We describe the paradigm with an illustration in Figure 1.2. In this paradigm, the entity mentions present in the text are linked to a Knowledge Base (KB), which provides types to the linked mentions. For example, in

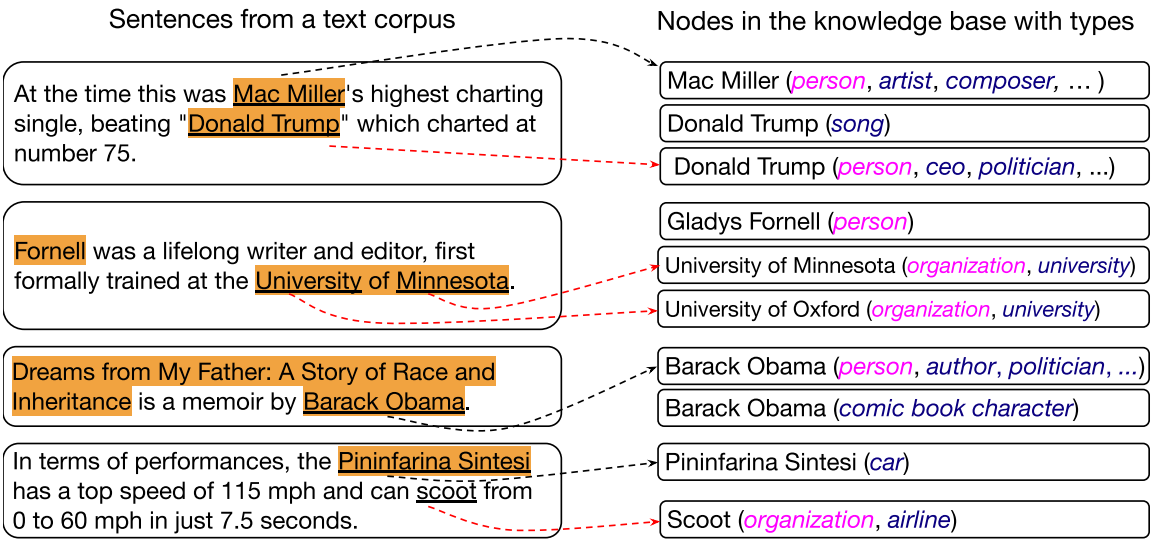


Figure 1.2: **Distant supervision paradigm:** In this paradigm, the entity mentions present in the sentences are linked to a KB, which provides the types to the linked mention. For this illustration, the highlighted tokens are actual entity mentions, whereas the underlined tokens are the mentions linked to the KB by the distant supervision paradigm. The black arrows denote correct linking, whereas the red arrows denote incorrect linking between tokens and KB entity.

Figure 1.2 first sentence, the entity mention Mac Miller is linked to a node in the KB with the same name. The types obtained for this entity mention from KB includes *person*, *artist*, and *composer*. The linking between entity mentions in text corpus to a KB can be via a named entity linker tool [19] or links can be manually created as in the case of Wikipedia corpus. The advantage of the distant supervision method is that it does not require human involvement[3], which could otherwise be very costly. On the other hand, there are some limitations to this method as listed below:

1. **Noise in datasets:** Since distant supervision is a fully automated method, it is not possible to avoid noise in the annotations. The noise can be categorized into two categories; entity type noise and entity boundary noise:

   (a) **Entity type noise:** The distant supervision process assigns context agnostic types to entity mentions. For example, for every mention of entity Barack Obama will receive the same set of labels irrespective of the context surrounding an entity mention as illustrated in Figure 1.2. For example, in the third sentence of the figure, the entity mention Barack Obama receives an out-of-context type

[3]Although humans create the links in Wikipedia text, they are created for better readability, thus, not specific for the Fine-ED or Fine-ET tasks.

politician. In datasets such as FIGER [17], this noise is around 28%, whereas, in the datasets such as D-BBN and D-ONTONOTES [20], this noise is around 26% and 22%, respectively [9][4]. Further, the labels assigned by distant supervision can be false positives (an extra label is assigned as illustrated) or false negatives (failed to assign a context specific label).

(b) **Entity boundary noise:** The distant supervision process can fail to mark the entity boundary correctly. For example, in Figure 1.2, the entity mentions Fornell and Dreams from My Father: A Story of Race and Inheritance has not been marked. The entity mention University of Minnesota has been incorrectly marked. A non-entity mention scoot has been marked. In the FIGER dataset, this noise is over 50% [23].

2. **Entity type coverage:** The entity types present in distantly supervised datasets are restricted to the types present in KBs. Although KBs have entity types from several domains, still some types might not be present in any KB. In such scenarios, where some required types are not present in any KB, creating a training dataset becomes a challenging task in the distant supervision paradigm.

3. **Text source:** The text source in the distant supervised methods is predominantly Wikipedia text. The primary reason is that several thousands of users have manually linked different mentions of concepts in Wikipedia text to Wikipedia articles. These articles act as a key in a KB and thus provide good quality annotations. For other text sources such as from news or medical forum, it is difficult to obtain a linked text corpus where links are manually created. For these text sources, named entity linker tools [19] can be used. However, the resultant annotation quality is sub-par when compared with Wikipedia text. The named entity linker tool can make frequent linking errors, as illustrated with red arrows in Figure 1.2. The linking errors introduce entity type and entity boundary noises in the resultant datasets.

1.3.2 Learning Models

The training datasets for the Fine-ED and Fine-ET tasks are not manually annotated and have some limitations, as discussed in the previous Section 1.3.1. Thus, the state-of-the-

[4]In the corresponding paper and related works, these datasets are referred to as ONTONOTES and BBN datasets. However, they are not the original ONTONOTES [21] and BBN [22] datasets, but a variant modified using the distant supervision paradigm. In the thesis, we use both the original and the modified variants in different chapters. Thus, two avoid confusion, the modified variants are referred to as D-ONTONOTES and D-BBN datasets.

art models also try to address dataset limitations along with the task characteristics via modelling. We categorize these approaches based on the way they address the dataset limitations.

1. **Noise-aware learning:** In this paradigm, it is assumed that in the training dataset, there is some annotation noise. Several methods have been proposed in this paradigm for different tasks such a binary classification [24], multi-label classification [25], and sequence labeling [26]. In the context of the Fine-ED and Fine-ET tasks, the Fine-ET task has received much attention from the noise-aware learning research community. The primary reasons are:

   (a) The label noise present in the existing distantly supervised datasets such as FIGER, D-BBN, and D-ONTONOTES is approx 22–28% [9]. For several noise-aware learning models [24–26], this level of noise is moderate. If the noise percentage is high (greater than 50%), it is difficult to learn useful information by learning models.

   (b) The entity boundary noise present in existing datasets such as FIGER is high, especially approx 50–60% of the entity mentions are not marked in these datasets. This setting makes it extremely difficult for learning models to learn useful information. Moreover, prior works [17] assumed that all the fine-grained types are just a sub-type of well studied coarse-grained types such as *person*, *location*, *organization* and *miscellaneous*, which is not the case, as found in our work [23].

   Thus due to the above reasons, there are several works (including ours) related to noise-aware learning models for the Fine-ET task.

2. **Efficient learning in data scarcity:** The entity type coverage and text source limitations of existing datasets can be considered as, in general, the data scarcity issue. Here, no or minimal (a few hundred to thousand sentences) training dataset is available for a particular text source or entities of some particular types. In this scenario, transfer learning [27], few-shot [28], and zero-shot [29] learning approaches can be used, and our work explores some of these directions for the Fine-ET task.

1.4 Our Contributions

We make three contributions to the Fine-ED and Fine-ET tasks. Our first two contributions are focused on proposing better leaning models for the Fine-ET task, and the third

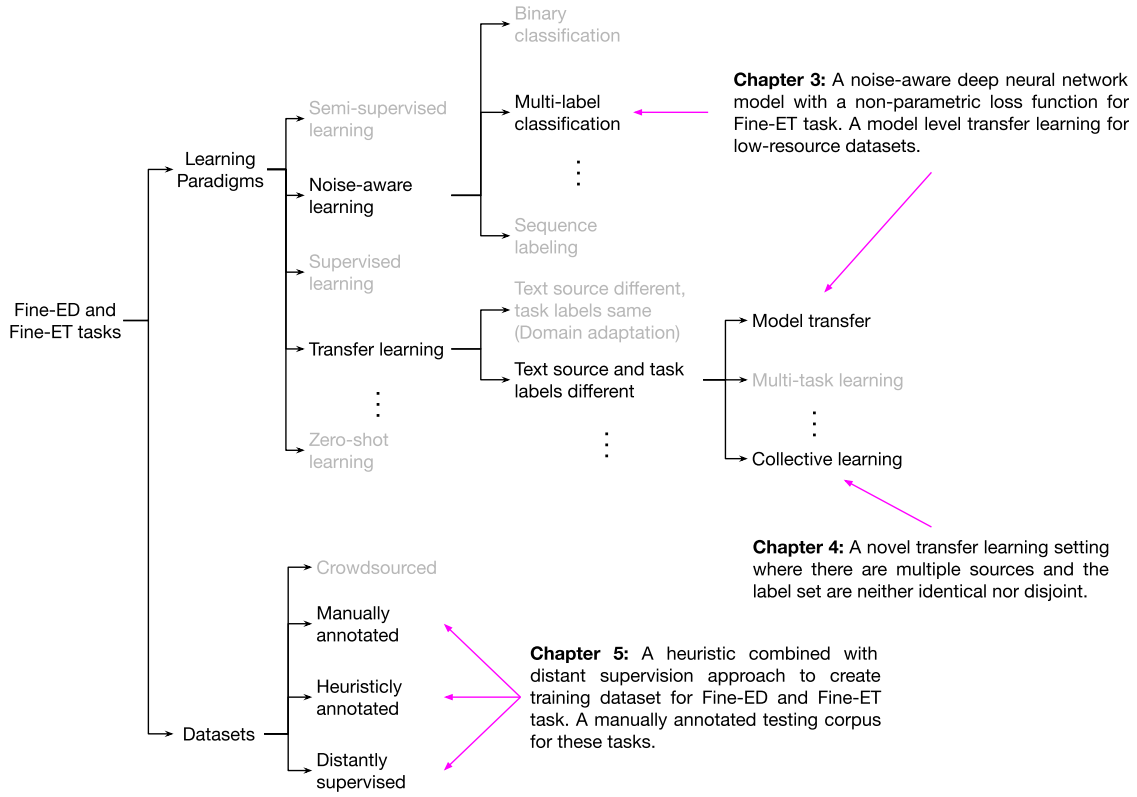


Figure 1.3: An overview of the thesis contributions.

contribution is focused on creating better datasets for the Fine-ED and Fine-ET tasks. The ordering of these contributions conveys the trend followed in the community for the past three years. An overview of the contributions is illustrated in Figure 1.3.

Noise-aware learning model and transfer learning for the Fine-ET task: While some of the earliest work for the Fine-ET task used hand-crafted features or assumed that the training datasets are noise-free. In Abhishek et al. [30], we propose a deep neural network model that learns representations for a given entity mention and its context, while incorporating noisy label information. The proposed model uses two different loss functions, one for the mentions with label noise and other for mentions, which are noise-free. Experimental results on the FIGER and the D-BBN datasets demonstrate the effectiveness of the proposed model, with an average relative improvement of 2.69% in the loose-micro-F1 score.

There is a significant variation in the number of annotations available per label on average for the D-BBN (1.8k) and the FIGER (21k) dataset. Thus, in order to learn a better representation of entity mentions for datasets with fewer annotations, in

addition to the noise-aware model, we also use model level and feature level transfer learning. We show that feature level transfer learning can be used to improve the performance of models such as proposed in Ren et al. [9] by up to 4.5% in loose-micro-F1 score on datasets with fewer annotations. Similarly, model level transfer learning can be used to improve the performance of the proposed model using different datasets by up to 3.8% in loose-micro-F1 score on datasets with fewer annotations. These transfer learning settings will be useful in cases where a new small dataset is created to either address the entity type coverage or text source limitation of the existing datasets.

Collective Learning Framework for the Fine-ET task: Prior work, as well as our earlier contribution related to transfer learning, were only limited to a setting where the target text source has a fine-grained annotated dataset available. However, there exist scenarios where the need is to build Fine-ET systems for a particular text source other than Wikipedia or with some types that might not be present in any KB, or both. Thus the creation of datasets in the distant supervision paradigm will be challenging. To address these issues, in Abhishek et al. [31], we propose a collective learning framework. The framework can aggregate type information, either fine-grained or coarse-grained, from different datasets. These datasets can be either distantly supervised or manually annotated. The aggregation is done to satisfy the desired requirement, i.e., the types and text source. Thus, for types that are not present in distantly supervised datasets, the framework can use other datasets where those types are present. For text sources where no fine-grained types are available, the framework can utilize coarse-grained types available for that text source and can predict fine-grained types learned from datasets with different text sources. The type sets between different datasets can have a partial overlap, and they need not be disjoint.

The core idea behind the framework is to organize the labels available in different datasets into a unified hierarchy. The unified hierarchy will then be used to train multiple datasets collectively in a single model with a single partial-loss function. Thus, enabling fine-grained predictions on all datasets. We demonstrate the efficacy of the proposed framework in an experimental setting consisting of seven diverse datasets, each with a different text source and different type set. The proposed approach outperforms competitive baselines with a significant margin.

New datasets for the Fine-ED and Fine-ET task: Prior work assumed that all fine-

grained types are subtypes of well studied coarse-grained types such as *person*, *location*, *organization*, and *miscellaneous*. Thus earlier works used a model trained on the CoNLL dataset [32] (which has these types manually annotated) for the Fine-ED task. In Abhishek et al. [23], we observe that this is not a valid assumption, and a model trained on these coarse-grained datasets misses lots of diverse entity mentions. Moreover, existing distantly supervised datasets such as FIGER, have very large entity boundary noise, which makes them not suitable for the Fine-ED task. To address these issues, we propose a Heuristics Allied with Distant Supervision (HAnDS) framework. The HAnDS framework uses a three-stage pipelined approach to construct the training dataset automatically. At each stage, different heuristics are used to reduce the errors introduced by naively using the distant supervision paradigm. For any given fine-grained types derived from a KB, the framework can be used to automatically construct quality datasets suitable for both the Fine-ED and Fine-ET tasks. The entity boundary noise is reduced by approx 50% compared with the FIGER dataset. Additionally, we provide a thousand sentence corpus of manually annotated entity mentions for the Fine-ED and Fine-ET tasks. This corpus is four times larger corpus than the FIGER evaluation corpus.

1.5 Thesis Outline

Following the just discussed two central themes of datasets and learning models for the Fine-ED and Fine-ET tasks, the thesis is organized as follows:

In Chapter 2, we first describe essential terminologies and concepts associated with the Fine-ED and Fine-ET tasks. Then we provide an overview of conventional modeling approaches and evaluation metrics for these tasks.

In the next three chapters, we present the three contributions of the thesis. In Chapter 3, we present a noise-aware deep learning model and transfer learning techniques for the Fine-ET task. In Chapter 4, we present a collective learning framework to address the text source and type limitations of exiting work. In Chapter 5, we present an approach to automatically construct quality datasets for both the Fine-ED and Fine-ET tasks.

Finally in Chapter 6, we conclude and discuss future research work.

2

Background

This chapter provides a background necessary to understand the thesis better. We begin by describing essential terminologies and concepts, followed by describing conventional modeling approaches and evaluation metrics related to the ED and ET tasks.

2.1 Preliminaries

Entity: An entity is an object that exists in an identified universe. An entity can have physical existence or can be abstract. For example, persons, locations, products, scientific concepts are considered as an entity. The scope of entities is usually defined by the guidelines of the particular NLP task in hand. For example, for the task of entity recognition using the CoNLL dataset [32], diseases are not considered as entities. Whereas, for the same task using the NCBI disease corpus [33], diseases are considered as entities, whereas persons are not considered as entities.

Entity Mention: The span of words or tokens in text, which refers to an entity, is denoted as an entity mention. Let us consider the following sentences: "Paris is a 2008 French film by Cédric Klapisch concerning a diverse group of people living in Paris. The film is set principally in Paris, with one thread of the story set in Africa." In these sentences, the entity mentions are underlined. There are three occurrences of entity mention Paris. Two of them refer to the entity `Paris` (a city), and one of them refers to the entity `Paris` (a movie). However, all of them are different entity mentions and will be assigned different entity mention ids.

Remarks: Use of word "Named" before Entity

The word "Named" in the phrase "Named Entity" restricts the scope of entities to those which have one or more referent rigid designator as defined by S. Kripke [34]. For example, the computer company founded by Steve Jobs in 1976 is referred to as Apple Inc. On the other hand, the word June could refer to a month of an undefined year, which could be June 2019 or June 2020. Thus it is not considered as a rigid designator. The scope of entities considered in this dissertation sometimes does not have a rigid designator; thus, the named word is omitted.

Entity Types: Entity type or label is a semantic category such as *person*, *location* or *city*, assigned to an entity mention or an entity. Entity mentions or entities which have same types tend to have similar characteristics. Two entity types can be mutually exclusive or can have an intersection, which usually is in the form of a hypernym/hyponym hierarchical structure. For example, type *city* is a hyponym of type *location*. In Figure 2.1, we illustrate a subset of type hierarchy used in DBpedia, which has around 750 entity types arranged in a hierarchical structure. Other KBs such as WikiData and Freebase have a different number of entity types and have different hierarchical structures.

Text source: A text source is a text corpus from which sentences are sampled to generate a training dataset for an NLP task. For example, the text source for the CoNLL dataset [32] is Reuters Corpora (RCV1) [35], which contains articles from English language news stories published by Reuters Ltd. The selection of text source depends on the end application for an NLP task. For example, if the application is to recognize entities mentioned in news articles by a particular publisher, then the task is ER, and an ideal text source will be sentences sampled from similar news articles. The same source can be used for multiple tasks, and the same task can have several datasets with different sources. For example, the Reuters Corpora (RCV1) is used as a text source for preparing the CoNLL dataset for the ER task, and the same source is used in preparing the AIDA-CoNLL [36] dataset for entity linking task. Also, there exists the W-NUT [37] dataset for the ER task, with text source as tweets. For these tasks, since the text source defines the input characteristics such as what features will be used and what will be the data distribution of the input, in the machine learning

Figure 2.1: The figure illustrates a subset of entity type hierarchy maintained by the DBpedia project. The full hierarchy is available at http://mappings.dbpedia.org/server/ontology/classes/.

nomenclature, text source is denoted as a **domain**.

Remarks: Text genre

A text genre can refer to a widely recognized class of text which have some common purpose or characteristics [38]. For example, news stories, fiction, and scientific abstracts are considered as different text genres. Within a genre, there could be several differences. For example, both the BBN dataset [22] and the CoNLL dataset have the same news genre, however, they have different sources. The text source for the BBN dataset is the Wall Street Journal text, and the text source for the CoNLL dataset is the Reuters Corpora. Due to the text source differences, their data characteristics, such as vocabulary and content, are different.

Knowledge Base: A Knowledge Base (KB) is a store of information that represents facts about the world. Unlike traditional databases, which are represented as tables, a

popular way to store KBs on a computer is in a graph structure format, called Knowledge Graph (KG). The nodes of the graph represent entities, and the labeled edges represent relations between entities [39]. A KG can be represented by a collection of (subject, predicate, object) (SPO) triples, where *subject* and *object* denote entities and *predicate* denotes a binary relation type. Multiple binary relations can represent Higher-arity relations in a KG. Some KGs, such as YAGO3[1], also store *time* and *location* information along with SPO triples. There are several KBs available today, and the prominent among them are:

1. **Freebase:** Freebase is a large collaborative KB, initially developed by MetaWeb in 2007, which was acquired by Google in 2010. Its data is harvested from many sources, including individuals and user-contributed Wikipedia edits. Freebase contains around 637 million non-redundant facts, in which there are around 40 million entity instances [2]. It is not updated since 2015 and is available for download at `http://freebase.com`.

2. **DBpedia:** The DBpedia project was started in 2007 jointly by the Free University of Berlin and the University of Leipzig to automatically extract structured information contained in Wikipedia, such as infoboxes, category information, geo-coordinates, and external links. It contains around 600 million triples in the English language and around 2.5 billion triples in other languages combined.

3. **WikiData:** WikiData[2] is a collaborative edited KB maintained by Wikimedia Foundation. Along with triples, the project also aims to capture sources of facts and represent them with tuples. It contains around 19 millions tuples.

4. **YAGO:** YAGO (Yet Another Great Ontology) is a KB developed at the Max Planck Institute for Computer Science, Germany. The current version of the YAGO project is YAGO3. It contains around 120 million facts related to 10 million entities. Based on various heuristics and algorithms, YAGO3 has been automatically created using Wikipedia, WordNet[3], and GeoNames[4] as data sources. Its accuracy is manually estimated to be around 95% and the triples are annotated with its confidence value.

[1] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/`
[2] `http://wikidata.org/`
[3] `https://wordnet.princeton.edu/`
[4] `http://www.geonames.org/`

A detailed comparison of these KBs is available in a survey paper by Fäber et al. [40]. All of these KBs have some different characteristics and are actively used in research.

Entity Linking Entity Linking (EL) is the task of identifying and linking text phrases to a concept representing those phrases in a KB. We illustrate the EL task with examples in Figure 2.2. In the Figure, a text excerpt is mentioned in the center surrounded by twelve nodes of a KB. The text phrases which represent some of the concepts in the KB are linked, as shown by arrows. For example, the phrase Miller is linked to a node Mac Miller. Note that Miller is also a representing phrase for two other nodes in this KB, namely a Miller (a name of a crater) and Miller (a name of a moth). However, the context surrounding the text phrase conveys that Mac Miller will be the most appropriate node.

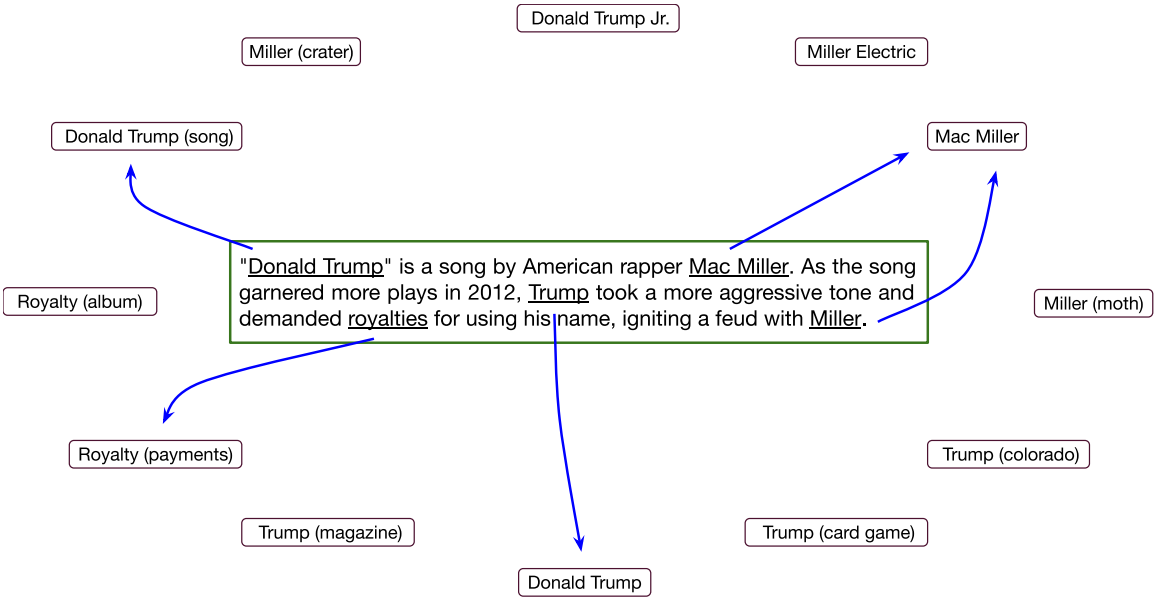


Figure 2.2: A toy illustration of the entity linking task, where text phrases are identified and linked to their representative nodes in a KB.

In contrast with the toy illustration of the EL task in Figure 2.2, EL using real KBs is a challenging task. There are millions of nodes in KBs such as DBpedia and Freebase. A text phrase could be a representative mention of thousands of candidate nodes, and a node could have several representative phrases. Recent state-of-the-art EL models use deep-neural network architectures [41] and achieve a performance of around 86% in the Micro-F1 score on standard datasets such as AIDA-CoNLL [36]. There also exist several off-the-self entity linker tools such as DBpedia spotlight [19], which can link text phrases to concepts in the DBpedia KB.

When compared with the Fine-ED and Fine-ET tasks, the Fine-ED task can be considered as a prerequisite to EL models that only does Entity Disambiguation (ED) [42], i.e., linking already marked representative phrases to a KB. On the other hand, the context-dependent fine-grained types of entity mention provided by Fine-ET models improve the performance of entity linker systems, as reported in Gupta et al. [43].

2.2 Learning Models

The nature and size of predefined categories majorly govern the choice of learning models. When there are a handful number of non-overlapping predefined categories, the ED and ET tasks are usually modeled jointly as a Entity Recognition (ER) task, and sequence labeling approaches are dominant. On the other hand, if there are a large number of overlapping predefined categories, as in the case of Fine-ED and Fine-ET tasks, then these tasks are modeled independently.

2.2.1 Sequence Labelling

In sequence labeling, the input to a learning model is a sequence of observations, and the output is a categorical label assigned to each observed value of the sequence. In Figure 2.3, we illustrate how a sequence labeling approach can be used for the task for ER. The input to the sequence labeler is a sentence, i.e., a sequence of words, and the output is a sequence of entity tags. As the entities can span several words, the entity tags are encoded using an encoding scheme such as IOB encoding [44], to capture multi-word entities. In simple words, in IOB encoding, the label of any token that begins a span of entity is prefixed with tag **B**, tokens that occur inside a span of entity are prefixed with tag **I** and any tokens that are outside the span of entities are labeled as **O**.

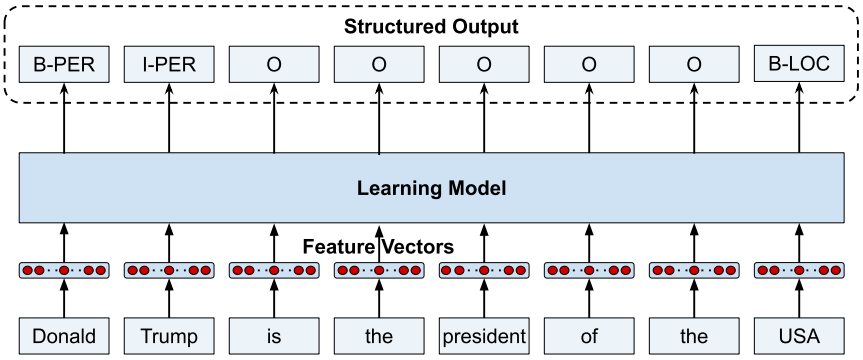


Figure 2.3: The Entity Recognition task modeled by a sequence labeling approach.

Given such encodings, we have reduced the span recognition task as a per-word labeling task. However, predicting labels for words independent of the labels assigned to neighboring words may not result in a good sequence of tags. For example, the predicted output sequence might contain the **I** tag followed by the **O** tag. Thus typically, a structured prediction model such as Conditional Random Field (CRF) [45] is used to find the optimal tag sequence instead of the optimum local tag for each word. The state-of-the-art sequence labeling models use a combination of deep recurrent neural networks with CRF [46].

2.2.2 Independent modeling approach for ED and ET tasks

When there are overlapping predefined categories, typically in the case of Fine-ED and Fine-ET tasks, the ED and ET tasks are modeled independently. The primary reason being that the state-space of hidden variables in the structured prediction models such as CRF grows exponentially due to the possibility of multiple labels per entity mention. In this case, the ED task is modeled as a sequence labeling task with one label, i.e., entity or not an entity, and the ET task is modeled as a classification task. An illustration of an independent modeling approach for the ED and ET tasks is provided in Figure 2.4.

2.3 Evaluation Metrics

In this section, we will first define evaluation metrics in a general setting where an entity mentions can have multiple labels. Then we reduce this general definition to individual cases such as evaluation metric for the ED task, for the ET task with multiple labels, etc. The evaluation metrics defined in this section are from Ling and Weld [17].

For the evaluation, we have to compare sentences annotated with true/gold annotations with the same sentences annotated by an output of a model. Let set $\mathcal{T}$ denotes all true entity mentions, and set $\mathcal{P}$ denotes all predicted entity mentions. For an entity mention m, we denote the true set of tags as t_m and predicted set of tags as $\hat{t}_m$. Since we are defining the evaluation metric for a multi-label setting, both t_m and $\hat{t}_m$ set can have more than one value. Also, if $m \notin \mathcal{T}$ then, $t_m = \emptyset$, i.e., t_m is an empty set. Similarly, if $m \notin \mathcal{P}$, then, $\hat{t}_m = \emptyset$, i.e., $\hat{t}_m$ is an empty set. Using these definitions, we can compute precision and recall in the following three ways:

Strict: In this metric, the prediction is considered correct if and only if $t_m = \hat{t}_m$.

$$\text{precision} = \frac{\sum_{m \in \mathcal{P} \cap \mathcal{T}} I(t_m = \hat{t}_m)}{|\mathcal{P}|} \tag{2.1}$$

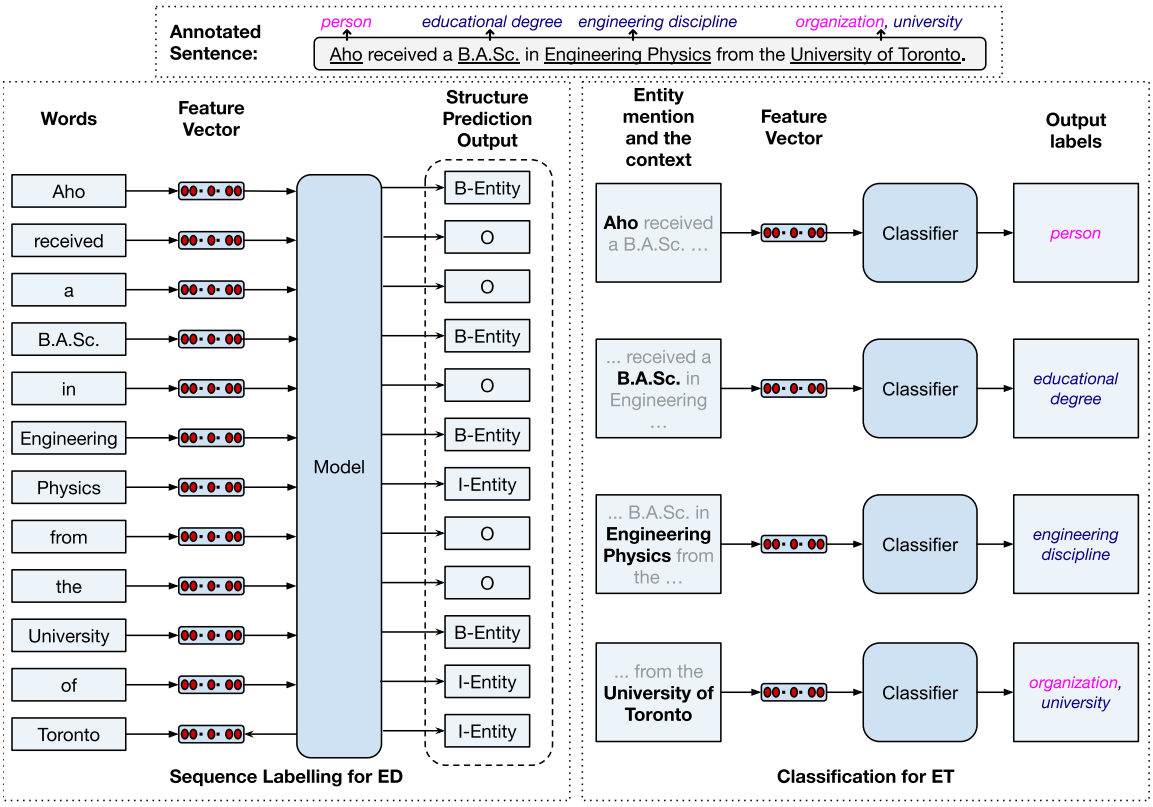


Figure 2.4: In the case of overlapping labels, the ED and ET tasks are modeled independently, as illustrated. For the ED task, a sequence labeling approach is used where the annotated sentence's multi-label annotations are converted into a binary (entity, non-entity) annotations. For the ET task, a classifier is used to assign labels to the entity mentions present in the sentence independently.

$$\text{recall} = \frac{\sum_{m \in \mathcal{P} \cap \mathcal{T}} I(t_m = \hat{t}_m)}{|\mathcal{T}|} \tag{2.2}$$

In the above equations, I is an indicator random variable, whose value is 1 if $t_m = \hat{t}_m$ else 0.

Loose Macro: In this metric, precision and recall scores are computed over each entity mention, based on the intersection of the true and predicted tag set. For each mention, m, a score between 0 to 1 is assigned, instead of exact 0 or 1, as assigned by the strict metric.

$$\text{precision} = \frac{1}{|\mathcal{P}|} \sum_{m \in \mathcal{P}} \frac{|\hat{t}_m \cap t_m|}{|\hat{t}_m|} \tag{2.3}$$

$$\text{recall} = \frac{1}{|\mathcal{T}|} \sum_{m \in \mathcal{T}} \frac{|\hat{t}_m \cap t_m|}{|t_m|} \tag{2.4}$$

Loose Micro: This metric also allows a non-zero score for a partial tag set match. However, the scores are aggregated globally instead of local aggregation in the case of the loose macro metric.

$$\text{precision} = \frac{\sum_{m \in \mathcal{P}} |t_m \cap \hat{t}_m|}{\sum_{m \in \mathcal{P}} |\hat{t}_m|} \tag{2.5}$$

$$\text{recall} = \frac{\sum_{m \in \mathcal{T}} |t_m \cap \hat{t}_m|}{\sum_{m \in \mathcal{T}} |t_m|} \tag{2.6}$$

From these definitions of precision and recall, the F1 score, which is a harmonic mean of precision and recall, is computed to compare learning models using a single number.

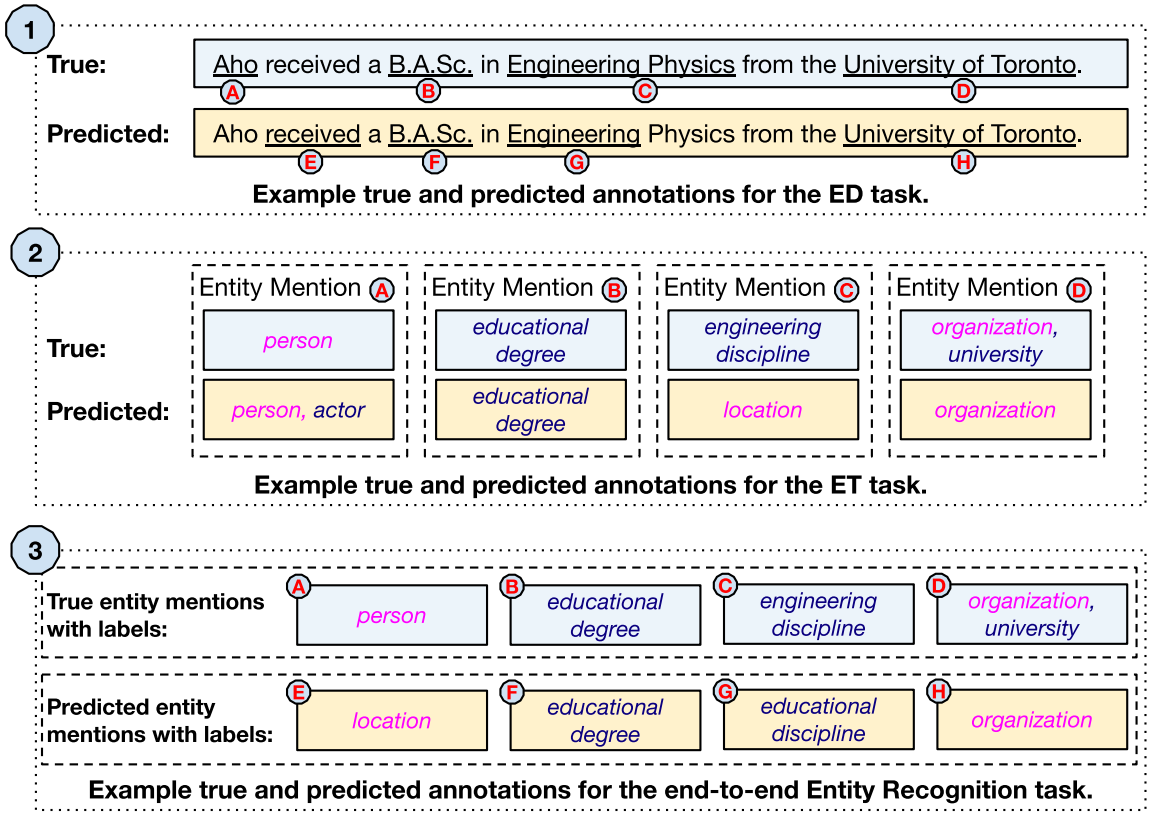


Figure 2.5: Different examples used to explain various evaluation metrics, as mentioned in Section 2.3.1, 2.3.2, and 2.3.1.

2.3.1 Evaluation Metric for ED

For the ED task, since a sequence of words can be either an entity mention or not an entity mention, i.e., a binary label, all of the above metrics reduce to the same values. For example, in Figure 2.5 (the first part, on the top), there are four true entity mentions, and four predicted entity mentions. The precision value from any of the above metrics

will be 0.5, and the recall value from any of the above metrics will be 0.5. Here, the precision and recall values are equivalent to the most common definition of precision and recall based on the true positive, false positive, and false negative. A standard way to compute precision and recall for ED task is to use the CoNLL evaluation script available at https://www.clips.uantwerpen.be/conll2002/ner/bin/conlleval.txt.

2.3.2 Evaluation Metric for ET

For the ET task, the set $\mathcal{P}$ and $\mathcal{T}$ are identical, as we are only interested in assigning types/labels to entity mentions. Now there are two possibilities; first, an entity mention can have a single label only; second, an entity mention can have multiple labels. The first setting is the multi-class classification problem, and the second setting is the multi-label classification problem (the Fine-ET task). For the multi-class problems, all of the above metrics reduces to accuracy measure. Whereas in the multi-label setting, the strict metric (also called subset accuracy) is one of the most strict metric, as if and only if all the true and predicted labels are equal, then this metric has a positive increment. The other two metrics, loose macro, and loose micro, allows partial match, and they differ in how they aggregate the scores of entity mentions. The loose macro computes partial precision and recall values per entity mention and then take an average. In contrast, the loose micro computes partial scores globally and then compute precision and recall.

We explain these metrics using the examples given in Figure 2.5 (the second part, in the middle). In the example, there are four entity mentions marked as **A**, **B**, **C**, and **D**. The true and predicted types of these entity mentions are also mentioned. The strict precision and recall values for this example will be as follows:

$$\text{precision} = \frac{0+1+0+0}{4} = 0.25$$

$$\text{recall} = \frac{0+1+0+0}{4} = 0.25$$

The precision and recall values will always be the same for the strict (subset accuracy) metric in the case of the ET task. Thus, in the next chapters, while evaluation ET models, we will refer to this metric as strict or subset accuracy and will not compute the F1 value.

The loose macro precision and recall values for the example will be as follows:

$$\text{precision} = \frac{1}{4}(\frac{1}{2} + \frac{1}{1} + \frac{0}{1} + \frac{1}{1}) = 0.625$$

$$\text{recall} = \frac{1}{4}(\frac{1}{1} + \frac{1}{1} + \frac{0}{1} + \frac{1}{2}) = 0.625$$

The loose micro precision and recall values for the example will be as follows:

$$\text{precision} = \frac{1+1+0+1}{5} = 0.6$$

$$\text{recall} = \frac{1+1+0+1}{5} = 0.6$$

2.3.3 Evaluation Metric for end-to-end Entity Recognition

The strict, loose macro and loose micro metrics can also be used in an end-to-end evaluation of entity recognition, i.e., both ED and ET tasks. Here, the ET task can be either a multi-class classification problem or a multi-label classification problem. In this case, where the ET task is a multi-class classification problem, then all of the above metrics will reduce to the same precision and recall metrics, as used in CoNLL NER evaluations [32]. In the other case, were ET task is a multi-label classification task, we explain these metrics using the example given in Figure 2.5 (the third part, on the bottom).

In the example, there are four true entity mentions marked as **A**, **B**, **C**, and **D**, and four predicted entity mentions marked as **E**, **F**, **G**, and **H**. Among these entity mentions, we can observe that the mentions **B** and **F** are identical, i.e., the prediction model detected the same boundaries as well as assigned the same type as that in true annotations. On the other hand, mentions **D** and **H** have the same boundaries but different labels. The other predicted mentions do not match with true annotations.

The strict precision and recall values for the example will be as follows:

$$\text{precision} = \frac{1+0}{4} = 0.25$$

$$\text{recall} = \frac{1+0}{4} = 0.25$$

The loose macro precision and recall values for the example will be as follows:

$$\text{precision} = \frac{1}{4}(\frac{0}{1} + \frac{1}{1} + \frac{0}{1} + \frac{1}{1}) = 0.5$$

$$\text{recall} = \frac{1}{4}(\frac{0}{1} + \frac{1}{1} + \frac{0}{1} + \frac{1}{2}) = 0.375$$

The loose micro precision and recall values for the example will be as follows:

$$\text{precision} = \frac{0+1+0+1}{4} = 0.5$$

$$\text{recall} = \frac{0+1+0+1}{5} = 0.4$$

Chapter Summary

In this chapter, we first described some of the key terminologies and concepts related to the ED and ET tasks such as entity, entity mentions, knowledge bases, and text sources. Then we briefly described conventional modeling approaches and evaluation metrics for these tasks.

3

Noise-aware Model and Transfer Learning for Fine-ET

Chapter Highlights

- The datasets available for the Fine-ET task have label noise.
- We propose a noise-aware deep neural network model for the Fine-ET task.
- The proposed model achieves state-of-the-art performance on two datasets, namely D-BBN and FIGER.
- We investigate transfer learning strategies to further improve the performance on the D-BBN dataset.
- We also analyzed under what conditions the proposed model works better and why learning models on the D-ONTONOTES datasets have inferior performance.
- This chapter is based on the publication "Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embedding" presented at EACL 2017.

3.1 Abstract

The distant supervision paradigm is extensively used to generate training data for the Fine-ET task. In this paradigm, the same set of labels is assigned to every mention of

an entity without considering its local context. These context agnostics labels act as a noise in the training datasets. In this chapter, we first propose a noise-aware deep learning model for the Fine-ET task. The model treats training data as noisy and uses a non-parametric variant of the hinge loss function to learn effectively in the presence of noise. Experiments show that the proposed model outperforms previous state-of-the-art methods on two publicly available datasets, namely FIGER and D-BBN, with an average relative improvement of 2.69% in the loose-micro-F1 score. We also investigate different transfer learning techniques such as model parameters transfer and learned feature transfer. These approaches of transferring knowledge further improve the performance of learning models trained on datasets with fewer training examples. The code to replicate the results reported in this chapter is available at https://github.com/abhipec/fnet.

3.2 Introduction

In the past decade, there has been a considerable amount of work on the ER task [32, 47–49], which classifies entity mentions into a small set of mutually exclusive types, such as *person*, *location*, *organization*, and *miscellaneous*. However, these types are not enough for some NLP applications such as Relation Extraction (RE) [16], KBC [2], and Question Answering (QA) [7]. In RE and KBC tasks, knowing fine-grained types for entities can significantly increase the performance of the relation classification systems [6, 17, 50] since this helps in filtering out candidate relation types that do not follow the type constraint. In QA systems, the fine-grained entity types provide additional information while matching questions to its potential answers and significantly improve performance [51]. For example, Li and Roth [52] rank questions based on their expected answer types (will the answer be *food*, *vehicle*, or *disease*).

In the Fine-ET task, there are over a hundred labels or types, typically arranged in a hierarchical structure. These types are context-dependent, i.e., two different mentions of same entity can have different labels. We illustrate the context-dependent type characteristics through an example in Figure 3.1. In the figure, all the three sentences S1, S2, and S3, mention the same entity, `Barack Obama`. However, looking at the context, we can infer that S1 mention `Barack Obama` as a *person, author*, S2 mention `Barack Obama` only as a *person*, and S3 mention `Barack Obama` as a *person, politician*.

Available training data for the Fine-ET task has noisy labels as they were automatically created via the distant supervision [15, 16] process. The distant supervision process links

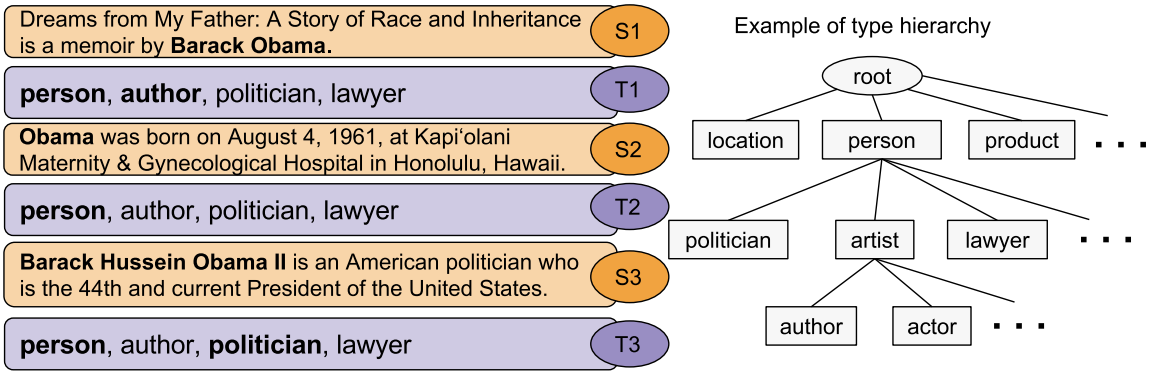


Figure 3.1: Context independent types assigned via the distant supervision process introduces noise in datasets. For example, the types assigned to an entity mention (bold typeface) in sentences (S1-S3) via the distant supervision process are mentioned in the T1-T3 field. Given the context, only a subset of these types is relevant, as denoted by bold typeface in T1-T3.

entity mentions in a corpus to a KB such as Freebase [1], DBpedia [3], or YAGO [4]. Then, the labels assigned to an entity in a KB are assigned to the linked entity mention. For example, let a KB have the labels *person*, *politician*, *lawyer*, and *author* for an entity `Barack Obama`. The distant supervision process will assign these four labels to every entity mention referring to the entity `Barack Obama`. Thus, the training data generated via the distant supervision paradigm will fail to assign context-dependent labels. This issue of noisy labels is also illustrated in Figure 3.1.

Existing Fine-ET systems have one or both of the following drawbacks: (1) they assume that the training dataset is noise-free [17, 18, 53, 54]; (2) they use hand-crafted features [9, 17, 18, 53]. In several real-world datasets such as FIGER and D-ONTONOTES, approximately twenty-five percent of training data has noisy labels [20]. The first drawback of assuming noise-free labels in the training data propagates noise to the Fine-ET models. Several existing state-of-the-art models use hand-crafted features for model training, which are extracted using various NLP tools. Since errors inevitably exist in such tools, the second drawback propagates errors of these tools to the Fine-ET models.

In this chapter, first, we propose a deep neural network based model to overcome the two drawbacks of existing Fine-ET systems. The model separates training data into *clean* and *noisy* partitions using the same method as proposed by Ren et al. [9]. For these partitions, we propose to use a simple yet effective non-parametric variant of the hinge loss function. To avoid the use of hand-crafted features, the proposed model learns representations for given entity mention and its context.

Feature learning based deep-learning models require large training datasets. However,

obtaining a large training dataset can be a challenging task for some specific text sources. In this chapter, we also investigate effectiveness of using transfer learning [55] techniques for the Fine-ET task both at feature and model level, for datasets with fewer annotations. We show that feature level transfer learning can be used to improve the performance of other Fine-ET systems such as proposed in [9] by up to 4.5% in the loose-micro-F1 score. Similarly, model level transfer learning can be used to improve the performance of the proposed model using different datasets by up to 3.8% in the loose-micro-F1 score.

Our contributions can be summarized as follows:

1. We propose a simple deep neural network model that learns representations for entity mention and its context and incorporates noisy label information using a variant of a non-parametric hinge loss function. Experimental results on two publicly available datasets demonstrate the effectiveness of the proposed model, with an average relative improvement of 2.69% in the loose-micro-F1 score.

2. We investigate the use of feature level and model level transfer-learning strategies in the domain of the Fine-ET task. The proposed transfer learning strategies further improve the state-of-the-art on the D-BBN dataset by 3.8% in the loose-micro-F1 score.

3.3 Related Work

The earliest works on the ER and ET tasks were focused primarily on few entity types such as *person*, *location*, and *organization* [56]. In the last two decades, there have been several attempts made to expand the type set to include more categories, both at coarse and fine-level. Some of the earliest such attempts were limited to type expansion in few domains as it requires human involvement by either providing heuristics or by manual annotations. For example, in Fleischman and Hovy [57] and Guiliano and Gliozzo [58], the authors propose a bootstrapping and heuristics approaches to classify entity mentions of type *person* into eight to twenty-one fine-categories. Similarly, in Fleischman [59] and Lee and Lee [60], the authors proposed heuristics and bootstrapping approaches to categorize entity mentions of type *location* into eight to ten fine-categories.

Later, heuristics and bootstrapping based approaches were expanded to cover multiple domains simultaneously. For example, in Sekine and Nobata [61], the authors proposed a heuristics approach to recognize entity mentions in approx 200 categories. Similarly, in Nadeau [62], the author proposed a bootstrapping approach to recognize entity mentions

belonging to hundred types. A survey of several earlier ER work is available in Nadeau and Sekine [12].

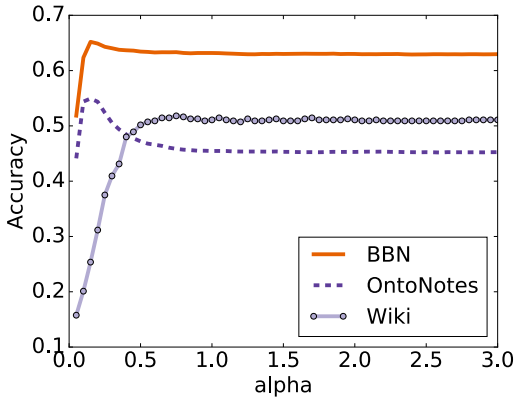
Heuristics and bootstrapping approaches are useful to make a Fine-ET system with little manual supervision quickly. However, accuracies of such systems are limited by the amount of effort made by humans in covering more domains and more data. Recently, with the widespread use of Wikipedia and KBs such as Freebase and YAGO, the distant supervision based methods gained popularity. In the distant supervision paradigm, one can exploit the linkage between Wikipedia and KBs to construct a dataset for the Fine-ET task quickly. The generated dataset will have some label noise as the annotations obtained are context-independent.

In the distant supervision paradigm, Ling and Weld [17] proposed the first system for Fine-ET task with 112 overlapping labels. They used a linear classifier perceptron for multi-label classification. Yosef et al. [18] used multiple binary SVM classifiers in a hierarchy to classify an entity mention to a set of 505 types. While the initial work assumed that all labels present in a training dataset for an entity mention are correct, Gillick et al. [63] introduced context dependent Fine-ET. They proposed a set of heuristics for pruning labels that might not be relevant given the entity mention's local context. Yogatama et al. [53] proposed an embedding based model where user-defined features and labels were embedded into a low dimensional feature space to facilitate information sharing among labels.

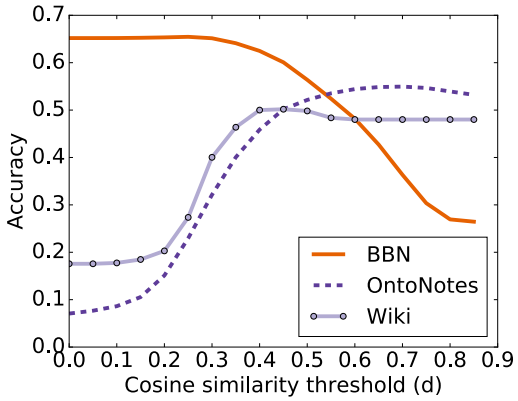
Shimaoka et al. [54] proposed an attentive neural network model that used LSTMs to encode entity mention's context and used an attention mechanism to allow the model to focus on relevant expressions in the entity mention's context. However, the model assumed that all labels obtained via distant supervision are correct. In contrast, our model does not assume that all labels are correct. To learn entity representation, we propose a scheme that is simpler yet more effective.

Most recently, Ren et al. [9] have proposed AFET, a Fine-ET system. It uses separate loss functions for *clean* and *noisy* entity mentions. The proposed loss function in AFET is parametric with a model parameter α used to model the label-label correlation information of the training dataset. During inference, AFET uses a threshold to separate positive labels from negative labels (similarity threshold parameter d). We observe that the AFET system is sensitive to change in α and d, as illustrated in Figure 3.2. In contrast, our model uses a simple yet effective variant of the hinge loss function. The function is non-parametric, and there is no data-dependent threshold used during the model inference.

Transfer learning is well applied to many NLP applications, such as cross-domain doc-



(a) The parameter α is used within the loss function to model label-label correlation. Higher the α, the lower is the margin between non-correlation labels.



(b) The threshold parameter d is used during inference, and the labels above this threshold are predicted as positive.

Figure 3.2: The effect of change of parameters α and d on AFET's performance evaluated on the D-BBN, D-ONTONOTES and FIGER datasets.

ument classification [64], multi-lingual word clustering [65], and sentiment classification [66]. Initialization of word vectors with pre-trained word vectors in neural network models can be considered as one of the best examples of transfer learning in NLP. A brief overview of the use of transfer learning in several NLP applications is available in a survey paper by Wang et al. [67].

3.4 The Proposed Model

In this section, we describe the proposed model, along with the proposed loss function. The proposed model assigns context-dependent types to entity mentions present in natural language sentences. A general overview of our proposed approach is illustrated in Figure 3.3.

3.4.1 Problem description

Input: The input to the model is a training and a testing corpus consisting of a set of sentences in which entity mentions have been identified. In the training corpus, every entity mention will have corresponding labels according to a given hierarchy. Formally, a training corpus $\mathcal{D}_{train}$ consists of a set of sentences, $\mathcal{S} = \{s^i\}_{i=1}^{N}$. Each sentence s^i will have one or more entity mentions denoted by $m_{j,k}^i$, where j and k denote indices of start and end tokens, respectively. Set $\mathcal{M}$ consists of all the entity mentions $m_{j,k}^i$. For every entity mention $m_{j,k}^i$, there will be a corresponding label vector $l_{j,k}^i \in \{0,1\}^K$, which is a

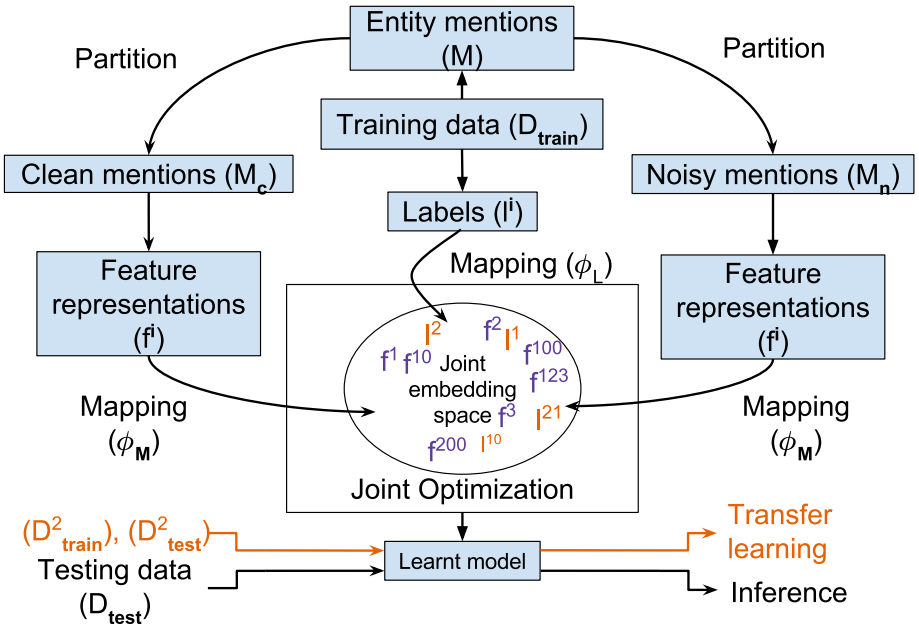


Figure 3.3: The overview of the proposed Fine-ET system.

binary vector, where $l^i_{j,k_t} = 1$ if t^{th} type is true otherwise it will be zero. K denotes the total number of labels in a given hierarchy Ψ. The testing corpus $\mathcal{D}_{test}$ will only contain sentences and entity mentions.

Output: For entity mentions in the testing corpus $\mathcal{D}_{test}$, predict their corresponding labels.

3.4.2 Training set partition

In the training dataset, the entity mentions have context-independent labels, which is considered as noise in the Fine-ET task. We use a heuristic as proposed in Ren et al. [9] to partition the mention set $\mathcal{M}$ of training corpus $\mathcal{D}_{train}$ into two partitions. The first partition set $\mathcal{M}_c$, consisting only of clean entity mentions and the second partition set $\mathcal{M}_n$, consisting only of noisy entity mentions.

An entity mention $m^i_{j,k}$ is said to be clean if its labels $l^i_{j,k}$ belong to only a single path (not necessary to be leaf) in the hierarchy Ψ, that is its labels are not ambiguous; otherwise, it is noisy. For example, as per hierarchy given in figure 3.1, an entity mention with labels *person*, *artist* and *politician* will be considered as noisy, whereas entity mention with labels *person*, *artist* and *actor* will be considered as clean.

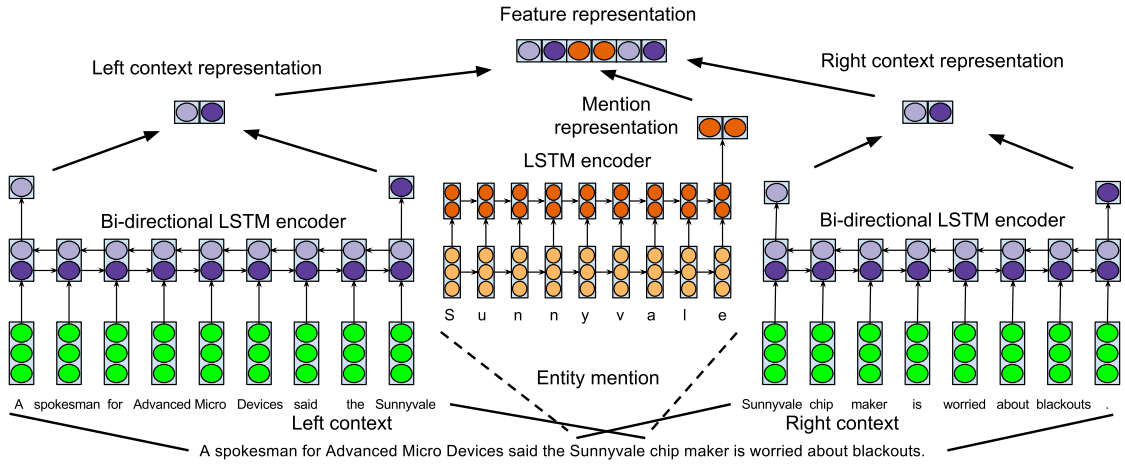


Figure 3.4: The feature learning architecture of the proposed Fine-ET model.

3.4.3 Feature representations

The model automatically learns input feature representations using a proposed deep-neural network architecture. The model learns two types of feature representations. The first is the representation of an entity mention learned via modeling the character sequences of an entity mention. The second is the representation of the context surrounding a mention via modeling the word sequences of a sentence.

Mention representation: This representation captures information about entity mention's morphology and orthography. We decompose an entity mention into a character sequence and use a vanilla LSTM encoder [68] to encode character sequences to a fixed dimensional vector. Formally, for entity mention $m^i_{j,k}$, we decompose it into a sequence of character tokens c^i_{j,k_1}, c^i_{j,k_2}, $\ldots$,$c^i_{j,k_{|m^i_{j,k}|}}$, where $|m^i_{j,k}|$ denotes the total number of characters present in the entity mention. For entity mention containing multiple tokens, we join these tokens with a space in between tokens. Every character will have a corresponding vector representation in a lookup table for characters. The character sequence is then fed one by one to an LSTM encoder, and the final output is used as a feature representation for entity mention $m^i_{j,k}$. We denote this process by a function $F_m : \mathcal{M} \rightarrow \mathbb{R}^{D_m}$, where D_m is the number of dimensions for mention representation. The whole process is illustrated in Figure 3.4 (mention representation).

Context representation: This representation captures information about the context surrounding the entity mention. The context representation is further divided into two parts, the left, and the right context representation. The left context consists of a sequence

of tokens within a sentence from the start of a sentence till the last token of the entity mention. The right context consists of a sequence of tokens from the start of the entity mention till the end of a sentence. We use bi-directional LSTM encoders [69] to encode token level sequences of both contexts to a fixed dimensional vector. Formally, for an entity mention $m^i_{j,k}$ present in a sentence s^i, decompose s^i into a sequence of tokens s^i_1, s^i_2, ..., s^i_k for the left context, and s^i_j, s^i_{j+1}, ..., $s^i_{|s^i|}$ for the right context, where $|s^i|$ denotes the number of tokens in the sentence. Every token will have a corresponding vector representation in a lookup tables for token. The token sequence is then fed one by one to a bi-directional LSTM encoder, and the final output will be used as feature representation. We denote this whole process by function $F_{lc} : (\mathcal{M}, \mathcal{S}) \rightarrow \mathbb{R}^{D_{lc}}$ for computing left context and $F_{rc} : (\mathcal{M}, \mathcal{S}) \rightarrow \mathbb{R}^{D_{rc}}$ for computing right context. D_{lc} and D_{rc} are the number of dimensions for the left context and the right context representation, respectively. The whole process is illustrated in the Figure 3.4 (left and right context representation).

The context representation described above is slightly different from what is proposed by Shimaoka et al. [54], where they exclude the entity mention from the context. In our proposed model, we include entity mention tokens within both the left and the right context, to explicitly encode context relative to an entity mention.

After obtaining the feature representations for the mention and the context, we concatenate these representations into a single D_f dimensional vector, where $D_f = D_m + D_{lc} + D_{rc}$. This complete process is denoted by a function $F : (\mathcal{M}, \mathcal{S}) \rightarrow \mathbb{R}^{D_f}$ given by:

$$F(m^i_{j,k}, s^i) = F_m(m^i_{j,k}) \oplus F_{lc}(m^i_{j,k}, s^i) \oplus F_{rc}(m^i_{j,k}, s^i) \tag{3.1}$$

where $\oplus$ denotes vector concatenation. For brevity, we will now omit the use of subscript j, k from $m^i_{j,k}$ and $l^i_{j,k}$, and will use f^i to denote feature representation for entity mention and its context obtained via equation 3.1.

3.4.4 Feature and label embeddings

Similar to Yogatama et al. [53] and Ren et al. [9], we embed feature representations and labels in a same dimensional space such that an object is embedded closer to the objects that share similar types than the objects that do not. Formally, we are trying to learn linear mapping functions $\phi_{\mathcal{M}} : \mathbb{R}^{D_f} \rightarrow \mathbb{R}^{D_e}$ and $\phi_{\mathcal{L}} : \mathbb{R}^{D_K} \rightarrow \mathbb{R}^{D_e}$, where D_e is the size of embedding space. These mappings are given by:

$$\phi_{\mathcal{M}}(f^i) = f^{i^T} U;\ \phi_{\mathcal{L}}(l^i_t) = l^{i^T}_t V \tag{3.2}$$

where, $U \in \mathbb{R}^{D_f \times D_e}$ and $V \in \mathbb{R}^{D_K \times D_e}$ are projection matrices for features representations and labels respectively and l_t^i is one-hot vector representation for label t. We assign a score to each label type t and feature vector as a dot product of their embeddings. Formally, we denote a score as:

$$s(f^i, l_t^i) = \phi_{\mathcal{M}}(f^i) \cdot \phi_{\mathcal{L}}(l_t^i) \tag{3.3}$$

3.4.5 Optimization

We use two different loss functions to model clean and noisy entity mentions. For the clean entity mentions, we use a hinge loss function. The intuition is simple: maintain a margin, centered at zero, between positive and negative type scores. The scores are computed by the similarity between an entity mention and label types (eq. 3.3). The hinge loss function has two advantages. First, it intuitively separates positive and negative labels during inference. Second, it is independent of data dependent parameters. Formally, for a given entity mention m^i and its label l^i we compute the associated loss as given by:

$$\begin{aligned} L_c(m^i, l^i) = &\sum_{t \in \gamma} \max(0, 1 - s(m^i, l_t^i)) \\ &+ \sum_{t \in \bar{\gamma}} \max(0, 1 + s(m^i, l_t^i)) \end{aligned} \tag{3.4}$$

where γ and $\bar{\gamma}$ are set of indices that have positive and negative labels respectively.

For noisy entity mentions, we propose a variant of a hinge loss where, like L_c, the score for all negative labels should go below -1. However, for positive labels, as the model does not know which labels are relevant to an entity mention's local context, we propose that the maximum score from the set of given positive labels should be greater than one. This maintains a margin between all negative types and the most relevant positive type. Formally, noisy label loss, L_n is defined as:

$$\begin{aligned} L_n(m^i, l^i) = &\sum_{t \in \bar{\gamma}} \max(0, 1 + s(m^i, l_t^i)) \\ &+ \max(0, 1 - s(m^i, l_{t^*}^i)); \\ t^* = &\arg\max_{t \in \gamma} s(m^i, l_t^i) \end{aligned} \tag{3.5}$$

Again, using this loss function makes it intuitive to set a threshold of zero during inference.

These loss functions are different from the loss functions used in Ren et al. [9] and

Datasets	FIGER	D-ONTONOTES	D-BBN
# types	128	89	47
# training mentions	2690286	220398	86078
# testing mentions	563	9603	13187
% clean training mentions	64.58	72.61	75.92
% clean testing mentions	88.28	94.00	100
% pronominal testing mentions[1]	0.00	6.78	0.00
Max hierarchy depth	2	3	2

Table 3.1: Statistics of the datasets used in the Fine-ET work.

Yogatama et al. [53] in a way that the proposed loss function makes strict absolute criteria to distinguish between positive and negative labels. Whereas in [9, 53], positive labels should have a higher score than negative labels. As their scoring is relative, the final result varies on the threshold used to separate positive and negative labels, which is dataset dependent.

To train the partitioned dataset together, we formulate the joint objective problem as:

$$\min_{\theta} O = \sum_{m \in \mathcal{M}_c} L_c(m, l) + \sum_{m \in \mathcal{M}_n} L_n(m, l) \tag{3.6}$$

where θ is the collection of all model parameters that need to be learned. To jointly optimize the objective O, we use Adam [70], a stochastic gradient-based optimization algorithm.

3.4.6 Inference

For every entity mention in the set $\mathcal{M}$ from $\mathcal{D}_{test}$, we perform a top-down search in the given type hierarchy Ψ and estimate the correct type path Ψ_*. Starting from the tree root, we recursively compute the best type among node's children by computing its score with obtained feature representations. We select the node that has a maximum score among other nodes. We continue this process till a leaf node is encountered, or the score associated with a node falls below an absolute threshold zero. The threshold is fixed across all datasets used.

3.4.7 Transfer learning

There is a significant variation of size in the available training datasets for the Fine-ET task, as indicated in Table 3.1. The FIGER dataset contains around 2.7 million entity

[1]We considered an entity mention as pronominal if all of its tokens have a POS tag as a pronoun.

mentions, whereas the D-BBN dataset contains around 0.08 million entity mentions. The deep-learning models are usually known to be data hungry in order to automatically learn good feature representations of the input [71]. Since it is not always possible to create a large-scale training dataset for a particular text source or domain, we investigate, whether the knowledge learned by the proposed model while trained on a large dataset can be transferred to other models and datasets.

We study two variants of transfer learning techniques, the feature level, and the model level transfer learning. In the feature level transfer learning, we study what contribution these feature representations make to existing feature engineering method such as AFET. To do so, we train the proposed model on one training dataset, namely the FIGER dataset, which has the highest number of entity mentions, among other datasets. Then we use this model to generate feature representations, that is, $F(m^i_{j,k}, s^i)$ for training and testing splits of other datasets. These representations, which are D_f dimensional vectors, are used as a feature for an existing state-of-the-art model, AFET, in place of the hand-crafted features that were initially used. The AFET model is then trained using these feature representations. On the other hand, in the model level transfer learning, while training the proposed model on small datasets, we initialize weights of LSTM encoders with the weights learned from the model trained on the FIGER dataset.

3.5 Experiments

3.5.1 Datasets used

We evaluate the proposed model on three publicly available datasets, provided in a pre-processed tokenized format by Ren et al. [9]. Some basic statistics of these datasets are listed in Table 3.1. The details of the datasets are as follows:

FIGER : The training data consisted of Wikipedia sentences and was automatically created via the distant supervision paradigm, by mapping hyperlinks in Wikipedia sentences to Freebase. The test data, mainly consisting of sentences from news reports, were manually annotated as described in Ling and Weld [17].

D-ONTONOTES : The D-ONTONOTES dataset consists of sentences from newswire documents present in the OntoNotes text corpus [21]. DBpedia spotlight [19] was used to link entity mentions in sentences to Freebase automatically. For this corpus, manually annotated test data was shared by Gillick et al. [63].

D-BBN : The D-BBN dataset consists of sentences from Wall Street Journal articles [22]. These sentences are manually annotated. To make this dataset adaptable to a KB hierarchy, Ren et al [20], linked the annotated entities to Freebase using DBpedia spotlight. The types obtained from Freebase were then applied to entity mentions and were mapped to the Freebase hierarchy. The original types which cannot be mapped to Freebase were discarded.

3.5.2 Evaluation setting

Baselines

We compared the proposed model with state-of-the-art entity typing methods:

FIGER: FIGER is a Fine-ET system proposed by Ling and Weld [17], which uses a multi-label perceptron model. The model used hand-crafted features as inputs to the perceptron and assumes that the training dataset is noise-free.

HYENA: HYENA is a Fine-ET system proposed by Yosef et al. [18], which uses several binary classifiers in a hierarchy. Similar to FIGER, the model uses hand-crafted features as inputs to the classifiers and assumes that the training dataset is noise-free.

AFET: AFET is a Fine-ET system proposed by Ren et al. [9], which uses a ranking based loss function. The model uses hand-crafted features and does not assume labels to be noise-free. To model label noise, the model uses a partial-label loss function and also models label-label correlation.

AFET-NoCo: A variant of the AFET model which does not use label-label correlation as described in Ren et al. [9]

AFET-CoH: A variant of the AFET model, which uses label-label correlation based on the label hierarchy, as described in Ren et al. [9].

Attentive: An attention mechanism based deep neural network model proposed by Shimaoka et al [54]. The model assumes that the training dataset is noise-free and automatically learns feature representations of the input.

We compare these baselines with variants of our proposed model: (1) **our**: the complete proposed model; (2) **our-AllC:** a model which assumes that all mentions are *clean*; (3) **our-NoM:** a model without mention representation component.

Experimental setup

For the evaluation of learning models, we use the strict (subset accuracy), and the harmonic mean (F1 values) of the precision and recall computed using the loose macro and loose micro evaluation metrics as described in Section 2.3. The existing methods for the Fine-ET task use the same measures [9, 17, 53, 54]. We removed the entity mentions that do not have any label in the training as well as the test set. We also remove entity mentions that have spurious indices, that is, entity mention length of 0. For all the three datasets, we randomly sampled 10% of the test set, and use it as a development set, on which we tune model parameters. The remaining 90% is used for final evaluation. For all our experiments, we train each model using the same hyperparameters five times and report their performance in terms of the loose-micro-F1 score on the development set, as shown in Figure 3.5. On the FIGER dataset, we observed a large variance in performance as compared to the other two datasets. This might be because the FIGER dataset has a very small development set. From each of these five runs, we pick the best performing model based on the development set and report its result on the test set.

Hyperparameter setting: All the deep neural network models mentioned in this chapter used 300-dimensional pre-trained word embeddings distributed by Pennington et al. [72]. The hidden layer size of word-level bi-directional LSTM was 100, and that of character-level LSTM was 200. We randomly initialized character embeddings of size 200 and updated the embeddings during model training. We use dropout with the probability of 0.5 on the output of LSTM encoders. The embedding dimension used was 500. We use Adam [70] as optimization method with a learning rate of 0.0005 to 0.001 and mini-batch size in the range of 800 to 1500. The proposed model and some of the baselines were implemented using the TensorFlow[2] framework.

3.5.3 Transfer learning

In the feature level transfer learning, we use the best performing proposed model trained on the FIGER dataset to generate representations, that is, D_f dimensional vector for every entity mention present in the train, development, and test set of the D-BBN and the D-ONTONOTES dataset. Figure 3.4 illustrates an example of the encoding process. Then we use these representations as a feature vector in place of the user-defined features and train the AFET model. Its hyperparameters were tuned on the development set. These results are shown in Table 3.2 as **feature level transfer-learning**.

[2] http://tensorflow.org/

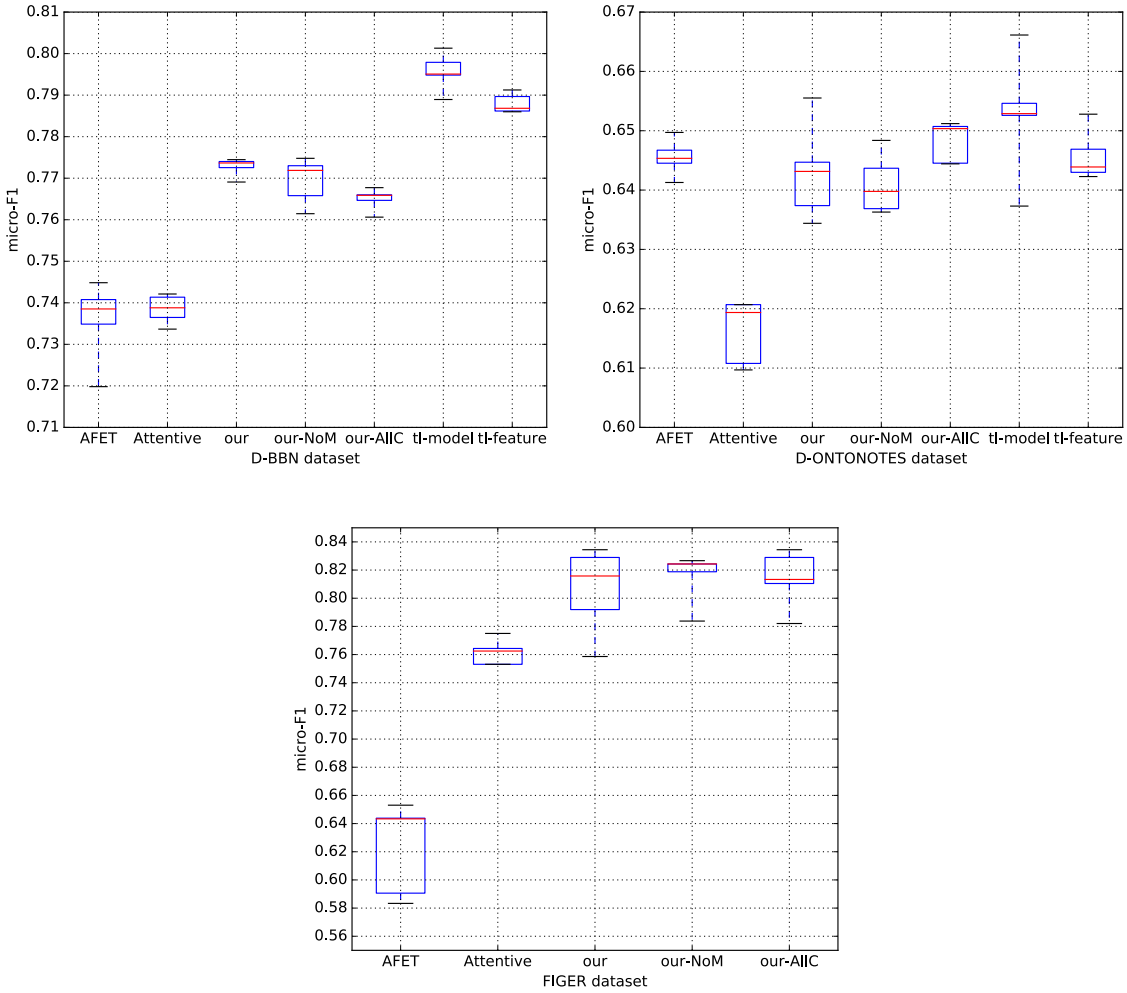


Figure 3.5: The performance of different models on the validation set illustrated using a box-whiskers plot. The red line, boxes, and whiskers indicate the median, quartiles, and range.

In the model level transfer learning, we use the learned weights of LSTM encoders from the best performing proposed model trained on the FIGER dataset and initialize the LSTM encoders of the same model with these weights while training on the D-BBN and the D-ONTONOTES datasets. These results are shown in Table 3.2 as **model level transfer learning**.

3.5.4 Performance comparison and analysis

The results obtained by the proposed model, its variants, and the baselines are listed in Table 3.2.

Comparison with other feature learning methods: The proposed model and its vari-

Typing methods	FIGER			D-ONOTNOTES			D-BBN		
	S-Acc	L-Ma-F1	L-Mi-F1	S-Acc	L-Ma-F1	L-Mi-F1	S-Acc	L-Ma-F1	L-Mi-F1
FIGER[*] [17]	0.474	0.692	0.655	0.369	0.578	0.516	0.467	0.672	0.612
HYENA[*] [18]	0.288	0.528	0.506	0.249	0.497	0.446	0.523	0.576	0.587
AFET-NoCo[*] [9]	0.526	0.693	0.654	0.486	0.652	0.594	0.655	0.711	0.716
AFET-CoH[*] [9]	0.433	0.583	0.551	0.521	0.680	0.609	0.657	0.703	0.712
AFET[*] [9]	0.533	0.693	0.664	0.551	0.711	0.647	0.670	0.727	0.735
AFET[†‡] [9]	0.509	0.689	0.653	**0.553**	**0.712**	**0.646**	0.683	0.744	0.747
Attentive[†] [54]	0.581	0.780	0.744	0.473	0.655	0.586	0.484	0.732	0.724
our-AllC[†]	**0.662**	0.805	0.770	0.514	0.672	0.626	0.655	0.736	0.752
our-NoM[†]	0.646	0.808	0.768	0.521	0.683	0.626	0.615	0.742	0.755
our[†]	0.658	**0.812**	**0.774**	0.522	0.685	0.633	0.604	0.741	0.757
model level transfer-learning[†]	—	—	—	0.531	0.684	0.637	0.645	0.784	**0.795**
feature level transfer-learning[†]	—	—	—	0.471	0.689	0.635	**0.733**	**0.791**	0.792

Table 3.2: The performance analysis of the proposed Fine-ET method and its baselines evaluated on the D-BBN, D-ONTONOTES, and FIGER datasets.

ants (**our-AllC, our-NoM**) perform better than the existing feature learning method by Shimaoka et al. [54] (**Attentive**), consistently on all datasets. The performance gain indicates the benefits of the proposed representation scheme and joint learning of representation and label embedding.

Comparison with feature engineering methods: The proposed model performs better than the existing feature engineered methods (**FIGER**, **HYENA**, **AFET-NoCo**, **AFET-CoH**) consistently across all datasets on loose-micro-F1 and loose-macro-F1 evaluation metrics. These methods do not model label-label correlation based on data. In comparison with **AFET**, the proposed model outperforms AFET on the FIGER and D-BBN datasets in terms of the loose-micro-F1 evaluation metric. This indicates the benefits of feature learning as well as data-driven label-label correlation. We make a type-wise performance comparison on the D-ONTONOTES dataset in Subsection 3.5.5 and found that the data-driven label-label correlation only helps in classifying entity mentions of miscellaneous types.

Comparison with variants of our model: The proposed model performs better on all dataset as compared to **our-AllC** in terms of loose-micro-F1 score. However, we find the performance difference on the FIGER and the D-ONTONOTES datasets is not statistically significant. We investigated it further and found that across all three datasets, there exist only a few entity types for which more than 85% of entity mentions are noisy. These types consist of approximately 3−4% of the test set, and our model fails on these types (zero loose-micro-F1 scores). However, **our-AllC** performs relatively well on these types. Examples

[*]These results are from Ren et al. [9] that also use 10% of the test set as a development set and the remaining for evaluation.

[‡]We used the publicly available code distributed by Ren et al. [9].

[†]All of these results are on the same train, development, and test set.

	FIGER				D-ONTONOTES				D-BBN			
	Support		L-Mi-F1		Support		L-Mi-F1		Support		L-Mi-F1	
	Train	Test	PM	AFET	Train	Test	PM	AFET	Train	Test	PM	AFET
Level 1	56.1%	74.7%	0.823	0.728	40.7%	71%	0.719	0.752	66.2%	58.9%	0.795	0.788
Level 2	43.9%	25.3%	0.605	0.5	42.8%	26.3%	0.333	0.23	33.8%	41.1%	0.692	0.683
Level 3	-	-	-		16.5%	2.7%	0.146	0.078	-	-	-	-

Table 3.3: The loose-micro-F1 scores of the proposed model (PM) and AFET at different hierarchy levels for the FIGER, D-ONTONOTES, and D-BBN datasets. Also, the percentage support of corresponding training and testing instances is mentioned.

of such types are: */building*, */person/political_figure*, */GPE/STATE_PROVINCE*. The analysis indicates two limitations of the proposed model. First, the separating of clean and noisy mentions based on the hierarchy has its inherent limitation of assuming labels within a path are correct. Second, our model learns better if more clean examples are available at the cost of not learning very noisy types. Compared with **our-NoM**, the proposed model performs slightly better across all datasets in terms of loose-micro-F1 score.

Feature level transfer learning analysis: In the feature level transfer learning, replacing the hand-crafted features in the AFET model (D-BBN dataset), with the features learned by the proposed model, increases the performance by 5%, 4.7%, and 4.5%, in terms of strict (subset accuracy), loose-macro-F1 and loose-micro-F1 scores, respectively. The improvement indicates the usefulness of the learned feature representations. However, if we repeat the same process with the D-ONTONOTES dataset, there is only a subtle change in performance. This is majorly because the data distribution of the D-ONTONOTES dataset is different from that of the FIGER dataset. This issue is discussed in the next subsection.

Model level transfer learning analysis: In the model level transfer learning, sharing knowledge from a similar dataset (FIGER to D-BBN) increases the performance by 4.1%, 4.3%, and 3.8% in terms of strict (subset accuracy), loose-macro-F1 and loose-micro-F1 scores, respectively. However, sharing knowledge from the FIGER to the D-ONTONOTES dataset slightly increases the performance by 0.4% in terms of the loose-micro-F1 score.

Level wise analysis: Table 3.3 reports the loose-micro-F1 score of the proposed model at a different level of the type hierarchy. For example, in the D-ONTONOTES hierarchy, *person* type is at level 1, *artist* type is at level 2 and *actor* type is at level 3. From the results, we can observe that consistently on all three datasets, the performance of the model is better on the levels up in the hierarchy. Also, the proposed model consistently outperforms AFET, on all hierarchy levels.

From the results in Table 3.3, we can also observe that as the hierarchy level increases,

his	thousands of angry people
A reporter	export competitiveness
Freddie Mac	Messrs. Malson and Seelenfreund
the numbers	Hollywood and New York
his explanation	April
volatility	This institution
their hands	the 1987 crash
it	January 4th
Macau	investment enterprises
France	any means

Table 3.4: 20 randomly sampled entity mentions present in the test set of D-ONTONOTES dataset.

the number of training instances decreases. Thus the models have to learn from fewer training instances for labels at depth two and three. Moreover, training and test data distributions are not similar, especially for the FIGER and the D-ONTONOTES datasets. In these datasets, the testing data is even more skewed towards the top level than the training data. For example, the support of test data at level one is 18.6% and 30.3% more than the training data of the FIGER and D-ONTONOTES datasets, respectively. Due to less support in training data and even less support in test data for fine-labels, the models perform poorly on levels two and three on the FIGER and D-ONTONOTES datasets. Whereas in the BBN dataset, the difference in support among the test and train data is less, so as the difference in fine-grained label performance.

3.5.5 Case analysis: D-ONTONOTES dataset

We observed three things: (i) all models perform relatively poor on the D-ONTONOTES dataset compared to their performance on the other two datasets; (ii) the proposed model outperforms other models including AFET on the other two datasets but gave a worse performance on the D-ONTONOTES dataset; (iii) the two variants of transfer learning significantly improve the performance of the proposed model on the D-BBN dataset but resulted in only a subtle performance change on the D-ONTONOTES dataset.

Statistics of the datasets (Table 3.1) indicate that the presence of pronominal or other kinds of mentions is relatively higher in the D-ONTONOTES dataset (6.78% in the test set) than the other two datasets (0% in the test set). Examples of such mentions are *100 people*, *It*, *the director*, etc. Table 3.4 shows 20 randomly sampled entity mentions from the test set of the D-ONTONOTES datasets. Some of these mentions are very generic and likely to be

Label type	Support	The Proposed Model			AFET		
		P	R	F1	P	R	F1
/other	42.6%	0.838	0.809	0.823	0.774	0.962	**0.858**
/organization	11.0%	0.588	0.490	**0.534**	0.903	0.273	0.419
/person	9.9%	0.559	0.467	**0.508**	0.669	0.352	0.461
/organization/company	7.8%	0.932	0.166	**0.282**	1.0	0.127	0.225
/location	7.5%	0.687	0.796	**0.737**	0.787	0.609	0.687
/organization/government	2.1%	0	0	0	0	0	0
/location/country	2.0%	0.783	0.614	**0.688**	0.838	0.498	0.625
/other/legal	1.8%	0	0	0	0	0	0
/location/city	1.8%	0.919	0.610	**0.733**	0.816	0.637	0.715
/person/political_figure	1.6%	0	0	0	0	0	0

Table 3.5: The performance analysis of the proposed model and AFET on top 10 (in terms of type frequency) types present in the D-ONTONOTES dataset.

dependent on previous sentences. As all the methods use features solely based on the current sentence, they fail to transfer cross-sentence boundary knowledge. Removing pronominal mentions from the test set increases the performance of all feature learning methods by around 3%.

Next, we analyze where the proposed model is failing as compared to the AFET model. For this, we look at type-wise performance for the top-10 most frequent types in the D-ONTONOTES test dataset. The results are shown in Table 3.5. Compared to AFET, the proposed model performs better in all types except *other* in the top-10 frequent types. The *other* type, which is dominant in the test set (42.6% of entity mentions are of type *other*) and is a collection of multiple broad subtypes such as *product*, *event*, *art*, *living_thing*, *food*. The performance of AFET significantly drops (*AFET-NoCo*) when data-driven label-label correlation is ignored, which indicates that modeling data-driven correlation helps. However, as shown in Figure 3.2a, the use of label-label correlation depends on appropriate values of parameters that vary from one dataset to another.

3.6 Conclusion

In this chapter, we propose a deep neural network model for the Fine-ET task. The proposed model learns representations of entity mention, its context and incorporates label noise information in a variant of a non-parametric hinge loss function. Experiments show that the proposed model outperforms existing state-of-the-art models on two publicly available datasets without explicitly tuning data-dependent parameters.

Our analysis indicates the following observations. First, the D-ONTONOTES dataset has a different distribution of entity mentions compared with the other two datasets. Second, if the data distribution is similar, then transfer learning is very helpful. Third, incorporating data-driven label-label correlation helps in the case of labels of mixed types. Fourth, there is an inherent limitation in assuming all labels to be clean if they belong to the same path of the hierarchy. Fifth, the proposed model fails to learn very noisy label types.

The analysis in this chapter also highlights a need to have a new dataset for the Fine-ET task, which can overcome some of the issues associated with existing datasets. We work in this direction in Chapter 5. Moreover, all existing datasets use a KB to assign labels to entity mentions in the distant supervision paradigm. We explore a different direction in Chapter 4 to build a Fine-ET system using multiples label sources.

4

Collective Learning Framework for Fine-ET

Chapter Highlights

- There are multiple diverse datasets available for the Fine-ET task.
- These datasets differ in the label set or the text-domain/source or both.
- The objective of this chapter is to build learning models that can predict the best possible label for entity mentions present in all available text-domain/source.
- The best possible label need not be present in the same domain dataset.
- We formulate this problem setting as ET-in-the-wild and propose a collective learning framework for this task.
- We also propose a set of evaluation schemes and metrics for the ET-in-the-wild task.
- The proposed framework outperforms competitive baselines with a significant margin in an evaluation setting containing seven diverse datasets.
- This chapter is based on the publication "Collective Learning From Diverse Datasets for Entity Typing in the Wild" presented at the EYRE workshop at CIKM 2019.

4.1 Abstract

The distant supervision method for creating training dataset for Fine-ET task mostly depends on Wikipedia as a text source and a KB as entity type source. However, many times

we need to build Fine-ET systems for text sources other than Wikipedia, with some types not present in any KB. In this chapter, we focus on this research direction, in particular, a more generalized scenario, which we refer to as Entity Typing in the wild task. In this task, there are n text sources, each annotated with some type or label set $\mathcal{Y}_i$. The $\mathcal{Y}_i$'s are neither disjoint nor identical, or in other words, they have a partial overlap. The objective is to build a Fine-ET system, which can predict entity type from the union on all $\mathcal{Y}_i$'s across all n text sources, without knowing about test instance text source or candidate entity type beforehand.

In this chapter, we describe the ET-in-the-wild problem setting and propose a collective learning framework for this problem. The framework 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 the proposed framework. The source code and the implementation details of this chapter are available at `http://github.com/abhipec/ET_in_the_wild`.

4.2 Introduction

The 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, a 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 [9, 17]. Figure 4.1 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 that 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 [51] and knowledge base completion [2], where ET models are used.

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

	Entity Mentions along with the context		Learning Models				
	Context	Entity Mention	M1†	M2‡	M3*	M4ǂ	Objective ♁
CoNLL	Former **Wallaby** captain **Nick Farr-Jones** believes ...	Wallaby	ORG	ADR	ORG	Disease	Sports team
		Nick Farr-Jones	PER	ADR	PER	Chemical	Athlete
BC5CDR	… an ability to reduce **cocaine** induced **seizures** without ...	cocaine	MISC	Drug	Food	Chemical	Drug
		seizures	MISC	ADR	Medicine	Disease	ADR

†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}.
♁Objective of this work is to predict the finest-possible label irrespective of the dataset.

Figure 4.1: The table illustrates the output of four learning models on typing four entity mentions. For example, the model M1 trained on the CONLL dataset assigns ORG type to the entity mention Wallaby, which is from the same dataset.

illustrated in Figure 4.1 (output with dashed line border). These labels may not be present in the same domain dataset. For example, the model should predict the label *sports team* for the entity mention Wallaby, even if the finest possible label (*sports team*) is not present in the same domain CONLL dataset [32]. 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, collective learning from diverse datasets is a challenging problem. Figure 4.2 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 [73] contains informally written sentences from a medical forum, whereas the JNLPBA dataset [74] 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 labels of these datasets. For example, both CONLL and FIGER [17] datasets have a label *person*. However, only the FIGER dataset has a label *athlete*, a subtype of *person*. This means that the CONLL dataset can also contain *athlete* mentions but were only annotated with a coarse label *person*. Thus, learning collectively from these diverse datasets 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 (CLF) for the ET in the wild problem. CLF first builds a unified hierarchical label set (UHLS) and associated label

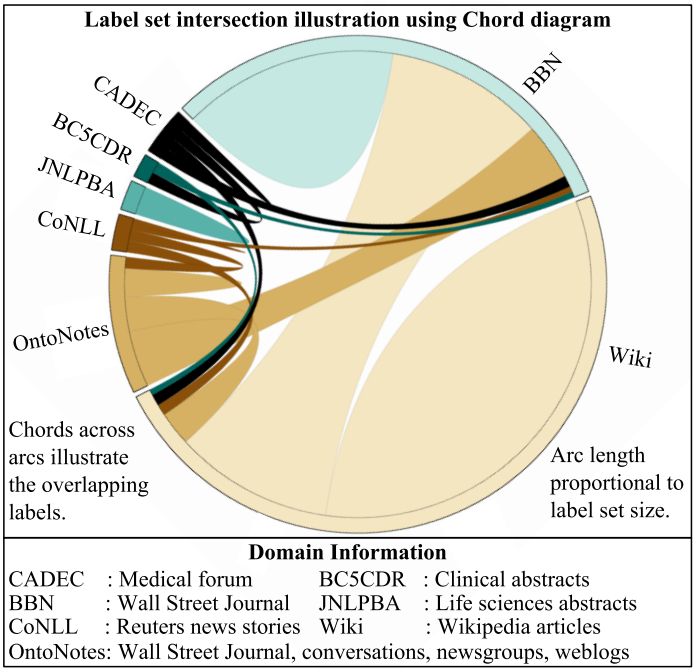


Figure 4.2: An illustration of the diversity of the seven ET datasets in their label set and domain using the chord diagram. The arc length is proportional to the number of labels in these datasets. The chords that connect arcs of different datasets illustrate the label overlap proportion.

mapping by pooling labels from diverse datasets. Then, a single classifier 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 the 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 the finest label.

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 [68] based encoder to learn a useful representation of the text followed by a partial hierarchical loss function [75] for label classification. This setup enables a single neural network classifier to predict fine-grained labels across all domains, even though the

finest 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 the learning model's performance on a test set, which is formed by merging test instances of seven diverse datasets. To excel on this merged test set, learning models must generalize beyond a single dataset. Our evaluation metrics are designed to measure the learning model's performance to predict the finest possible 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 (Section 4.3).

2. We propose a novel collective learning framework that makes it possible to train a single classifier on an amalgam of diverse ET datasets, enabling finest prediction across all the datasets, i.e., a generalized model for ET task as a whole (Section 4.4).

3. We propose evaluation schemes and evaluation metrics to compare learning models for the ET in the wild problem setting (Sections 4.5.5, 4.5.6).

4.3 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 different 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.

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 finest label possible from the set of all labels, $\mathbb{Y} = \bigcup_{i=1}^{n} \{\mathcal{Y}_i\}$, for a test entity mention. The finest possible label might not be present in any in-domain dataset.

4.4 Collective Learning Framework (CLF)

Figure 5.2a 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 $\mathcal{Y}_u \subseteq \mathbb{Y}$. In $\mathcal{T}_u$, the fine-grained labels are present at the leaf level of the hierarchy, and non-leaf nodes represent coarse labels (Section 4.4.1).

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 the same as the label space of the union of the mapped nodes (Section 4.4.1).

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 (Sections 4.4.2, 4.4.2).

4.4.1 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 the literature, there are several existing hierarchies, such as WordNet [76] and ConceptNet [77]. Even two ET datasets, BBN [22] and FIGER, organize labels in a hierarchy. However, none of these hierarchies can be directly used as discussed next.

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

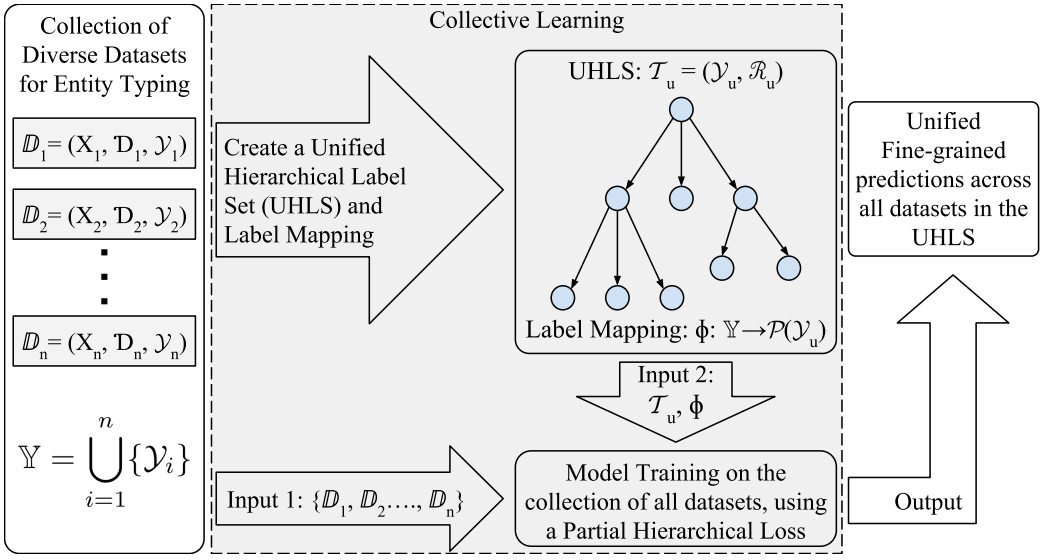


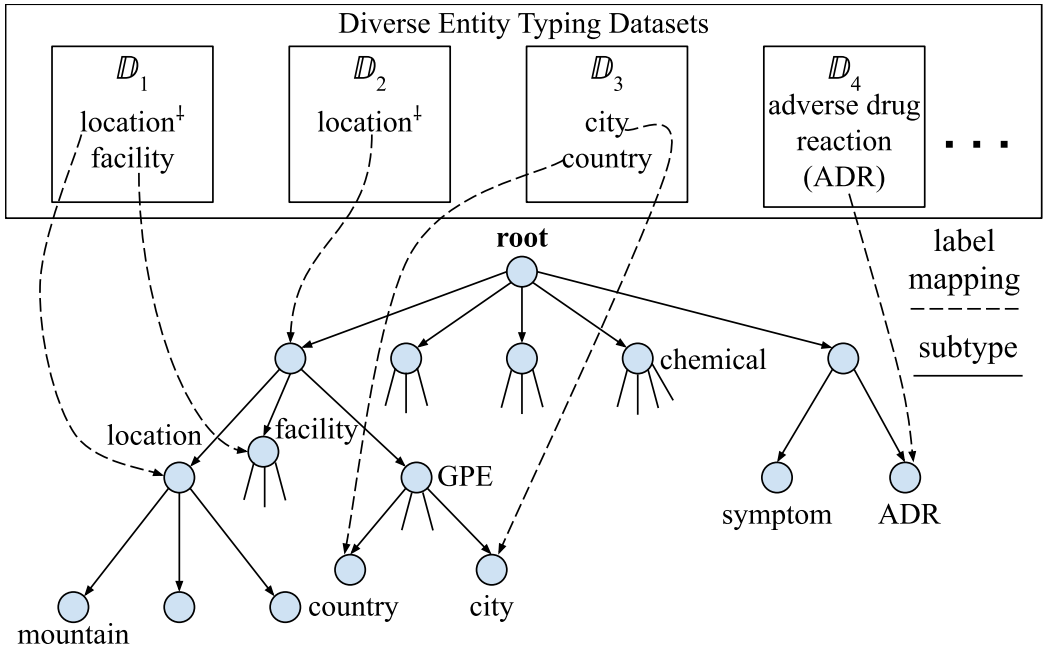
Figure 4.3: An overview of the proposed collective learning framework.

spaces are the same. For example, in the CONLL dataset, the label space for the label *location* includes facilities, whereas, in the ONTONOTES dataset [21], 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 4.4 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 [1] or WikiData [5]).

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, and if the label



‡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 4.4: A simplified illustration of the UHLS and the label mapping from individual datasets.

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 the 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 that 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, all of the decisions related to comparison of two label spaces, are made by a domain expert. For example, is the label space of the label *person* from the CONLL dataset the same as the label space of the label *person* from the FIGER dataset? The

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.

Initialize: $\mathcal{Y}_u = \{root\}, \mathcal{R}_u = \{\}$
for $y \in \mathbb{Y}$ **do**
  if $\exists \mathcal{S} \subseteq \mathcal{Y}_u$ *s.t.* $\mathcal{L}(y) == \mathcal{L}(\mathcal{S})$ **then** // Case 2
    $\phi(y) \mapsto \mathcal{S}$
  else // Case 1
    $v = \underset{size(\mathcal{L}(v))}{\arg\min} \{v \mid v \in \mathcal{Y}_u \,\&\, \mathcal{L}(y) \prec \mathcal{L}(v)\}$
    $\mathcal{Y}_u = \mathcal{Y}_u \cup \{y\}$
    $\mathcal{R}_u = \mathcal{R}_u \cup \{(y, v)\}$
    $\phi(y) \mapsto \{y\}$
    for $(x, v) \in \mathcal{R}_u$ **do** // Update existing nodes
      if $x \neq y \,\&\, \mathcal{L}(x) \prec \mathcal{L}(y)$ **then**
        $\mathcal{R}_u = \mathcal{R}_u - \{(x, v)\}$
        $\mathcal{R}_u = \mathcal{R}_u \cup \{(x, y)\}$
    for $\hat{v} \in \mathcal{Y}_u$ **do** // Restrict to tree hierarchy
      if $\mathcal{L}(\hat{v}) \prec \mathcal{L}(y) \,\&\, \hat{v} \notin subtree(y)$ **then**
        $\phi(y) \mapsto \{\hat{v}\}$

Algorithm 1: UHLS and label mapping creation algorithm.

answer to this question is that their label space is not the same. In the CONLL dataset, entity mentions such as the name of various gods are assigned type *person*, whereas, in the FIGER dataset, they have a separate type and are not assigned the type *person*. Thus the label *person* in these datasets do not have the same label space. The expert makes the decision based on the annotation guidelines for the queried labels and using an 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 finer prediction across domains. An illustration of UHLS and label mapping is provided in Figure 4.4.

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

4.4.2 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.

Neural Mention and Context Encoder

The input to our model is a sentence with the start and end index of the entity mentions. Following our previous work (Chapter 3), we use Bi-directional LSTMs [69] 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.

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 $\mathcal{Y}_m$[1]. 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 the 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*, ...}. Additional, some labels have a one-to-many mapping, for example, for the label *MISC* in the 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 the model to predict finer labels across datasets.

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

Partial Hierarchical Label Loss

A partial label loss deals with a situation where a training example has a set of candidate labels and among which only a subset is correct for a given example [78–80].

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 [75]. We first compute the probability distribution for the labels available in $\mathcal{Y}_u$ as described in equation 4.1. Here W is a weight matrix of size $|R| \times |\mathcal{Y}_u|$, b is the bias variable of size $|\mathcal{Y}_u|$, and x is the input entity mention along with its context.

$$p(y|x) = softmax(RW + b) \tag{4.1}$$

Then we compute $\hat{p}(y|x)$, a distribution adjusted to include a weighted sum of the ancestor's probability for each label as defined in equation 4.2. Here $\mathcal{A}_t$ is the set of ancestors of the label y in $\mathcal{R}_u$, and β is a hyperparameter.

$$\hat{p}(y|x) = p(y|x) + \beta * \sum_{t \in \mathcal{A}_t} p(t|x) \tag{4.2}$$

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.

4.5 Experiments and Analysis

4.5.1 Datasets

Table 4.1 describes the seven datasets used in this chapter. 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 FIGER [17] dataset is automatically generated using distant supervision process [15] and has multiple labels per

Dataset	Domain	No. of Labels	Mention count	Fine labels
BC5CDR [81]	Clinical abstracts	2	9,385	No
CONLL [32]	Reuters news stories	4	23,499	No
JNLPBA [74]	Life sciences abstracts	5	46,750	Yes
CADEC [73]	Medical forum	5	5,807	Yes
ONTONOTES [21]	Newswire, conversations, newsgroups, weblogs	18	1,16,465	No
BBN [22]	Wall Street Journal text	73	86,921	Yes
FIGER [17]	Wikipedia	116	20,00,000	Yes

Table 4.1: Description of the seven ET datasets used.

entity mention in its label set. The other remaining datasets have a single label per entity mention.

4.5.2 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 a total of 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 4.2. The *MISC* label from the CONLL dataset has the highest ten number of mappings to the UHLS nodes. FIGER 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.

4.5.3 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

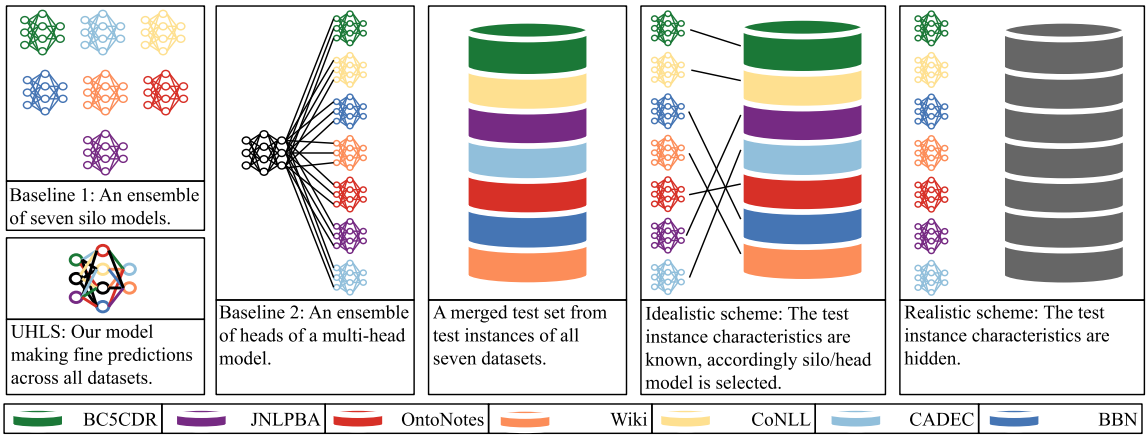


Figure 4.5: A pictorial illustration of the complete experimental setup.

this model a silo ensemble model[2]. In this ensemble model, each silo model has the same mention and context encoder structure described in Section 4.4.2. 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 [82]. In this baseline, all seven datasets are fed through a common mention and context encoder. For each dataset, there is a separate classifier head with the output labels the same as that was available in the respective original dataset. We name this baseline as a multi-head ensemble baseline[3]. 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.

4.5.4 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

[2]Here unlike traditional ensemble models, in silo ensemble, the learning models are trained on different datasets.

[3]Here since the "task" is the same, i.e., entity typing, we use the term multi-head instead of multi-task for the baseline.

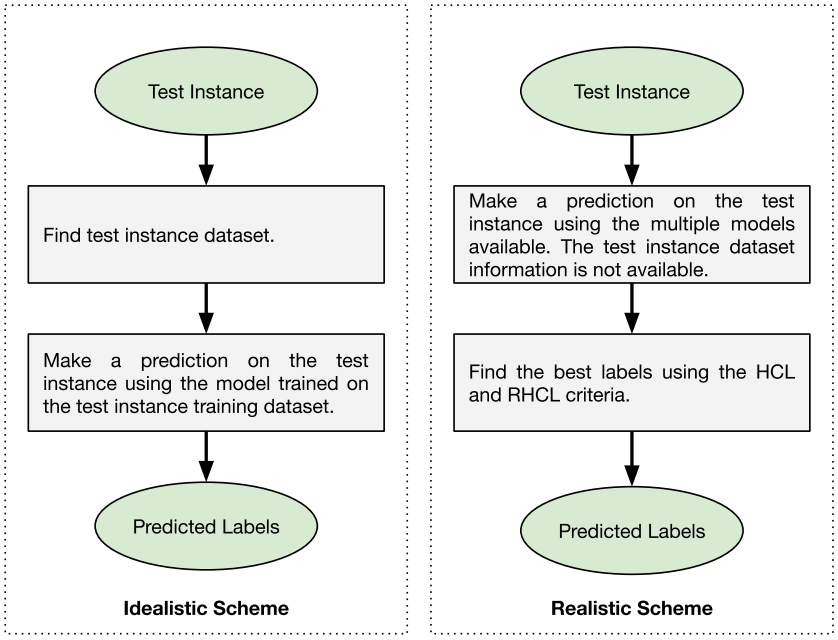


Figure 4.6: A flow-chart illustrating the workflow of the idealistic and realistic schemes.

the model on the training splits of all seven datasets together and select the best model using the combined validation split.

4.5.5 Experimental Setup

Figure 4.5 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 merged the test instances from the seven datasets listed in Table 4.1 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.

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. For example, if the test instance is from the CONLL dataset, then the silo ensemble model (or head of the multi-head ensemble model) trained on the CONLL dataset will be chosen for prediction. An illustration of this scheme is available in Figure 4.5 and a flow-chart in Figure 4.6. This scheme gives an advantage to the ensemble baselines and compares the

models in 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 a 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. An illustration of this scheme is available in Figure 4.5 and a flow-chart in Figure 4.6. 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 RHCL 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 [83], 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.

4.5.6 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,

Input: y, $\hat{y}$, $\mathcal{T}_u$, and ϕ // A true label, a predicted label, the UHLS and label mapping.
Output: t, $\hat{t}$ // The true and predicted label after modification.
$y_m = \phi(y)$, $\hat{y}_m = \phi(\hat{y})$ // The true and predicted labels mapped to the UHLS nodes.
$\mathcal{D} = \text{descendants}(y_m) \cup y_m$ // The descendants of true label in the UHLS.
if $\mathcal{D} \cap \hat{y}_m$ **then** // If the predicted label is among the descendants of true label.
    $\hat{t} = y_m$ // The predicted label is modified to be the same as true label.
    $t = y_m$
else
    $\hat{t} = \hat{y}_m$
    $t = y_m$
Return t, $\hat{t}$ // Return the modified true and predicted labels, respectively.

Algorithm 2: The procedure to convert the dataset-specific true and predicted labels to labels in UHLS on the best effort basis.

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 allow us to compare learning models making predictions in different label sets with minimum re-annotation effort.

In the first metric, we compute a loose micro F1 score on the best effort basis. It is based on the intuition that if the labels are only annotated at a coarse level (e.g. *person*) in the gold test annotations, then even if a model predicts a fine-label within that coarse label (e.g. *artist*), this metric should not penalize such cases[4]. 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. The mapping or modification process is described in Algorithm 2. After the best-effort mapping process, we can use the existing loose micro F1 score typically used to evaluate the ET task, as defined in Section 2.3.2. We compute the loose micro F1 scores 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]Exception is where the source dataset also has fine-grained labels.

4.5.7 Result and Analysis

Analysis of the idealistic scheme results

In Figure 4.7, 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 the multi-head ensemble model. Note that this performance comparison is in a setting that 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, the 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 (Section 4.5.7).

Analysis of the realistic scheme results

In Figure 4.7, we can observe that both the silo ensemble and the multi-head ensemble model perform 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 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 Wikipedia 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.

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*

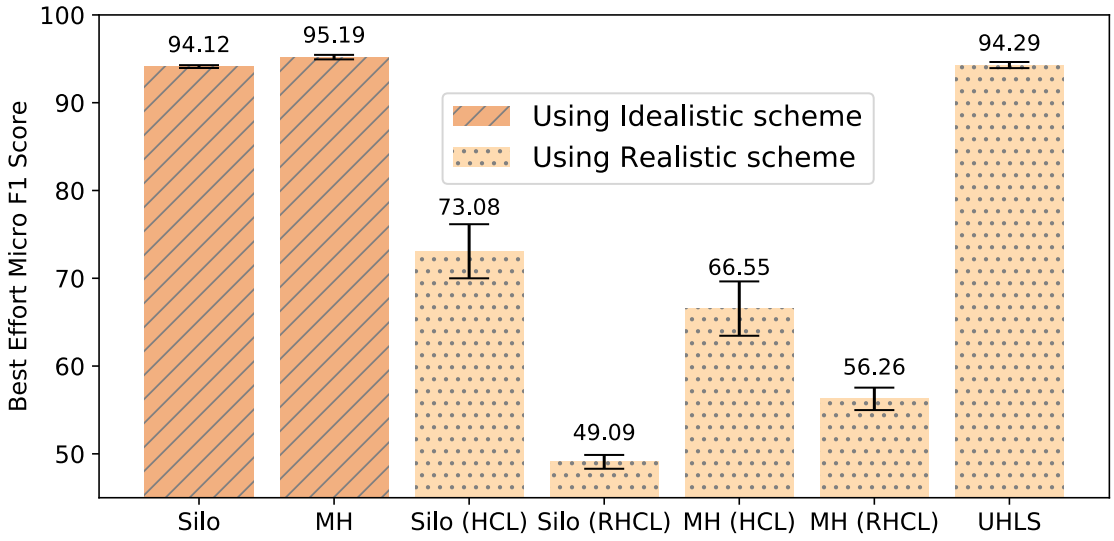


Figure 4.7: Comparison of learning models in the idealistic and realistic schemes.

event label is contributed by the FIGER dataset, whose domain is Wikipedia. The results (Figure 4.8) 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 the 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 the finest possible 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 (FIGER), and MH (FIGER) 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, the 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 includes other events such as *elections*. Similarly, the model/head with the FIGER 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

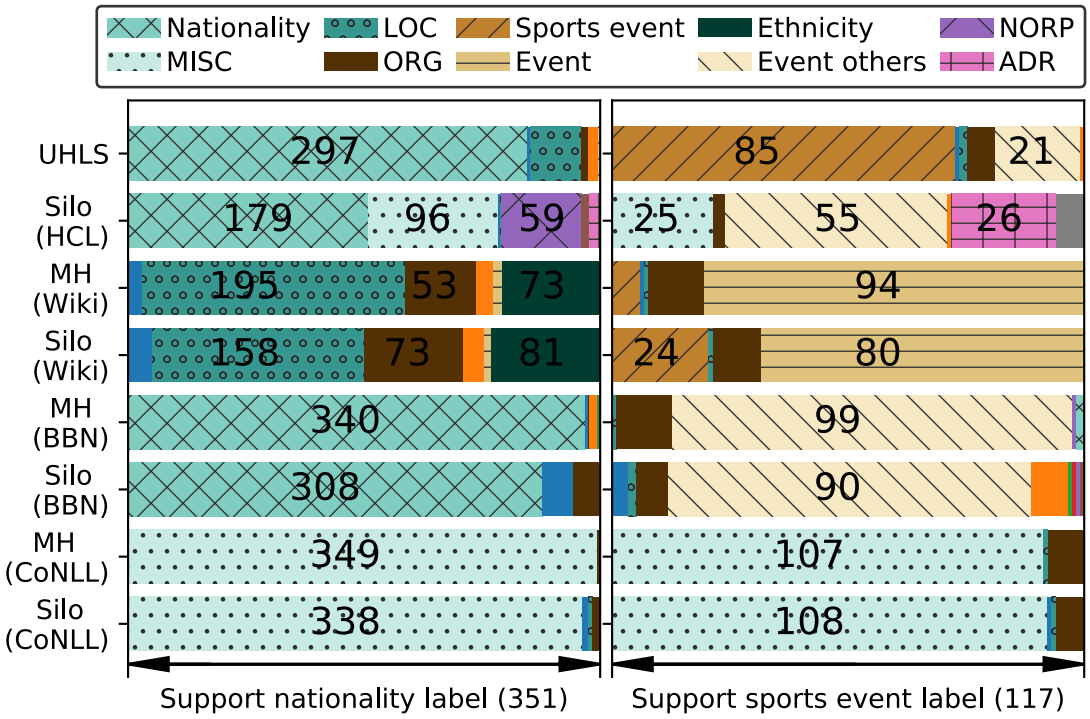


Figure 4.8: Analysis of Fine-grained label predictions. The two columns specify results for nationality and sports event label. Each row represents a model used for prediction. The results can be interpreted as, out of 351 entity mentions with type nationality, model Silo (CONLL) predicted 338 as MISC type and the remaining as other types illustrated.

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 the finest label 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.

Example output on different datasets

In Figure 4.9, 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 FIGER dataset. For example, for the entity mention `Magellan`, a label *spacecraft* is assigned. The *spacecraft* label is only present in the FIGER 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, the *ADR* label is only present in

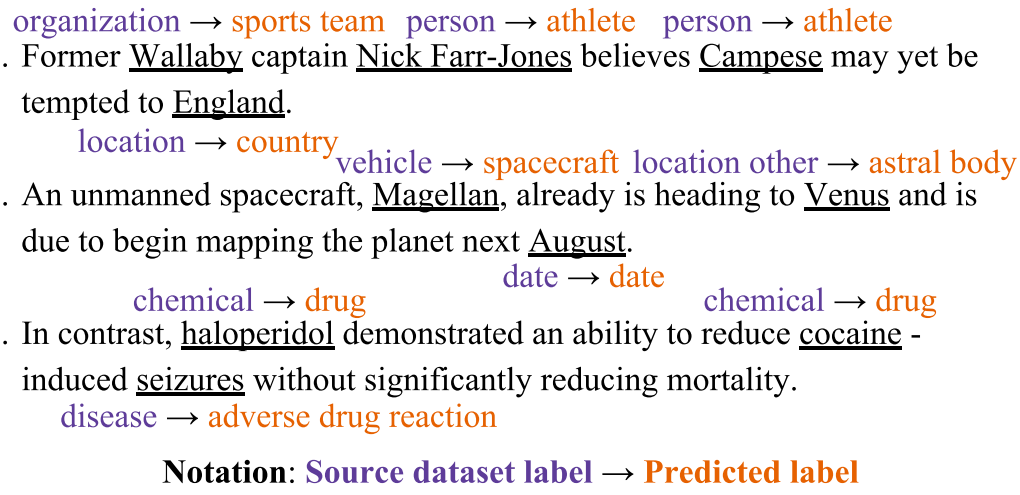


Figure 4.9: Example output of our proposed approach. Sentence 1, 2, 3 are from the CONLL, BBN and BC5CDR dataset respectively.

the CADEC dataset with the medical forum domain. The proposed approach can aggregate fine-labels across datasets and makes unified fine-grained predictions.

Result and analysis summary

The collective learning framework allows a limitation of one dataset being covered by some other dataset. 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.

4.6 Related Work

In this section, we will first describe the works which are closely related to our work, followed by work in other related areas.

To the best of our knowledge, the work of [84] in the visual object recognition task is closet to our work. They consider two datasets. First, a coarse-grained and second, a fine-grained. The 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 a universal POS tagset [85]. 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 Ren et al. [9] and Abhishek et al. [30]. In our work, we use the current state-of-the-art hierarchical partial loss function proposed by Xu et al. [75]. A comparison among these loss functions is available in [75].

4.7 Conclusion

In this chapter, 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 finest possible 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 [27] 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 [8], relation classification [86], and named entity recognition [12], 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. Some of these tasks, such as relation classification, have similar characteristics to that of the ET task, where the objective is to assign a label to a given input. For these tasks, our proposed CLF can be directly used, whereas other tasks may require suitable modifications.

5

New Datasets for the Fine-ED and Fine-ET tasks

Chapter Highlights

- We observe that when the scope of entity mentions are diverse, the existing entity detection models have a poor recall.
- The primary reason for the poor recall is the lack of annotated entity mentions from diverse categories in the existing datasets.
- We propose a heuristics allied with a distant supervision approach to automatically construct training datasets for the Fine-ED and Fine-ET tasks.
- We do an extensive evaluation of the created datasets, both intrinsically and extrinsically.
- We also release a manually annotated corpus for the evaluation of Fine-ED and Fine-ET models.
- The new manually annotated corpus has 2.7 times more entity types than the FIGER evaluation corpus.
- This chapter is based on the publication "Fine-grained Entity Recognition with Reduced False Negatives and Large Type Coverage" presented at AKBC 2019.

5.1 Abstract

In this chapter, we first provide an analysis that in the Fine-ET setting using the existing datasets, detection of entity mention becomes a limitation for supervised learning models. The analysis conveys that in the existing datasets, the boundaries of entities covering a large spectrum of entity types are not properly annotated. To address this limitation, we propose a Heuristics Allied with Distant Supervision (HAnDS) framework, which automatically constructs quality datasets suitable for both the Fine-ED and Fine-ET tasks (thus end-to-end Fine-grained Entity Recognition (Fine-ER) task). The HAnDS framework exploits the high interlink among Wikipedia and Freebase in a pipelined manner, reducing annotation errors introduced by naively using the distant supervision approach. Using the HAnDS framework, we create two datasets, one suitable for building Fine-ER systems recognizing up to 118 entity types based on the FIGER type hierarchy and another for up to 1115 entity types based on the TypeNet hierarchy. Our extensive empirical experimentation warrants the quality of the generated datasets. Along with this, we also provide a manually annotated dataset for benchmarking Fine-ER systems. The code and datasets to replicate the experiments are available at https://github.com/abhipec/HAnDS.

5.2 Introduction

In the literature, the problem of recognizing a handful of coarse-grained types such as *person*, *location*, and *organization* has been extensively studied [12, 13]. We term this as a Coarse-grained Entity Recognition (Coarse-ER) task. For Coarse-ER, there exist several datasets, including manually annotated datasets such as CoNLL [32] and automatically generated datasets such as WP2 [87]. Manually constructing a dataset for the Fine-ER task is an expensive and time-consuming process as an entity mention could be assigned multiple types from a set of thousands of types.

In recent years, one of the subproblems of Fine-ER, the Fine-ET problem has received lots of attention particularly in expanding its type coverage from a handful of coarse-grained types to thousands of fine-grained types [88, 89]. The primary driver for this rapid expansion is exploitation of cheap but fairly accurate annotations from Wikipedia and Freebase [1] via the distant supervision process [15, 16]. The Fine-ET problem assumes that the entity boundaries are provided by an oracle.

We observe that the detection of entity mentions at the granularity of Fine-ET is a bottleneck. The existing Fine-ER systems, such as FIGER [17], follow a two-step approach

in which the first step is to detect entity mentions, and the second step is to categorize the detected entity mentions. For entity detection, it is assumed that all the fine-categories are subtypes of the following four categories: *person*, *location*, *organization*, and *miscellaneous*. Thus, a model trained on the CoNLL dataset [32], which is annotated with these types, can be used for entity detection. Our analysis indicates that in the context of Fine-ER, this assumption is not valid. As a face value, the *miscellaneous* type should ideally cover all entity types other than *person*, *location*, and *organization*. However, it only covers 68% of the remaining types of the FIGER hierarchy and 42% of the TypeNet [88] hierarchy. Thus, the models trained using CoNLL data are highly likely to miss a significant portion of entity mentions relevant to automatic knowledge bases construction applications.

Our work bridges this gap between entity detection and Fine-ET. We propose to automatically construct a quality dataset suitable for the Fine-ER, i.e., both Fine-ED and Fine-ET using the proposed HAnDS framework. HAnDS is a three-stage pipelined framework wherein each stage uses different heuristics. These heuristics reduce the errors introduced via naively using the distant supervision paradigm, including but not limited to the presence of large false negatives. The heuristics are data-driven and use the information provided by hyperlinks, alternate names of entities, and orthographic and morphological features of words.

Using the HAnDS framework and the two popular type hierarchies available for Fine-ET, the FIGER type hierarchy [17], and TypeNet [88], we automatically generated two corpora suitable for the Fine-ER task. The first corpus contains around 38 million annotated entity mentions with 118 entity types. The second corpus contains around 46 million annotated entity mentions with 1115 entity types. Our extensive intrinsic and extrinsic evaluation of the generated datasets warrants its quality. As compared with existing automatically generated datasets, supervised learning models trained on our induced training datasets perform significantly better (approx 20 point improvement on the micro-F1 score). Along with the automatically generated dataset, we provide a manually annotated corpora of around a thousand sentences annotated with 117 entity types for benchmarking of Fine-ER models.

Our contributions are highlighted as follows:

- We analyzed that the existing practice of using models trained on the CoNLL dataset have poor recall for entity detection in the Fine-ET setting, where the type set spans several diverse domains (Section 5.4).
- We propose the HAnDS framework, a heuristics allied with the distant supervision

approach to automatically construct datasets suitable for Fine-ER problem, i.e., both fine entity detection and fine entity typing (Section 5.5).

- We establish the state-of-the-art baselines on our new manually annotated corpus, which covers 2.7 times more finer-entity types than the FIGER gold corpus, the current de facto Fine-ER evaluation corpus (Section 5.6).

5.3 Related Work

We divide the related work into two parts. First, we describe work related to the automatic dataset construction in the context of the entity recognition task followed by related work on noise reduction techniques in the context of automatic dataset construction task.

In the context of Fine-ER task, Ling and Weld [17] proposed to use distant supervision paradigm [16, 90] to automatically generate a dataset for the Fine-ET problem, which is a sub-problem of Fine-ER. We term this as a Naive Distant Supervision (NDS) approach. In NDS, the linkage between Wikipedia and Freebase is exploited. If there is a hyperlink in a Wikipedia sentence, and that hyperlink is assigned to an entity present in Freebase, then the hyperlinked text is an entity mention whose types are obtained from Freebase. However, this process can only generate positive annotations, i.e., if an entity mention is not hyperlinked, no types will be assigned to that entity mention. The positive-only annotations are suitable for the Fine-ET task, but it is not suitable for learning entity detection models as there are large number of false negatives (Section 5.4). This dataset is publicly available as the FIGER dataset, along with a manually annotated evaluation corpus. The NDS approach is also used to generate datasets for some variants of the Fine-ET problem such as the Corpus level Fine-Entity typing [91] and Fine-Entity typing utilizing knowledge base embeddings [92]. Much recently, Choi et al. [89] generated an entity typing dataset with a very large type set of size 10k using head words as a source of distant supervision as well as using crowdsourcing.

In the context of the Coarse-ER task, [87, 93, 94] proposed an approach for creating training datasets using a combination of bootstrapping process and heuristics. The bootstrapping was used to classify a Wikipedia article into five categories, namely *PER*, *LOC*, *ORG*, *MISC*, and *NON-ENTITY*. The bootstrapping requires initial manually annotated seed examples for each type, which limits its scalability to thousands of types. The heuristics were used to infer additional links in un-linked text, however, the proposed heuristics limit the scope of the entity and non-entity mentions. For example, one of the heuristics used mostly restricts entity mentions to have at least one character capitalized. This as-

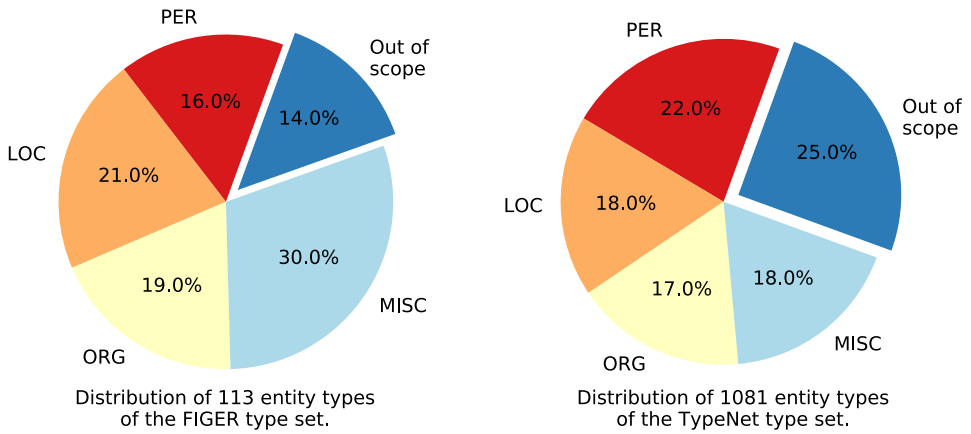


Figure 5.1: The figure illustrates the entity type coverage analysis of the FIGER and the TypeNet type set. A significant portion of entity types (out of scope portion) are not a descendant of any of the four types present in the CoNLL dataset.

sumption is not true in the context for Fine-ER, where entity mentions are from several diverse domains, including the biomedical domain.

There are other notable works which combine NDS with heuristics for generating entity recognition training dataset, such as [95] and [96]. However, their scope is limited to the application of Coarse-ER. Our work revisits the idea of automatic corpus construction in the context of Fine-ER. In the HAnDS framework, our main contribution is to design data-driven heuristics, which are generic enough to work for thousands of diverse entity types while maintaining a good annotation quality.

An automatic dataset construction process involving heuristics and distant supervision will inevitably introduce noise and its characteristics depend on the dataset construction task. In the context of the Fine-ED and Fine-ET tasks, the dominant noise is false negatives and false positives, respectively. Whereas, for the relation extraction task both false negatives and false positives noise is present [97, 98].

5.4 Case study: Entity Detection in the Fine Entity Typing Setting

In this section, we systematically analyzed existing entity detection systems in the setting of Fine-ET. We aim to answer the following question: How good are entity detection systems when it comes to detecting entity mentions belonging to a large set of diverse types? We performed two analyses. The first analysis is about the type coverage of entity detection

Models	FIGER			1k-WFB-g		
	Precision	Recall	F1	Precision	Recall	F1
LSTM-CNN-CRF (FIGER)	87.17	28.95	43.47	91.41	37.13	52.81
CoreNLP	83.82	80.99	82.38	75.46	64.12	69.33
NER Tagger	80.44	84.01	82.19	77.25	68.52	72.62

Table 5.1: The performance analysis of various entity detection models trained on existing datasets and the evaluation datasets are the FIGER and 1k-WFB-g datasets.

systems, and the second analysis is about the actual performance of entity detection systems on two manually annotated Fine-ER datasets.

5.4.1 Is the Fine-ET type set an expansion of the extensively researched coarse-grained types?

For this analysis, we manually inspected the most commonly used Coarse-ER dataset, CoNLL 2003. We analyzed how many entity types in the two popular Fine-ET hierarchies, FIGER, and TypeNet are descendent of the four coarse-types present in the CoNLL dataset, namely *person*, *location*, *organization*, and *miscellaneous*. The results are available in Figure 5.1. We can observe that in the FIGER typeset, 14% of types are not descendants of the CoNLL types. This share increases in TypeNet, where 25% of types are not descendants of CoNLL types. These types are from various diverse domains, including biomedical, legal processes, and entertainment. Recognition of entity mentions from these domains is important in the aspect of several applications including knowledge base construction. The lack of coverage of these diverse domains in the CoNLL dataset can be attributed to the fact that since 2003, the entity recognition problem has evolved a lot both in going towards finer-categorization as well as capturing entities from diverse domains.

5.4.2 How do entity detection systems perform in the Fine-ET setting?

For this analysis, we evaluate two publicly available state-of-the-art entity detection systems, the Stanford CoreNLP [49] and the NER Tagger system proposed by Lample et al. [46]. Along with these, we also train an LSTM-CNN-CRF based sequence labeling model proposed by Ma and Hovy [99] on the FIGER dataset. We evaluated the learning models on manually annotated FIGER corpus and 1k-WFB-g corpus, a new in-house developed corpus specifically for Fine-ER model evaluations. For these evaluations, we used precision, recall, and F1 metrics, the standard evaluation metrics for the ED task. A detailed description of these metrics is available in Section 2.3.1. The results are presented in Table 5.1.

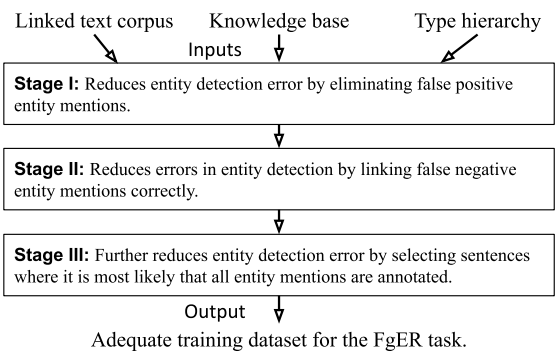
From the results, we can observe that a state-of-the-art sequence labeling model, LSTM-CNN-CRF, trained on a dataset generated using the NDS approach, such as the FIGER dataset, has lower recall compared with precision. On average, the recall is 58% lower than precision. The lower recall is primarily because the NDS approach generates positive only annotations, and the remaining un-annotated tokens contain a large number of entity mentions. Thus the resulting dataset has large false negatives.

On the other hand, learning models trained on the CoNLL dataset (CoreNLP and NER Tagger), have a much more balanced performance in precision and recall. The balanced performance is because of being a manually annotated dataset, it is less likely that any entity mention (according to the annotation guidelines) will remain un-annotated. However, the recall is much lower (16% lower) on the 1k-WFB-g corpus as on the FIGER corpus. The lower recall on the 1k-WFB-g corpus is because, when designing 1k-WFB-g, we ensured that it has sufficient examples covering 117 entity types. Whereas, the FIGER evaluation corpus has only has 42 types of entity mentions, and 80% of mentions are subtypes of *person*, *location*, and *organization* types. These results also highlight the coverage issue, mentioned in Section 5.4.1. When the evaluation set is balanced, covering a large spectrum of entity types, the performance of models trained on the CoNLL dataset goes down because of the presence of out-of-scope entity types. An ideal entity detection system should be able to work on the traditional as well as other entities relevant to the Fine-ER problem, i.e., good performance across all types. A statistical comparison of FIGER and 1k-WFB-g corpus is provided in Table 5.2.

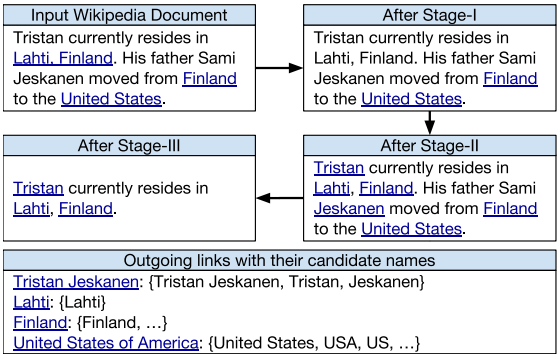
The use of CoreNLP or learning models trained on the CoNLL dataset is a standard practice to detect entity mentions in existing Fine-ER research [17]. Our analysis conveys that this practice has its limitations in terms of detecting entities belonging to diverse domains. In the next section, we will describe our approach of automatically creating training datasets for the Fine-ER task. The same learning models, when trained on our created training datasets, will have a better and a balanced precision and recall.

5.5 HAnDS Framework

The objective of the HAnDS framework is to automatically create a corpus of sentences where every entity mention is correctly detected and is being characterized into one or more entity types. The scope of entities, i.e., what types of entities should be annotated, is decided by a type hierarchy, which is one of the inputs of the framework. Figure 5.2 gives



(a) The flow-chart provides a high-level overview of the three stages of the HAnDS framework, along with each stage's objective.



(b) The top four boxes illustrate how annotations change during different stages. The bottom box illustrates the outgoing links and the candidate names for the example document.

Figure 5.2: An overview of HAnDS framework (left) along with an illustration of the framework in action on an example document (right).

an overview of the HAnDS framework.

5.5.1 Inputs

The framework requires three inputs, a linked text corpus, a knowledge base, and a type hierarchy.

Linked text corpus: A linked text corpus is a collection of documents where sporadically important concepts are hyperlinked to another document. For example, Wikipedia is a large-scale multi-lingual linked text corpus. The framework considers the span of hyperlinked text (or anchor text) as potential candidates for entity mentions.

Knowledge base: A knowledge base (KB) captures concepts, their properties, and inter-concept properties. Freebase, WikiData [5], and UMLS [100] are examples of popular knowledge bases. A KB usually has a type of property where multiple fine-grained semantic types/labels are assigned to each concept.

Type hierarchy: A type hierarchy ($\mathcal{T}$) is a hierarchical organization of various entity types. For example, an entity type *city* is a descendant of a type *geopolitical entity*. There have been various hierarchical organization schemes of fine-grained entity types proposed in the literature, which includes, a 200 type scheme proposed by Sekine [101], a 113 type scheme proposed by Ling and Weld [17], an 87 type scheme proposed by Gillick et al. [63], and a 1081 type scheme proposed by Murty et al. [88]. However, in our work, we use two such hierarchies, FIGER[1] and TypeNet. FIGER being the most extensively used hierarchy

[1]Based on our observations, we made a few changes to the original FIGER hierarchy (seven additions,

and TypeNet being the latest and largest entity type hierarchy.

5.5.2 The three stages of the HAnDS framework

Automatic corpora creation using distant supervised methods inevitably will contain errors. For example, in the context of Fine-ER, the errors could be at annotating entity boundaries, i.e., entity detection errors, or assigning an incorrect type, i.e., entity linking errors or both. The three-step process in our proposed HAnDS framework tries to reduce these errors.

Stage-I: Link categorization and Preprocessing

The objective of this stage is to reduce false positives entity mentions, where an incorrect anchor text is detected as an entity mention. To do so, we first categorize all hyperlinks of the document being processed as *entity links* and *non-entity links*. Further, every link is assigned a tag of being a *referential link* or not.

Entity links: These are a subset of links whose anchor text represents the candidate entity mentions. If the labels obtained by a KB for a link, belongs to $\mathcal{T}$, we categorize that link as an entity link. Here, the $\mathcal{T}$ decides the scope of entities in the generated dataset. For example, if T is the FIGER type hierarchy, then the hyperlink photovoltaic cell is not an entity link as its labels obtained by Freebase is not present in T. However, if T is the TypeNet hierarchy, then the hyperlink photovoltaic cell is an entity link of type *invention*.

Non-entity links: These are a subset of links whose anchor text does not represent an entity mention. Since knowledge bases are incomplete, if a link is not categorized as an *entity link*, it does not mean that the link will not represent an entity. We exploit corpus level context to categorize a link as a *non-entity link* using the following criteria: across the complete corpus, the link should be mentioned at least 50 times (support threshold) and at least 50% of times (confidence threshold) with a lowercase anchor text. The intuition of this criteria is that we want to be sure that a link represents a non-entity. For example, this heuristic categorizes RBI as a *non-entity link* as there is no label present for this link in Freebase. Here RBI refers to the term "run batted in", which is frequently used in the context of baseball and softball. In contrast, Nothman et al. [93] discard non-entity mentions having capitalized words, whereas, our data-driven heuristics does not put any such hard constraints.

Referential links: A link is said to be referential if its anchor text has a direct case-insensitive match with the list of allowed candidate names for the linked concept. A KB

one correction, one merger, one deletion, and one substitute).

can provide such a list. For example, for an entity `Bill Gates`, the candidate names provided by Freebase include `Gates` and `William Henry Gates`. However, in Wikipedia, there exists hyperlinks such as Bill and Melinda Gates linking to Bill Gates page, which is erroneous as the hyperlinked text is not the correct referent of the entity `Bill Gates`.

After the categorization of links, except for *referential entity links*, we unlink all other links. Unlinking non-referential links such as Bill and Melinda Gates reduce *entity detection errors* by eliminating false positive entity mentions. The unlinked text span or a part of it can be referential mention for some other entities, as in the above example, `Bill` and `Melinda Gates`. Figure 5.2b also illustrates this process where Lahti, Finland, gets unlinked after this stage. The next stage tries to re-link the unlinked tokens correctly.

Stage-II: Infer additional links

The objective of this stage is to reduce false-negative entity mentions, where an entity mention is not annotated. The false negatives can be reduced by linking the correct referential name of the entity mention to the correct node in KB.

To reduce entity linking errors, we use the document level context by restricting the candidate links (entities or non-entities) to the outgoing links of the current document being processed. For example, in Figure 5.2b, while processing an article about a Finnish-American luger Tristan Jeskanen, it is unlikely to observe mention of a 1903 German novel having the same name, i.e., Tristan.

To reduce false-negative entity mentions, we construct two trie trees capturing the outgoing links and their candidate referential names for each document. The first trie contains all links, and the second trie only contains links of entities which are predominantly expressed in lowercase phrases[2] (e.g., names of diseases). For each non-linked uppercase character, we match the longest matching prefix string within the first trie and assign the matching link. In the remaining non-linked phrases, we match the longest matching prefix string within the second trie and assign the matching link. Linking the candidate entities in unlinked phrases reduce entity detection error by eliminating false negative entity mentions.

Unlike Nothman et al. [93], the two-step string matching process ensures the possibility of a lowercase phrase being an entity mention (e.g., *lactic acid*, *apple juice*, *bronchoconstriction*, etc.) and a word with a first uppercase character being a non-entity (e.g., *Jazz*, *RBI*,[3] etc.).

[2]More than 50% of anchor text across corpus should be a lowercase phrase.
[3]A run batted in (RBI) is a statistic in baseball and softball.

Figure 5.2b shows an example of the input and output of this stage. In this stage, the phrases *Tristan*, *Lahti*, *Finland*, and *Jeskanen* get linked.

Stage-III: Sentence selection

The objective of this stage is to reduce entity detection errors further. This stage is motivated by the incomplete nature of practical knowledge bases. KBs do not capture all entities present in a linked text corpus and do not provide all the referential names for an entity mention. Thus, after stage-II there will be still a possibility of having both types of entity detection errors, false positives, and false negatives.

To reduce such errors in the induced corpus, we select sentences where it is most likely that all entity mentions are annotated correctly. The resultant corpora of selected sentences will be our final dataset. To select these sentences, we exploit sentence-level context by using POS tags and a list of the frequent sentence starting words. We only select sentences where all unlinked tokens are most likely to be a non-entity mention. If an unlinked token has capitalized characters, then it likely to be an entity mention. We do not select such sentences, except in the following cases. In the first case, the token is a sentence starter, and is either in a list of frequent sentence starter word[4] or its POS tag is among the list of permissible tags[5]. In the second case, the token is an adjective, or belongs to occupational titles or is a name of day or month.

Figure 5.2b shows an example of the input and output of this stage. Here only the first sentence of the document is selected because, in the other sentence, the name **Sami** is not linked. The sentence selection stage ensures that the selected sentences have high-quality annotations. We observe that only around 40% of sentences are selected by stage III in our experimental setup. We provide an analysis of several characteristics of the discarded and retained sentences, in the intrinsic evaluation Section 5.6.1. Our extrinsic analysis in Section 5.6.2 shows that this stage helps models to have significantly better recall.

In the next section, we describe the dataset generated using the HAnDS framework along with its evaluations.

5.6 Dataset Evaluation

Using the HAnDS framework, we generated two datasets as described below:

WikiFbF: A dataset generated using Wikipedia, Freebase, and the FIGER hierarchy as

[4] 150 most frequent words were used in the list.

[5] POS tags such as DT, IN, PRP, CC, WDT etc. that are least likely to be candidate for entity mention.

Data sets	Wiki-FbT	Wiki-FbF	1k-WFB-g	FIGER
# of sentences	32,583,731	31,896,989	982	434
# of entity mentions	45,696,943	37,734,658	2,420	563
# of unique entities	2,557,122	2,506,518	—	—
# of unique mentions	3,427,161	3,264,876	2,151	331
# of tokens	707,347,974	690,086,692	25,658	10,008
# of unique tokens	2,280,446	2,250,565	7,245	2,578
μ sentence length	21.71	21.63	26.13	23.06
μ label per entity	9.60	2.12	1.64	1.38
# of types	1115	118	117	43

Table 5.2: Statistics of the different datasets generated or used in this work.

an input for the HAnDS framework. This dataset contains around 38 million annotated entity mentions with 118 different types.

WikiFbT: A dataset generated using Wikipedia, Freebase, and the TypeNet hierarchy as an input for the HAnDS framework. This dataset contains around 46 million annotated entity mentions with 1115 different types.

In our experiments, we use the September 2016 Wikipedia dump. Table 5.2 lists various statistics of these datasets. In the next subsections, we estimate the quality of the generated datasets, both intrinsically and extrinsically. Our intrinsic evaluation is focused on quantitative analysis, and the extrinsic evaluation is used as a proxy to estimate precision and recall of annotations.

5.6.1 Intrinsic evaluation

In this section, we aim to analyze the quality of the HAnDS generated dataset intrinsically. We perform two kinds of analysis. First, we compare the annotations of the HAnDS generated datasets with the NDS generated datasets. Second, we compare the data characteristics of the discarded and retained sentence of the HAnDS framework.

Comparision of the annotations generated by the HAnDS framework with the NDS approach:

We analyzed these datasets quantitatively, and the result of this analysis is presented in Table 5.3. We can observe that on the same sentences, the HAnDS framework is able to generate about 1.9 times more entity mention annotations and about 1.6 times more entities for the WikiFbT corpus compared with the NDS approach. Similarly, there are around 1.8 times more entity mentions and about 1.6 times more entities in the WikiFbF corpus. In

	TypeNet hierarchy	FIGER hierarchy
$\lvert\mathcal{H}_m\rvert$	45,696,943	37,734,658
$\lvert\mathcal{N}_m\rvert$	24,594,804	20,590,776
$\lvert\mathcal{H}_m - \mathcal{N}_m\rvert$	22,585,152	18,261,738
$\lvert\mathcal{H}_m \cap \mathcal{N}_m\rvert$	23,111,791	19,472,920
$\lvert\mathcal{N}_m - \mathcal{H}_m\rvert$	1,483,013	1,117,856

(a) Analysis of entity mentions.

	TypeNet hierarchy	FIGER hierarchy
$\lvert\mathcal{H}_e\rvert$	2,557,122	2,506,518
$\lvert\mathcal{N}_e\rvert$	1,630,078	1,585,518
$\lvert\mathcal{H}_e - \mathcal{N}_e\rvert$	959,694	952,638
$\lvert\mathcal{H}_e \cap \mathcal{N}_e\rvert$	1,597,428	1,553,880
$\lvert\mathcal{N}_e - \mathcal{H}_e\rvert$	32,650	31,638

(b) Analysis of entities.

Table 5.3: Quantitative analysis of dataset generated using the HAnDS framework with the NDS approach of dataset generation. Here $\mathcal{H}_m$ and $\mathcal{H}_e$ denotes a set of entity mentions and set of entities, respectively, generated by the HAnDS framework, and $\mathcal{N}_m$ and $\mathcal{N}_e$ denotes a set of entity mentions and set of entities, respectively, generated by the NDS approach.

Section 5.6.2, we will observe that despite around 1.6 to 1.9 times more new annotations, these annotations have a very high linking precision. Also, there is a large overlap among annotations generated using the HAnDS framework and the NDS approach. Around above 95% of entity mention (and entity) annotations generated using the NDS approach are present in the HAnDS framework induced corpora. This high overlap indicates that the existing links present in Wikipedia are of high quality. The HAnDS framework removed the remaining 5% links as false positive entity mentions.

Comparison of the data characteristics of the sentences retained and discarded by the HAnDS framework:

We analyzed the discarded and retained sentences from the HAnDS framework on the following parameters:

1. **Lengths of the discarded sentences:** Are the discarded sentences longer on average?

2. **Lengths of entity mention:** Are the entity mentions in the discarded sentences longer on average?

3. **Distribution of token and entity mention:** Is there is a fundamental change in the token and entity mention distribution of discarded and retained sentences?

This analysis is done while generating WikiFbF dataset. The number of sentences in the retained corpus is 31.92 million, whereas the number of sentences in the discarded corpus is 50.33 million.

Sentence length distribution analysis: Figure 5.3 illustrates the sentence length distribution among the discarded and retained sentences. We observe that errors in sentence

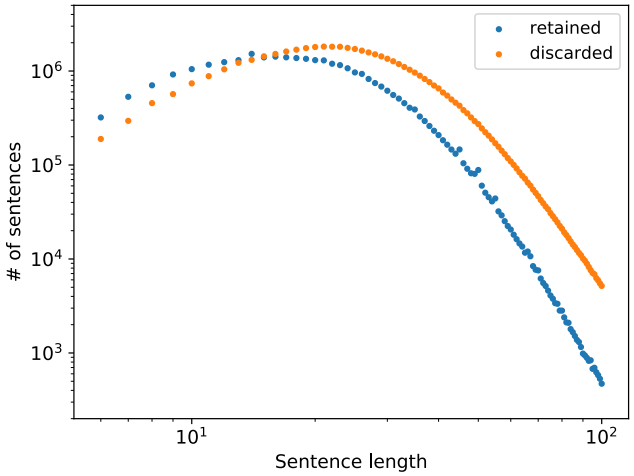


Figure 5.3: Distribution of retained and discarded sentences of length between 6 and 100 on a log-log plot.

segmentation mostly caused sentences shorter than six tokens and greater than 100 tokens. Thus we have plotted the distribution for the sentences in between 6 and 100 tokens.

In Figure 5.3, we can observe that the discarded sentences are longer. The mean length for discarded sentences is 27.29, whereas the mean length for retained sentences is 21.63.

Entity length distribution analysis: Figure 5.4 illustrates the entity length distribution among the discarded and retained sentences. We can observe that there is no notable difference between these two plots. In both these corpus, there are about 10k entity mentions with length ten tokens.

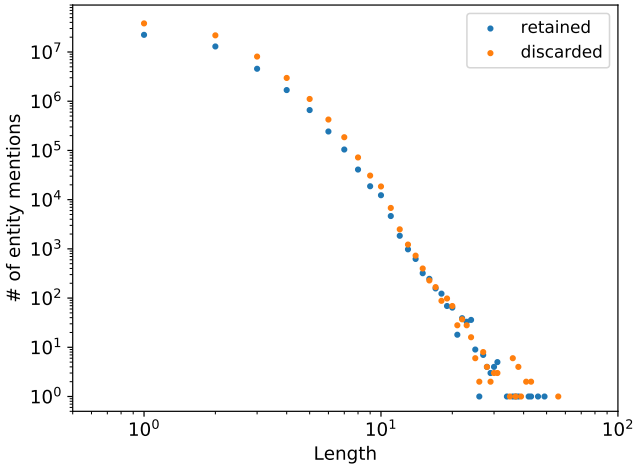
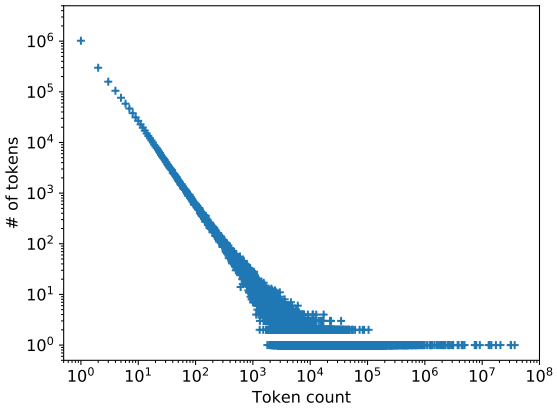
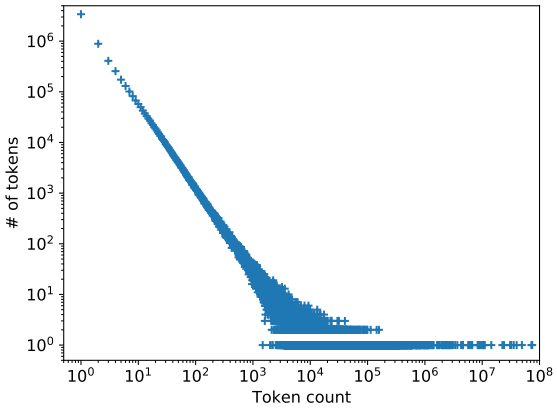


Figure 5.4: The analysis of entity length in the retained and discarded sentences on log-log scale.

Token distribution analysis: Figure 5.5 illustrates the token distribution among the

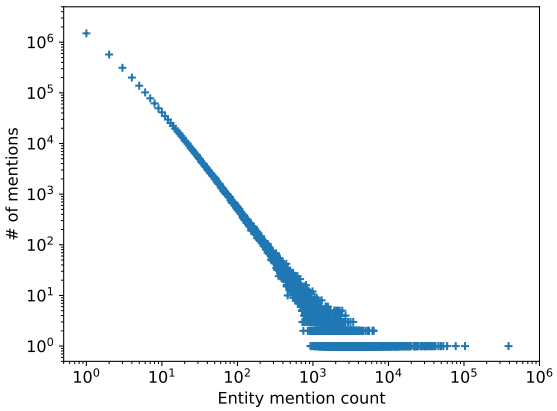


(a) Token distribution in the retained sentences.

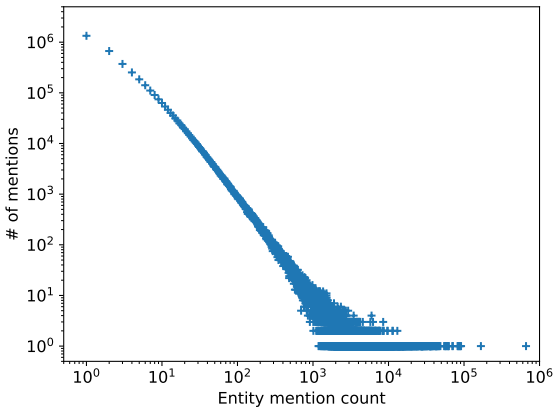


(b) Token distribution in the discarded sentences.

Figure 5.5: Token distribution analysis on log-log scale.



(a) Entity mention distribution in the retained sentences.



(b) Entity mention distribution in the discarded sentences.

Figure 5.6: Entity mention distribution analysis on log-log scale.

discarded and retained sentences. We can observe that other than a slight change in slope and absolute magnitude, there is no notable difference between these two plots. This can be attributed to the fact that the retained corpus has 31.92 million sentences, and the discarded corpus has 50.33 million sentences.

Entity mention distribution analysis: Figure 5.6 illustrates the entity mention distribution among the discarded and retained sentences. We can observe that there is no notable difference between these two plots.

Sentence length distribution comparison across multiple datasets: In Figure 5.7 and Figure 5.8, we compare the sentence length distribution of the retained and discarded sentences with five NER datasets, namely CoNLL [32], OntoNotes [21], BBN [22], CADEC [73], and BC5CDR [81]. These datasets have different writing styles, as mentioned

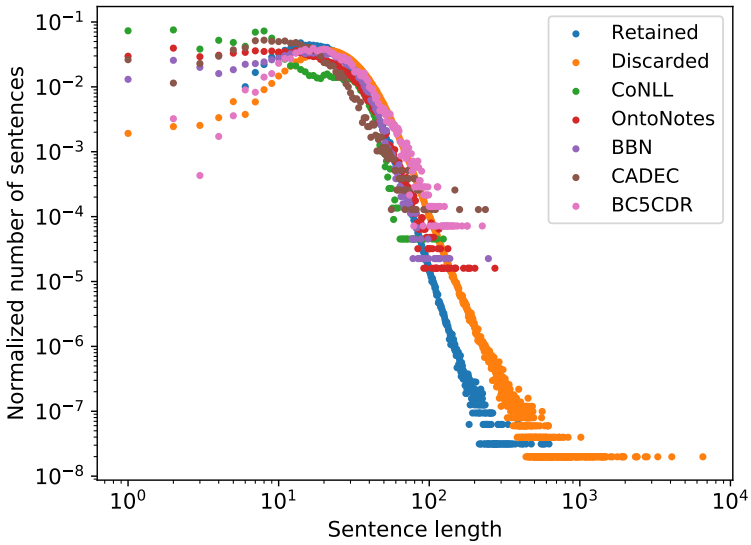


Figure 5.7: Distribution of sentence length compared across five NER datasets with the retained and discarded sentences.

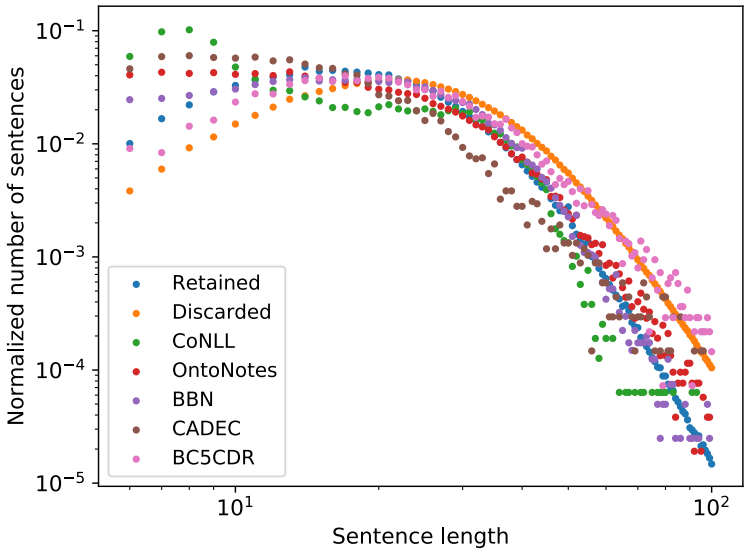


Figure 5.8: Distribution of sentence length between 6 to 100 compared across five NER datasets with the retained and discarded sentences.

below:

1. **CoNLL:** The dataset contains sentences sampled from news articles of Reuters news corpus.

2. **OntoNotes:** The dataset contains sentences sampled from news, conversation text, broadcast conversation, and weblogs.

3. **BBN:** The dataset contains sentences sampled from news article of the Wall Street

Journal.

4. **CADEC:** The dataset contains sentences sampled from a medical forum discussion related to adverse drug reactions.

5. **BC5CDR:** The dataset contains sentences sampled from clinical abstracts.

In Figure 5.7 there is no restriction on sentence length. We can observe that the most notable distribution change occurs at either shorter sentences or longer sentences.

In Figure 5.8 we only consider sentence whose length are in between 6 and 100. Here we can observe that the distribution of sentence length in five NER datasets are close to the retained sentence length distribution.

Analysis summary: Our analysis indicates that there is no significant difference in the data distribution of the retained and discarded sentence, other than the sentence length distributions. Note that this result has a subtle interpretation. We observe that in the discarded sentences, there are more than 100k sentences with a length greater than 100 tokens. A corpus constituting of only these longer sentences is larger than several news-domain NER datasets. We observe that the majority of these sentences are caused due to incorrect sentence segmentation or they follow a list like patterns such as:

1. PER, PER, PER, PER, PER, ...
2. PER - PER - PER - PER - PER - ...
3. Director (Movie), Director (Movie), Director (Movie), ...
4. Project (year), project (year), project (year), ...
5. NUMBER NUMBER NUMBER NUMBER ...
6. Movie (year), movie (year), movie (year), ...
7. PER | PER | PER | PER | PER | ...

The longest sentence length in discarded sentences is 6564 tokens. Our dataset also captures these long sentences, but the number is far less: 7664 sentences with a length greater than 100. The longest sentence length in the retained sentences is 624 tokens.

Although being a basic evaluation, the analysis conveys that these longer sentences might not be suitable for applications where NER systems are used. To support this claim, we plotted the sentence length distribution of five NER datasets from different domains in Figures 5.7 and 5.8. The result conveys that sentences longer than 100 words rarely occur

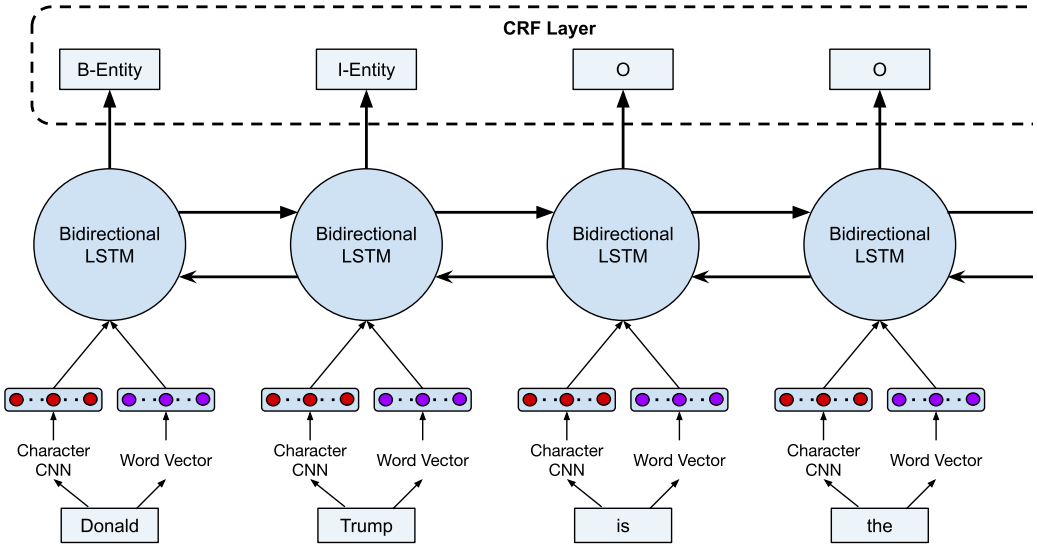


Figure 5.9: The figure illustrates the LSTM-CNN-CRF model used for the Fine-ED task.

in these domains, and the sentence length distribution in the retained sentences is closer to the sentence length distribution in these domains when compared with the discarded sentences.

5.6.2 Extrinsic evaluation

In extrinsic evaluation, we evaluate the performance of learning models when trained on datasets generated using the HAnDS framework. Due to resource constraints, we perform this evaluation only for the WikiFbF dataset and its variants.

Learning Models

Following Ling and Weld [17], we divided the Fine-ER task into two subtasks: Fine-ED, a sequence labeling problem, and Fine-ET, a multi-label classification problem. We use the existing state-of-the-art models for the respective sub-tasks. The Fine-ER model is a simple pipeline combination of a Fine-ED model followed by a Fine-ET model.

Fine-ED model: For the Fine-ED task, we use a state-of-the-art sequence labeling based LSTM-CNN-CRF model proposed by Ma and Hovy [99]. In Figure 5.9, we illustrate the LSTM-CNN-CRF model architecture. In this model, a Convolutional Neural Network (CNN) [102] is used to compute each word's representation using the word's characters. The word representation obtained using CNN is concatenated with the word representation obtained through a pre-trained word embedding models such as GloVe [72]. The concate-

nated feature representation is the input to a bi-directional LSTM network. The LSTM layer's output is the input to a Conditional Random Field (CRF) layer, which makes the final prediction.

Fine-ET model: For the Fine-ET task, we use our noise-aware LSTM based model, as described earlier in Chapter 3 [30].

Hyperparameter setting: All the deep neural network models mentioned in this chapter used 300-dimensional pre-trained word embeddings distributed by Pennington et al. [72]. The hidden layer size of word-level bi-directional LSTM was 100 for the Fine-ED model and 200 for the Fine-ET model. The hidden layer size of character-level used in the Fine-ET model was 200. A total of 30 filters of size three, were used in character CNN for the Fine-ED model. We randomly initialized character embeddings of size 50 and 200 for the Fine-ED and Fine-ET model, respectively. The character embeddings were updated during the model training. We use dropout with the probability of 0.5 on the output of LSTM encoders. We use Adam [70] as an optimization method with a learning rate of 0.001 to 0.002 and a mini-batch size of 500. The models were implemented using the TensorFlow[6] framework.

Datasets

The two learning models are trained on the following datasets:

(1) **Wiki-FbF:** Dataset created by the HAnDS framework.

(2) **Wiki-FbF-w/o-III:** Dataset created by the HAnDS framework without using stage III of the pipeline.

(3) **Wiki-NDS:** Dataset created using the NDS approach with the same Wikipedia version used in our work.

(4) **FIGER:** Dataset created using the NDS approach by Ling and Weld [17].

Except for the FIGER dataset, for other datasets, we randomly sampled two million sentences for model training due to computational constraints. However, during model training, we ensured that every model irrespective of the dataset is trained for approximately the same number of examples. Thus, each model is trained on the same number of examples and reduces the data size bias. All extrinsic evaluation experiments, subsequently reported in this section, are performed on these randomly sampled datasets. Also, the same dataset is used to train Fine-ED and Fine-ET learning model. This setting is different from Ling and Weld [17], where an entity detection model is trained on the CoNLL dataset. Hence,

[6] https://www.tensorflow.org/

Models	FIGER			1k-WFB-g		
	Precision	Recall	F1	Precision	Recall	F1
LSTM-CNN-CRF (FIGER)	87.17	28.95	43.47	91.41	37.13	52.81
CoreNLP	83.82	80.99	82.38	75.46	64.12	69.33
NER Tagger	80.44	84.01	82.19	77.25	68.52	72.62
LSTM-CNN-CRF (Wiki-NDS)	86.14	30.91	45.49	**92.80**	47.09	62.48
LSTM-CNN-CRF (Wiki-FbF-w/o-III)	**88.07**	44.58	59.2	92.55	65.03	76.39
LSTM-CNN-CRF (Wiki-FbF)	79.80	**86.32**	**82.94**	89.89	**81.98**	**85.75**

Table 5.4: The performance comparison of various entity detection models on the FIGER and 1k-WFB-g datasets.

the result reported in their work is not directly comparable.

We evaluated the learning models on the following two datasets:

(1) **FIGER:** This is a manually annotated evaluation corpus which has been created by Ling and Weld [17]. The corpus contains 563 entity mentions and overall 43 different entity types. The type distribution in this corpus is skewed as only 11 entity types are mentioned more than ten times.

(2) **1k-WFB-g:** This is a new manually annotated evaluation corpus developed specifically to cover a large type set. The corpus contains 2420 entity mentions and 117 different entity types. In this corpus, 84 entity types are mentioned more than ten times. The sentences for this dataset construction were sampled from Wikipedia text.

The statistics of these datasets are available in Table 5.2.

Evaluation Metric

For the Fine-ED task, we evaluated model performance using the *precision*, *recall*, and *F1* metrics as described in Section 2.3.1. For the Fine-ET and the Fine-ER task, we use the *strict* (or *subset accuracy*), *loose macro* and *loose micro* evaluation metrics described in Sections 2.3.2 and 2.3.3.

Result analysis for the Fine-ED task

The results of the entity detection models on the two evaluation datasets are presented in Table 5.4. We perform the following two analyses of these results. First, the effect of training datasets on models performance and second, the performance comparison among the two manually annotated datasets.

In the first analysis, we observe that the LSTM-CNN-CRF model, when trained on the Wiki-FbF dataset, has the highest F1 score on both the evaluation corpus. Moreover, the

average difference in precision and recall for this model is the lowest, which indicates a balanced performance across both evaluation corpus. When compared with the models trained on the NDS generated datasets (Wiki-NDS and FIGER), we observe that these models have the best precision across both corpus, however, lowest recall. The result indicates that a large number of false negatives entity mentions are present in the NDS induced datasets. In the case of the model trained on the Wiki-FbF-w/o-III dataset, the performance is in between the performance of a model trained on Wiki-NDS and Wiki-FbF datasets. However, they have significantly lower recall, on average, around 28% lower than the model trained on Wiki-FbF. This highlights the role of stage-III, by selecting only quality annotated sentence, erroneous annotations are removed, resulting in learning models trained on WikiFbF to have a better and balanced performance.

In the second analysis, we observe that models trained on datasets generated using Wikipedia as sentence source performs better on the 1k-WFB-g evaluation corpus as compared to the FIGER evaluation corpus. These datasets are FIGER training corpus, WikiFbF, Wiki-NDS, and Wiki-FbF-w/o-III. The primary reason for better performance is that the sentences constituting the 1k-WFB-g dataset were sampled from Wikipedia.[7] Thus, this evaluation is the same domain evaluation. On the other hand, the FIGER evaluation corpus is based on sentences sampled from news and specialized magazines (photography and veterinary domains). It has been observed in the literature that in a cross domain evaluation setting, learning model performance is reduced compared to the same domain evaluation [94]. Moreover, this result also conveys that, to some extent, learning models trained on the large Wikipedia text corpus is also able to generalize on evaluation dataset consisting of sentences from news and specialized magazines.

Our analysis in this section, as well as in Section 5.4.1, indicates that although the type coverage of FIGER evaluation corpus is low (43 types), it helps to measure the model's generalizability in a cross-domain evaluation better. Whereas, 1k-WFB-g helps to measure performance across a large spectrum of entity types (117 types). Learning models trained on Wiki-FbF perform best on both of the evaluation corpora. This warrants the usability of the generated corpus as well as the framework used to generate the corpus.

Result analysis for the Fine-ET and the Fine-ER task

We observe that for the Fine-ET task, there is not a significant difference between the performance of learning models trained on the Wiki-NDS dataset and models trained on

[7] We ensured that the test sentences are not present in any of the training datasets.

Training Datasets	FIGER			1k-WFB-g		
	S-Acc	L-Ma-F1	L-Mi-F1	S-Acc	L-Ma-F1	L-Mi-F1
FIGER	25.07	34.56	36.47	27.76	35.14	37.31
Wiki-NDS	30.07	37.89	38.55	39.12	49.28	51.60
Wiki-FbF	**56.31**	**70.70**	**68.23**	**53.34**	**68.42**	**69.23**

Table 5.5: Performance comparison for the Fine-ER task.

the Wiki-FbF dataset. The later model performs approx 1% better in the micro-F1 metric computed on the 1k-WFB-g corpus. This indicates that in the HAnDS framework stage-II, where false negative entity mentions were reduced by relinking them to Freebase, has a very high linking precision similar to NDS, which is estimated to be about $97 - 98\%$ [103].

The results for the complete Fine-ER system, i.e., Fine-ED followed by Fine-ET, are available in Table 5.5. These results support our claim in Section 5.4.1 that the current bottleneck for the Fine-ER task is Fine-ED, specifically lack of resource with quality entity boundary annotations while covering a large spectrum of entity types. Our work directly addressed this issue. In the Fine-ER task performance measure, learning model trained on WikiFbF has an average absolute performance improvement of at least 18% on all of the evaluation metrics.

Level-wise result analysis for the Fine-ET task

	Support (Train)	Support (Test)	L-Mi-F1
Level 1	55.4%	62.9%	0.838
Level 2	44.6%	37.1%	0.699

Table 5.6: The loose-micro-F1 scores of the Fine-ET model at different hierarchy levels for the Wiki-FbF (1k-WFB-g) datasets. Also, the percentage support of corresponding training and testing instances is mentioned.

From the results in Table 5.6, we can observe that the difference in support and the difference in the fine-grained label performance is less as compared to the level-wise results presented in Chapter 3. The results also indicate the quality of the HAnDS generated dataset and the use of 1k-WFB-g for the evaluation of the Fine-ET task.

5.7 Conclusion and Discussion

In this work, we initiate a push towards moving from Coarse-ER systems to Fine-ER systems, i.e., from recognizing entities from a handful of types to thousands of types. We

propose the HAnDS framework to automatically construct a quality training dataset for different variants of Fine-ER tasks. The two datasets constructed in our work, along with the evaluation resource, are currently the largest available training and testing datasets for the entity recognition problem. They are backed with empirical experimentation to warrant the quality of the constructed corpora.

The datasets generated in our work open up two new research directions related to the entity recognition problem. The first direction is towards an exploration of sequence labeling approaches in the setting of Fine-ER, where each entity mention can have more than one type. The existing state-of-the-art sequence labeling models for the Coarse-ER task, can not be directly applied in the Fine-ER setting due to state space explosion in the multi-label setting. The second direction is towards noise-robust sequence labeling models, where some of the entity boundaries are incorrect. For example, in our induced datasets, there are still entity detection errors, which are inevitable in any heuristic approach. There has been some work explored in [26] assuming that it is a priori known which tokens have noise. This information is not available in our generated datasets.

Additionally, the generated datasets are much richer in entity types compared to any existing entity recognition datasets. For example, the generated dataset contains entities from several domains such as biomedical, finance, sports, products, and entertainment. In several downstream applications where NER is used on a text writing style different from Wikipedia, the generated dataset is a good candidate as a source dataset for transfer learning to improve domain-specific performance.

6

Conclusion and Future Directions

In this thesis, we proposed learning models and dataset creation methods for the Fine-ED and Fine-ET tasks. One of the major challenges for these tasks is data-scarcity, and our contributions address this challenge either directly or indirectly.

In Chapter 3, we proposed a noise-aware deep neural network model that can learn well in the presence of positive label noise for the Fine-ET task. The proposed model significantly outperformed existing models that assumed that the training dataset is noise-free. Our analysis concludes that the noise-aware models are essential for the Fine-ET task as the training datasets are generated automatically using the distant supervision paradigm. We also proposed transfer learning approaches to improve performance on datasets with limited annotations.

In Chapter 4, we proposed a collective learning framework to use diverse, partially overlapping datasets together for the task of Fine-ET. The proposed framework efficiently utilizes datasets with partial label overlap and predicts fine-grained labels even if some of the in-domain datasets do not have those labels annotated. Our analysis conveys that the framework does not rely on any one of the datasets as a source or target. Instead, it permits a multi-directional knowledge flow where every dataset is a source and target. The framework eliminates or reduces the need for creating or reannotating datasets as it can use existing datasets that can have a subset of labels annotated.

In Chapter 5, we proposed the HAnDS framework to construct better datasets for the fine-ET and fine-ED tasks automatically. Our analysis conveys that the constructed datasets are of good quality and provide a significant improvement to the learning models for the Fine-ED and Fine-ET task.

6.1 Limitations of the Proposed Work

While the proposed work achieves state-of-the-art results on the respective tasks, there are several fundamental limitations of the proposed work as discussed:

Positive only label noise: The proposed work in Chapter 3, only address label noise where only a subset of labels among annotated positive labels is correct. However, since the datasets are generated using the distant supervision paradigm, there are also instances where the actual correct label might not be annotated.

Domain expert involvement in the hierarchy creation process: The proposed work in Chapter 4 is dependent on a domain expert to compare two labels during the hierarchy creation process (Section 4.4.1). We observe that some of the comparisons are straightforward, while others require a thorough analysis of the annotation guidelines and referring to external sources.

Entity boundary noise: The proposed work in Chapter 5 assumed that in the training dataset, the boundaries of entities are entirely accurate. However, since the dataset is generated using the distant supervision paradigm, this assumption is not valid.

6.2 Future Work Directions

While the dissertation has made significant progress in the advancement of the Fine-ED and Fine-ET tasks, there are still several open problems that remain unaddressed. Many of which are worth pursuing as future work, as discussed:

Better modeling of the label noise for the Fine-ET task: In the training datasets for the Fine-ET task, a significant portion of entity mentions are annotated with incorrect labels either as false positives or false negatives. While the proposed work, as well as works published afterward [104, 105], focuses only on the false-positive label noise, the false-negative noise problem remains open.

Modeling of entity boundary noise for the Fine-ED task: The existing work related to sequence labeling in the presence of noise assumes that the tokens where the noise is present are already known [26]. However, in the setting of Fine-ED, the tokens that have noise is not known beforehand. Building sequence labeling models in the presence of boundary noise is research direction worth pursuing.

Domain generalization and adaptation in collective learning: In our proposed work in Chapter 4, there are multiple datasets, each with a different text source/domain. In our evaluations, we assumed that the testing domain is known in the way that it can be any one of the training domain. However, there exist scenarios where the testing domain is entirely unknown, or no labeled training dataset is available in the testing domain. In such cases, domain generalization [106] and unsupervised domain adaptation [107] techniques are used. However, the existing work focuses on either one source, and the target domain or the source domains have the same labels. Domain generalization and adaption in the setting of partially overlapping labels is an open research problem.

Publications

Manuscripts Published

[1] Abhishek Abhishek, Ashish Anand, and Amit Awekar. "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*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 797–807. URL: http://www.aclweb.org/anthology/E17-1075.

[2] Abhishek Abhishek. "FgER: Fine-Grained Entity Recognition". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 8008–8010. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16718.

[3] Abhishek Abhishek, Sanya Taneja, Garima Malik, Ashish Anand, and Amit Awekar. "Fine-grained Entity Recognition with Reduced False Negatives and Large Type Coverage". In: *Automated Knowledge Base Construction*. Amherst, USA, May 2019. URL: https://openreview.net/forum?id=HylHE-9p6m.

[4] Abhishek Abhishek, Amar Prakash Azad, Balaji Ganesan, Ashish Anand, and Amit Awekar. "Collective Learning From Diverse Datasets for Entity Typing in the Wild". In: *2nd International Workshop on EntitY REtrieval (EYRE'19), CIKM 2019.* (2019). URL: http://ceur-ws.org/Vol-2446/paper3.pdf.

Bibliography

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge". In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD '08. Vancouver, Canada: ACM, 2008, pp. 1247–1250. ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376746. URL: http://doi.acm.org/10.1145/1376616.1376746.

[2] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. "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. New York, New York, USA: ACM, 2014, pp. 601–610. ISBN: 978-1-4503-2956-9. DOI: 10.1145/2623330.2623623. URL: http://doi.acm.org/10.1145/2623330.2623623.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. "DBpedia: A Nucleus for a Web of Open Data". In: *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*. ISWC'07/ASWC'07. Busan, Korea: Springer-Verlag, 2007, pp. 722–735. ISBN: 3-540-76297-3, 978-3-540-76297-3. URL: http://dl.acm.org/citation.cfm?id=1785162.1785216.

[4] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. "Yago: A Core of Semantic Knowledge". In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: ACM, 2007, pp. 697–706. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242667. URL: http://doi.acm.org/10.1145/1242572.1242667.

[5] Denny Vrandečić and Markus Krötzsch. "Wikidata: A Free Collaborative Knowledgebase". In: *Commun. ACM* 57.10 (Sept. 2014), pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489.

[6] T. Mitchell et al. “Never-ending Learning”. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI’15. Austin, Texas: AAAI Press, 2015, pp. 2302–2310. ISBN: 0-262-51129-0. URL: http://dl.acm.org/citation.cfm?id=2886521.2886641.

[7] Thomas Lin, Mausam, and Oren Etzioni. “No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities”. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 893–903. URL: http://www.aclweb.org/anthology/D12-1082.

[8] W. Shen, J. Wang, and J. Han. “Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (Feb. 2015), pp. 443–460. ISSN: 1041-4347. DOI: 10.1109/TKDE.2014.2327028.

[9] Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. “AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label Embedding”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1369–1378. DOI: 10.18653/v1/D16-1144. URL: https://aclweb.org/anthology/D16-1144.

[10] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. “Relation Classification via Convolutional Deep Neural Network”. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2335–2344. URL: https://www.aclweb.org/anthology/C14-1220.

[11] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. “Open Information Extraction from the Web”. In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. IJCAI’07. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, pp. 2670–2676. URL: http://dl.acm.org/citation.cfm?id=1625275.1625705.

[12] David Nadeau and Satoshi Sekine. “A survey of named entity recognition and classification”. In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[13] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. "Named Entity Recognition: Fallacies, challenges and opportunities". In: *Computer Standards & Interfaces* 35.5 (2013), pp. 482–489. ISSN: 0920-5489. DOI: http://dx.doi.org/10.1016/j.csi.2012.09.004. URL: http://www.sciencedirect.com/science/article/pii/S0920548912001080.

[14] Vikas Yadav and Steven Bethard. "A Survey on Recent Advances in Named Entity Recognition from Deep Learning models". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2145–2158. URL: https://www.aclweb.org/anthology/C18-1182.

[15] Mark Craven and Johan Kumlien. "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, 1999, pp. 77–86. ISBN: 1-57735-083-9. URL: http://dl.acm.org/citation.cfm?id=645634.663209.

[16] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. "Distant supervision for relation extraction without labeled data". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 1003–1011. URL: http://www.aclweb.org/anthology/P/P09/P09-1113.

[17] Xiao Ling and Daniel S. Weld. "Fine-grained Entity Recognition". In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI'12. Toronto, Ontario, Canada: AAAI Press, 2012, pp. 94–100. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5152.

[18] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. "HYENA: Hierarchical Type Classification for Entity Names". In: *Proceedings of COLING 2012: Posters*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 1361–1370. URL: http://www.aclweb.org/anthology/C12-2133.

[19] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. "Improving Efficiency and Accuracy in Multilingual Entity Extraction". In: *Proceedings of the 9th International Conference on Semantic Systems*. I-SEMANTICS '13. Graz, Austria:

ACM, 2013, pp. 121–124. ISBN: 978-1-4503-1972-0. DOI: 10.1145/2506182.2506198. URL: http://doi.acm.org/10.1145/2506182.2506198.

[20] Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. "Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding". In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 1825–1834. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939822. URL: http://doi.acm.org/10.1145/2939672.2939822.

[21] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. "Ontonotes release 5.0 LDC2013T19". In: *Linguistic Data Consortium, Philadelphia, PA* (2013).

[22] Ralph Weischedel and Ada Brunstein. "BBN Pronoun Coreference and Entity Type Corpus LDC2005T33". In: *Linguistic Data Consortium, Philadelphia* 112 (2005).

[23] Abhishek Abhishek, Sanya Taneja, Garima Malik, Ashish Anand, and Amit Awekar. "Fine-grained Entity Recognition with Reduced False Negatives and Large Type Coverage". In: *Automated Knowledge Base Construction*. Amherst, USA, May 2019. URL: https://openreview.net/forum?id=HylHE-9p6m.

[24] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. "Learning with Noisy Labels". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 1196–1204. URL: http://papers.nips.cc/paper/5073-learning-with-noisy-labels.pdf.

[25] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. "Learning from massive noisy labeled data for image classification". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 2691–2699. DOI: 10.1109/CVPR.2015.7298885.

[26] Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. "Sequence Learning from Data with Multiple Labels". In: *ECML/PKDD Workshop on Learning from Multi-Label Data (MLD)* (2009), pp. 39–48. URL: http://lpis.csd.auth.gr/workshops/mld09/mld09.pdf.

[27] Sinno Jialin Pan, Qiang Yang, et al. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

[28] Yaqing Wang and Quanming Yao. "Few-shot learning: A survey". In: *arXiv preprint arXiv:1904.05046* (2019). URL: https://arxiv.org/abs/1904.05046.

[29] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. "Zero-Shot Learning Through Cross-Modal Transfer". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 935–943. URL: http://papers.nips.cc/paper/5027-zero-shot-learning-through-cross-modal-transfer.pdf.

[30] Abhishek Abhishek, Ashish Anand, and Amit Awekar. "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*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 797–807. URL: https://www.aclweb.org/anthology/E17-1075.

[31] Abhishek Abhishek, Amar Prakash Azad, Balaji Ganesan, Ashish Anand, and Amit Awekar. "Collective Learning From Diverse Datasets for Entity Typing in the Wild". In: *2nd International Workshop on EntitY REtrieval (EYRE'19), CIKM 2019.* (2019). URL: http://ceur-ws.org/Vol-2446/paper3.pdf.

[32] Erik F. Tjong Kim Sang and Fien De Meulder. "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*. CONLL '03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 142–147. DOI: 10.3115/1119176.1119195. URL: http://www.aclweb.org/anthology/W03-0419.pdf.

[33] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. "NCBI disease corpus: a resource for disease name recognition and concept normalization". In: *Journal of biomedical informatics* 47 (2014), pp. 1–10. DOI: https://doi.org/10.1016/j.jbi.2013.12.006.

[34] Saul A Kripke. "Naming and necessity". In: *Semantics of natural language*. Springer, 1972, pp. 253–355.

[35] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. "RCV1: A New Benchmark Collection for Text Categorization Research". In: *J. Mach. Learn. Res.* 5 (Dec. 2004),

pp. 361–397. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1005332.1005345.

[36] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. "Robust Disambiguation of Named Entities in Text". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, July 2011, pp. 782–792. URL: https://www.aclweb.org/anthology/D11-1072.

[37] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. "Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition". In: *Proceedings of the 3rd Workshop on Noisy User-generated Text*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 140–147. DOI: 10.18653/v1/W17-4418. URL: https://www.aclweb.org/anthology/W17-4418.

[38] Brett Kessler, Geoffrey Nunberg, and Hinrich Schutze. "Automatic Detection of Text Genre". In: *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*. Madrid, Spain: Association for Computational Linguistics, July 1997, pp. 32–38. DOI: 10.3115/976909.979622. URL: https://www.aclweb.org/anthology/P97-1005.

[39] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. "A Review of Relational Machine Learning for Knowledge Graphs". In: *Proceedings of the IEEE* 104.1 (Jan. 2016), pp. 11–33. DOI: 10.1109/JPROC.2015.2483592.

[40] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. "Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago". In: *Semantic Web* 9.1 (2018), pp. 77–129.

[41] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. "End-to-End Neural Entity Linking". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 519–529. DOI: 10.18653/v1/K18-1050. URL: https://www.aclweb.org/anthology/K18-1050.

[42] Özge Sevgili, Alexander Panchenko, and Chris Biemann. "Improving Neural Entity Disambiguation with Graph Embeddings". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*.

Florence, Italy: Association for Computational Linguistics, July 2019, pp. 315–322. DOI: 10.18653/v1/P19-2044. URL: https://www.aclweb.org/anthology/P19-2044.

[43] Nitish Gupta, Sameer Singh, and Dan Roth. "Entity Linking via Joint Encoding of Types, Descriptions, and Context". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2681–2690. DOI: 10.18653/v1/D17-1284. URL: https://www.aclweb.org/anthology/D17-1284.

[44] Lance Ramshaw and Mitch Marcus. "Text Chunking using Transformation-Based Learning". In: *Third Workshop on Very Large Corpora*. 1995. URL: https://www.aclweb.org/anthology/W95-0107.

[45] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. ISBN: 1-55860-778-1.

[46] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. "Neural Architectures for Named Entity Recognition". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 260–270. URL: http://www.aclweb.org/anthology/N16-1030.

[47] Michael Collins and Yoram Singer. "Unsupervised Models for Named Entity Classification". In: *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. 1999, pp. 100–110.

[48] Lev Ratinov and Dan Roth. "Design Challenges and Misconceptions in Named Entity Recognition". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 147–155. URL: http://www.aclweb.org/anthology/W09-1119.

[49] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Lin-*

guistics: System Demonstrations. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 55–60. URL: http://www.aclweb.org/anthology/P14-5010.

[50] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. "Type-Aware Distantly Supervised Relation Extraction with Linked Arguments". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1891–1901. URL: http://www.aclweb.org/anthology/D14-1203.

[51] Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. "A Hybrid Neural Model for Type Classification of Entity Mentions". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 1243–1249. ISBN: 978-1-57735-738-4. URL: http://dl.acm.org/citation.cfm?id=2832415.2832422.

[52] Xin Li and Dan Roth. "Learning Question Classifiers". In: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*. COLING '02. Taipei, Taiwan: Association for Computational Linguistics, 2002, pp. 1–7. DOI: 10.3115/1072228.1072378. URL: http://dx.doi.org/10.3115/1072228.1072378.

[53] Dani Yogatama, Daniel Gillick, and Nevena Lazic. "Embedding Methods for Fine Grained Entity Type Classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 291–296. URL: http://www.aclweb.org/anthology/P15-2048.

[54] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. "An Attentive Neural Architecture for Fine-grained Entity Type Classification". In: *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*. San Diego, CA: Association for Computational Linguistics, June 2016, pp. 69–74. URL: http://www.aclweb.org/anthology/W16-1313.

[55] Lorien Y. Pratt. "Discriminability-Based Transfer Between Neural Networks". In: *Advances in Neural Information Processing Systems 5*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 204–211. ISBN: 1-55860-274-7. URL: http://dl.acm.org/citation.cfm?id=645753.668046.

[56] Ralph Grishman and Beth Sundheim. "Message Understanding Conference-6: A Brief History". In: *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*. COLING '96. Copenhagen, Denmark: Association for Computational Linguistics, 1996, pp. 466–471. DOI: 10.3115/992628.992709. URL: https://doi.org/10.3115/992628.992709.

[57] Michael Fleischman and Eduard Hovy. "Fine Grained Classification of Named Entities". In: *COLING 2002: The 19th International Conference on Computational Linguistics*. 2002. URL: https://www.aclweb.org/anthology/C02-1130.

[58] Claudio Giuliano and Alfio Gliozzo. "Instance-Based Ontology Population Exploiting Named-Entity Substitution". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, Aug. 2008, pp. 265–272. URL: https://www.aclweb.org/anthology/C08-1034.

[59] Michael Fleischman. "Automated subcategorization of named entities". In: *In ACL (Companion Volumei)*. 2001, pp. 25–30. DOI: 10.1.1.486.1401.

[60] Seungwoo Lee and Gary Geunbae Lee. "Heuristic Methods for Reducing Errors of Geographic Named Entities Learned by Bootstrapping". In: *Second International Joint Conference on Natural Language Processing: Full Papers*. 2005. DOI: 10.1007/11562214_58. URL: https://www.aclweb.org/anthology/I05-1058.

[61] Satoshi Sekine and Chikashi Nobata. "Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/65.pdf.

[62] David Nadeau. "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision". PhD thesis. University of Ottawa, Nov. 2007. URL: http://cogprints.org/5859/.

[63] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. "Context-dependent fine-grained entity type tagging". In: *arXiv preprint arXiv:1412.1820* (2014).

[64] Lei Shi, Rada Mihalcea, and Mingjun Tian. "Cross Language Text Classification by Model Translation and Semi-Supervised Learning". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA:

Association for Computational Linguistics, Oct. 2010, pp. 1057–1067. URL: http://www.aclweb.org/anthology/D10-1103.

[65] Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. "Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 477–487. URL: http://www.aclweb.org/anthology/N12-1052.

[66] Lili Mou, Ran Jia, Yan Xu, Ge Li, Lu Zhang, and Zhi Jin. "Distilling Word Embeddings: An Encoding Approach". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: ACM, 2016, pp. 1977–1980. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983888. URL: http://doi.acm.org/10.1145/2983323.2983888.

[67] Dong Wang and Thomas Fang Zheng. "Transfer learning for speech and language processing". In: *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE. 2015, pp. 1225–1237. DOI: 10.1109/APSIPA.2015.7415532.

[68] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[69] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.

[70] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[71] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436. DOI: https://doi.org/10.1038/nature14539.

[72] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

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

[74] Nigel Collier and Jin-Dong Kim. "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 (NLPBA/BioNLP)*. Geneva, Switzerland: COLING, Aug. 2004, pp. 73–78.

[75] Peng Xu and Denilson Barbosa. "Neural Fine-Grained Entity Type Classification with Hierarchy-Aware Loss". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 16–25. URL: https://www.aclweb.org/anthology/N18-1002.

[76] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[77] Hugo Liu and Push Singh. "ConceptNet—a practical commonsense reasoning tool-kit". In: *BT technology journal* 22.4 (2004), pp. 211–226.

[78] Nam Nguyen and Rich Caruana. "Classification with Partial Labels". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '08. Las Vegas, Nevada, USA: ACM, 2008, pp. 551–559. ISBN: 978-1-60558-193-4. DOI: 10.1145/1401890.1401958. URL: http://doi.acm.org/10.1145/1401890.1401958.

[79] Timothee Cour, Ben Sapp, and Ben Taskar. "Learning from Partial Labels". In: *Journal of Machine Learning Research* 12 (July 2011), pp. 1501–1536. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1953048.2021049.

[80] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. "Disambiguation-free partial label learning". In: *IEEE Transactions on Knowledge and Data Engineering* 29.10 (2017), pp. 2155–2167. DOI: 10.1109/TKDE.2017.2721942.

[81] 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. "BioCreative V CDR task corpus: a resource for chemical disease relation extraction". In: *Database: The Journal of Biological Databases and Curation* 2016 (2016).

[82] Rich Caruana. “Multitask learning”. In: *Machine learning* 28.1 (1997), pp. 41–75.

[83] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. 1st. Chapman & Hall/CRC, 2012. ISBN: 1439830037, 9781439830031.

[84] J. Redmon and A. Farhadi. “YOLO9000: Better, Faster, Stronger”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690.

[85] Slav Petrov, Dipanjan Das, and Ryan McDonald. “A Universal Part-of-Speech Tagset”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. Istanbul, Turkey: European Languages Resources Association (ELRA), May 2012, pp. 2089–2096.

[86] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. “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. 2009, pp. 94–99. URL: http://dl.acm.org/citation.cfm?id=1859664.1859670.

[87] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. “Learning multilingual named entity recognition from Wikipedia”. In: *Artificial Intelligence* 194 (2013), pp. 151–175. DOI: 10.1016/j.artint.2012.03.006.

[88] Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. “Hierarchical Losses and New Resources for Fine-grained Entity Typing and Linking”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 97–109. URL: https://www.aclweb.org/anthology/P18-1010.

[89] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. “Ultra-Fine Entity Typing”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 87–96. URL: http://aclweb.org/anthology/P18-1009.

[90] William J Black, Fabio Rinaldi, and David Mowatt. “FACILE: Description of the NE System Used for MUC-7”. In: *Seventh Message Understanding Conference (MUC-7):*

Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1. 1998. URL: https://www.aclweb.org/anthology/M98-1014.

[91] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schuetze. "Corpus-Level Fine-Grained Entity Typing". In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 835–862. ISSN: 1076-9757.

[92] Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. "Improving Neural Fine-Grained Entity Typing With Knowledge Attention". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018. URL: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16321.

[93] Joel Nothman, James R. Curran, and Tara Murphy. "Transforming Wikipedia into Named Entity Training Data". In: *Proceedings of the Australasian Language Technology Association Workshop 2008*. Hobart, Australia, Dec. 2008, pp. 124–132. URL: https://www.aclweb.org/anthology/U08-1016.

[94] Joel Nothman, Tara Murphy, and James R. Curran. "Analysing Wikipedia and Gold-Standard Corpora for NER Training". In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 612–620. URL: https://www.aclweb.org/anthology/E09-1070.

[95] Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. "POLYGLOT-NER: Massive Multilingual Named Entity Recognition". In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. 2014, pp. 586–594. DOI: 10.1137/1.9781611974010.66.

[96] Abbas Ghaddar and Phillippe Langlais. "WiNER: A Wikipedia Annotated Corpus for Named Entity Recognition". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 413–422. URL: https://www.aclweb.org/anthology/I17-1042.

[97] Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. "A Survey of Noise Reduction Methods for Distant Supervision". In: *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. AKBC '13. San Francisco, California, USA: ACM, 2013, pp. 73–78. ISBN: 978-1-4503-2411-3. DOI: 10.1145/2509558.2509571.

[98] Van-Thuy Phi, Joan Santoso, Masashi Shimbo, and Yuji Matsumoto. "Ranking-Based Automatic Seed Selection and Noise Reduction for Weakly Supervised Relation Extraction". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 89–95. URL: https://www.aclweb.org/anthology/P18-2015.

[99] Xuezhe Ma and Eduard Hovy. "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)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1064–1074. URL: http://www.aclweb.org/anthology/P16-1101.

[100] Olivier Bodenreider. "The unified medical language system (UMLS): integrating biomedical terminology". In: *Nucleic acids research* 32 (2004), pp. D267–D270. DOI: 10.1093/nar/gkh061.

[101] Satoshi Sekine. "Extended Named Entity Ontology with Attribute Information". In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA), May 2008. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/21_paper.pdf.

[102] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[103] Chengyu Wang, Rong Zhang, Xiaofeng He, and Aoying Zhou. "Error Link Detection and Correction in Wikipedia". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: ACM, 2016, pp. 307–316. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983705.

[104] Junshuang Wu, Richong Zhang, Yongyi Mao, Hongyu Guo, and Jinpeng Huai. "Modeling Noisy Hierarchical Types in Fine-Grained Entity Typing: A Content-Based Weighting Approach". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5264–5270. DOI: 10.24963/ijcai.2019/731. URL: https://doi.org/10.24963/ijcai.2019/731.

[105] Ying Lin and Heng Ji. "An Attentive Fine-Grained Entity Typing Model with Latent Type Representation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6196–6201. DOI: 10.18653/v1/D19-1641. URL: https://www.aclweb.org/anthology/D19-1641.

[106] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. "Domain Generalization via Invariant Feature Representation". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, pp. I-10–I-18. URL: http://dl.acm.org/citation.cfm?id=3042817.3042820.

[107] Yaroslav Ganin and Victor Lempitsky. "Unsupervised Domain Adaptation by Backpropagation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1180–1189. URL: http://proceedings.mlr.press/v37/ganin15.html.

Brief Biography of the Author

Abhishek joined the M.Tech + Ph.D. dual degree program at the Department of Computer Science and Engineering of the Indian Institute of Technology Guwahati (IITG), India, in July 2014. In summer 2018, he did a research internship at IBM Research India. Before joining the dual degree program, the author worked as a software developer at Hi-Tech Robotic Systemz Ltd. He did his Bachelor of Engineering in Electronics and Electrical Communication from PEC University of Technology, Chandigarh, India, in May 2013. He was awarded the LDC Data scholarship 2016, Google Travel Grant 2017, ACM-IARCS travel grant 2017, and AAAI travel grant 2018. Other than conference presentations, he has also participated and given talks/tutorials at various events and institutes such as IWML (2016), NIT Raipur (2017), TBML ICTS Bangalore (2018), Amazon Research Day (2018), CoDS-COMAD (2019), Diffbot (2019) and Amazon Research Day (2019). His research interest includes Natural Language Processing, Machine Learning, and Data Mining.

Contact Information

Email	:	abhishek.abhishek@iitg.ac.in, abhishekmehta1992@gmail.com
Web	:	http://abhishek.ind.in
Address	:	1880, Sector 7-C, Chandigarh - 160019, INDIA