

Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embeddings

Abhishek, Ashish Anand and Amit Awekar
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Assam, India - 781039

{abhishek.abhishek, anand.ashish, awekar}@iitg.ernet.in

Abstract

Fine-grained entity type classification (FETC) is the task of classifying an entity mention to a broad set of types. Distant supervision paradigm is extensively used to generate training data for this task. However, generated training data assigns same set of labels to every mention of an entity without considering its local context. Existing FETC systems have two major drawbacks: assuming training data to be noise free and use of hand crafted features. Our work overcomes both drawbacks. We propose a neural network model that jointly learns entity mentions and their context representation to eliminate use of hand crafted features. Our model treats training data as noisy and uses non-parametric variant of hinge loss function. Experiments show that the proposed model outperforms previous state-of-the-art methods on two publicly available datasets, namely FIGER(GOLD) and BBN with an average relative improvement of 2.69% in micro-F1 score. Knowledge learnt by our model on one dataset can be transferred to other datasets while using same model or other FETC systems. These approaches of transferring knowledge further improve the performance of respective models.

1 Introduction

Entity type classification is the task for assigning types or labels such as *organization*, *location* to entity mentions in a document. This classification is useful for many natural language processing (NLP) tasks such as relation extraction (Mintz et al., 2009), machine translation (Koehn et al.,

2007), question answering (Lin et al., 2012) and knowledge base construction (Dong et al., 2014).

There has been considerable amount of work on Named Entity Recognition (NER) (Collins and Singer, 1999; Tjong Kim Sang and De Meulder, 2003; Ratnov and Roth, 2009; Manning et al., 2014), which classifies entity mentions into a small set of mutually exclusive types, such as *Person*, *Location*, *Organization* and *Misc*. However, these types are not enough for some NLP applications such as relation extraction, knowledge base construction (KBC) and question answering. In relation extraction and KBC, knowing fine-grained types for entities can significantly increase the performance of the relation extractor (Ling and Weld, 2012; Koch et al., 2014; Mitchell et al., 2015) since this helps in filtering out candidate relation types that do not follow the type constrain. Fine-grained entity types provide additional information while matching questions to its potential answers and significantly improves performance (Dong et al., 2015). For example, Li and Roth (2002) rank questions based on their expected answer types (will the answer be *food*, *vehicle* or *disease*).

Typically, FETC systems use over hundred labels, arranged in a hierarchical structure. An important aspect of FETC is that based on local context, two different mentions of same entity can have different labels. We illustrate this through an example in Figure 1. All three sentences *S1*, *S2*, and *S3* mention same entity *Barack Obama*. However, looking at the context, we can infer that *S1* mentions Obama as a *person/author*, *S2* mentions Obama only as a *person*, and *S3* mentions Obama as a *person/politician*.

Available training data for FETC has noisy labels. Creating manually annotated training data for FETC is time consuming, expensive, and error prone. Note that, a human annotator will

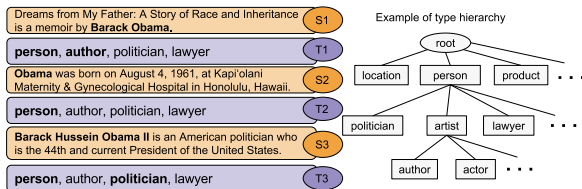


Figure 1: Noise introduced via distant supervision process. S1-S3 indicates sentences where only a subset of labels for entity mention (bold typeface) are relevant given context, highlighted in T1-T3.

have to assign a subset of correct labels from a set of around hundred labels for each entity mention in the corpus. Existing FETC systems use distant supervision paradigm (Craven and Kumlien, 1999) to automatically generate training data. Distant supervision maps each entity in the corpus to knowledge bases such as Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007). This method assigns same set of labels to all mentions of an entity across the corpus. For example, *Barack Obama* is a person, politician, lawyer, and author. If a knowledge base has these four matching labels for Barack Obama, then distant supervision assigns all of them to every mention of Barack Obama. Training data generated with distant supervision will fail to distinguish between mentions of Barack Obama in sentences *S1*, *S2*, and *S3*.

Existing FETC systems have one or both of following drawbacks:

1. Assuming training data to be noise free (Ling and Weld, 2012; Yosef et al., 2012; Yogatama et al., 2015; Shimaoka et al., 2016)
2. Use of hand crafted features (Ling and Weld, 2012; Yosef et al., 2012; Yogatama et al., 2015; Ren et al., 2016)

We have observed that for real world datasets, more than twenty five percent of training data has noisy labels. First drawback propagates this noise in training data to the FETC model. To extract hand crafted features various NLP tools are used. Since errors inevitably exist in such tools, the second drawback propagates errors of these tools to FETC model.

We propose a neural network based model to overcome the two drawbacks of existing FETC systems. First, we separate training data into *clean* and *noisy* partitions using the same method as in AFET system (Ren et al., 2016). For these parti-

tions, we use simple yet effective non-parametric variant of hinge loss function while training. To avoid use of hand crafted features, we learn representations for given entity mention and its context.

Additionally, we investigate effectiveness of using transfer learning (Pratt, 1993) for FETC task both at feature and model level. We show that feature level transfer learning can be used to improve performance of other FETC system such as AFET, by up to 4.5% in micro-F1 score. Similarly, model level transfer learning can be used to improve performance of the same model using different dataset by up to 3.8% in micro-F1 score.

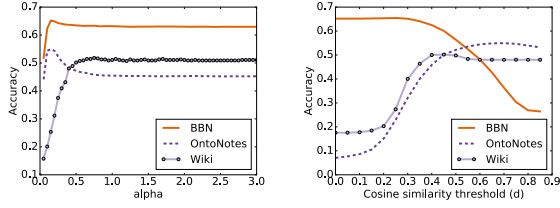
Our contributions can be summarized as follows:

1. We propose a simple neural network model that learns representations for entity mention and its context, and incorporate noisy label information using a variant of non-parametric hinge loss function. Experimental results on two publicly available datasets demonstrate the effectiveness of proposed model, with an average relative improvement of 2.69% in micro-F1 score.
2. We investigate the use of feature level and model level transfer-learning strategies in the domain of the FETC task. The proposed transfer learning strategies further improve the state-of-the-art on BBN dataset by 3.8% in micro-F1 score.

2 Related Work

Ling et al. (2012) proposed the first system for FETC task, which used 112 overlapping labels. They used linear classifier perceptron for multi-label classification. Yosef et al. (2012) 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. (2014) introduced context dependent FETC and proposed a set of heuristics for pruning labels that might not be relevant given the entity mention's local context. Yogatama et al. (2015) 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. (2016) proposed an attentive neural network model that used LSTMs to encode entity mention's context and used an atten-



(a) α models label-label correlation. Higher the α , lower above this threshold are predicted as positive correlation labels.

Figure 2: Effect of change of parameters on AFET’s performance.

tion 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 which is simpler yet more effective.

Most recently, Ren et al. (2016) have proposed AFET, an FETC system. AFET separates the loss function for *clean* and *noisy* entity mentions. AFET uses label-label correlation information obtained by given data in its parametric loss function (model parameter α). During inference, AFET uses a threshold to separate positive types from negative types (similarity threshold parameter d). However, AFET’s loss function is sensitive to change in parameters, which are data dependent. Figure 2 shows the effect of parameter α and d , on AFET performance evaluated on different datasets. In contrast, our model uses a simple yet effective variant of hinge loss function. This function does not need to tune the similarity threshold.

Transfer learning is well applied to many NLP applications, such as cross-domain document classification (Shi et al., 2010), multi-lingual word clustering (Täckström et al., 2012) and sentiment classification (Mou et al., 2016). Initialization of word vectors with pre-trained word vectors in neural network models can be considered as one of the best example of transfer learning in NLP. Wang et al. (2015) provide a broad overview of transfer learning techniques used for language processing.

3 The Proposed Model

3.1 Problem description

Our task is to automatically classify type information of entity mentions present in natural language

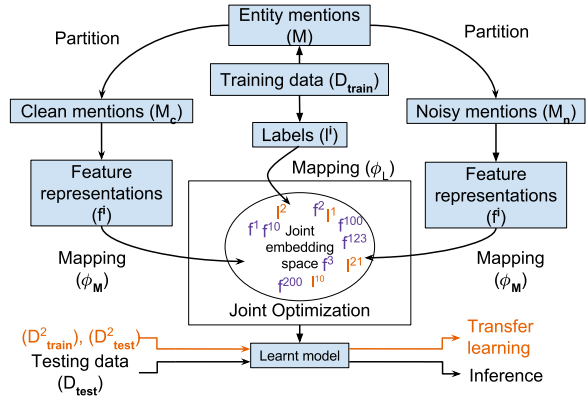


Figure 3: The system overview.

sentences. Figure 3 shows a general overview of our proposed approach.

Input: The input to the model is a training and testing corpus consisting of a set of sentences on which entity mentions have been identified. In 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 denotes 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 binary vector, where $l_{j,k}^i = 1$ if t^{th} type is true otherwise it will be zero. K denotes the total number of labels in a given hierarchy Ψ . Testing corpus \mathcal{D}_{test} will only contain sentences and entity mentions.

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

3.2 Training set partition

Similar to AFET, we partition the mention set \mathcal{M} of training corpus \mathcal{D}_{train} into two parts, a set \mathcal{M}_c , consisting only of *clean* entity mentions and a set \mathcal{M}_n , consisting only of *noisy* entity mentions. An entity mention $m_{j,k}^i$ is said to be *clean* if its labels $l_{j,k}^i$ 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 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*.

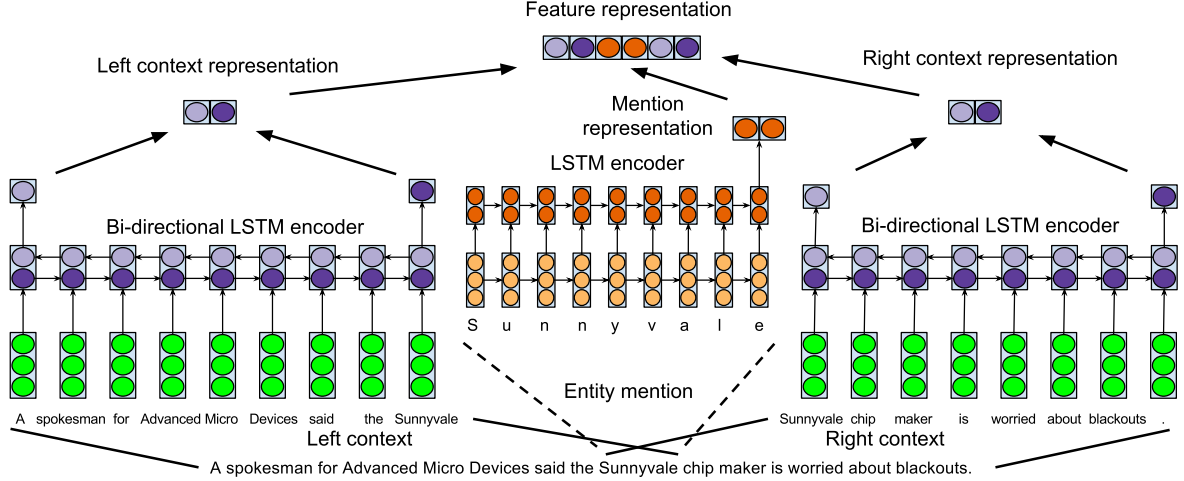


Figure 4: The architecture of the proposed model.

3.3 Feature representations

Mention representation: This representation captures information about entity mention’s morphology and orthography. We decompose an entity mention into character sequence, and use a vanilla LSTM encoder (Hochreiter and Schmidhuber, 1997) to encode character sequences to a fixed dimensional vector. Formally, for entity mention $m_{j,k}^i$, we decompose it into a sequence of character tokens $c_{j,k_1}^i, c_{j,k_2}^i, \dots, c_{j,k_{|m_{j,k}^i|}}^i$, where $|m_{j,k}^i|$ 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 corresponding vector representation in a lookup table for characters. The character sequence is then fed one by one to a LSTM encoder, and the final output is used as a feature representation for entity mention $m_{j,k}^i$. 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 4 (Mention representation).

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

of tokens from the start of entity mention till the end of a sentence. We use bi-directional LSTM encoders (Graves et al., 2013) to encode token level sequences of both context to a fixed dimensional vector. Formally, for an entity mention $m_{j,k}^i$ present in a sentence s^i , decompose s^i into a sequence of tokens $s_1^i, s_2^i, \dots, s_k^i$ for the left context, and $s_j^i, s_{j+1}^i, \dots, s_{|s^i|}^i$ 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 figure 4 (Left and right context representation).

The context representation described above is slightly different from what was proposed in (Shimaoka et al., 2016), here we include entity mention tokens within both left and right context, to explicitly encode context relative to an entity mention.

In the end, we concatenate entity mention and its context representation 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:

$$F(m_{j,k}^i, s^i) = F_m(m_{j,k}^i) \oplus F_{lc}(m_{j,k}^i, s^i) \oplus F_{rc}(m_{j,k}^i, s^i) \quad (1)$$

where \oplus denotes vector concatenation. For brevity, we will now omit the use of subscript j, k from $m_{j,k}^i$ and $l_{j,k}^i$, and will use f^i to denote feature representation for entity mention and its context obtained via equation 1.

3.4 Feature and label embeddings

Similar to Yogatama et al. (2015) and Ren et al. (2016), 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:

$$\phi_{\mathcal{M}}(f^i) = f^{iT} U; \quad \phi_{\mathcal{L}}(l_t^i) = l_t^{iT} V \quad (2)$$

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 type 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:

$$s(f^i, l_t^i) = \phi_{\mathcal{M}}(f^i) \cdot \phi_{\mathcal{L}}(l_t^i) \quad (3)$$

3.5 Optimization

We use two different loss functions to model *clean* and *noisy* entity mentions. For *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 similarity between an entity mention and label types (eq. 3). Hinge loss function has two advantages. First, it intuitively separates positive and negative labels during inference. Second, it is independent of data dependent parameter. Formally, for a given entity mention m^i and its label l^i we compute the associated loss as given by:

$$L_c(m^i, l^i) = \sum_{t \in \gamma} \max(0, 1 - s(m^i, l_t^i)) + \sum_{t \in \bar{\gamma}} \max(0, 1 + s(m^i, l_t^i)) \quad (4)$$

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 , score for all negative labels should go below -1 . However, for positive labels, as we don't know which labels are relevant to 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:

$$L_n(m^i, l^i) = \sum_{t \in \bar{\gamma}} \max(0, 1 + s(m^i, l_t^i)) + \max(0, 1 - s(m^i, l_{t^*}^i)); \quad t^* = \arg \max_{t \in \gamma} s(m^i, l_t^i) \quad (5)$$

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 (Yogatama et al., 2015; Ren et al., 2016) in a way that, we make strict absolute criteria to distinguish between positive and negative labels. Whereas in (Yogatama et al., 2015; Ren et al., 2016) 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.

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

$$\min_{\theta} O = \sum_{m \in \mathcal{M}_c} L_c(m, l) + \sum_{m \in \mathcal{M}_n} L_n(m, l) \quad (6)$$

where θ is the collection of all model parameters that needs to be learned. To jointly optimize the objective O , we use Adam (Kingma and Ba, 2014), a stochastic gradient-based optimization algorithm.

3.6 Inference

For every entity mention in 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 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.

3.7 Transfer learning

We want to investigate, whether the feature representations learnt for an entity mention are useful. We study what contribution these feature representations make to an existing feature engineering based method such as AFET. We learn the proposed model on one training dataset, namely Wiki dataset, which has the highest number of entity mentions among other datasets and use this model to generate representations that is $F(m_{j,k}^i, s^i)$ for another training and testing data. These representations, which are D_f dimensional vectors, are used as feature for an existing state-of-the-art model, AFET, in place of the hand-crafted features that were originally used. AFET model is then trained using these feature representations. We call this as feature level transfer learning. On the other hand, we also evaluate model level transfer learning, where we initialize weights of LSTM encoders for a new dataset with the weights learnt from the model trained on another dataset, namely Wiki dataset.

4 Experiments

4.1 Datasets used

We evaluate the proposed model on three publicly available datasets, provided in a pre-processed tokenized format by Ren et al. (2016). Statistics of the datasets used in this work are shown in Table 1. The details of the datasets are as follows:

Wiki/FIGER(GOLD): The training data consists of Wikipedia sentences and was automatically generated in distant supervision paradigm, by mapping hyperlinks in Wikipedia articles to Freebase. The test data, mainly consisting of sentences from news reports, was manually annotated as described in (Ling and Weld, 2012).

OntoNotes: OntoNotes dataset consists of sentences from newswire documents present in OntoNotes text corpus (Weischedel et al., 2013). DBpedia spotlight (Daiber et al., 2013) was used to automatically link entity mention in sentences to Freebase. For this corpus, manually annotated test data was shared by Gillick et al. (2014).

BBN: BBN dataset consists of sentences from Wall Street Journal articles and is completely manually annotated (Weischedel and Brunstein, 2005). Please refer to (Ren et al., 2016) for more details

Datasets	Wiki/FIGER(GOLD)	OntoNotes	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 1: Statistics of the datasets used in this work.

of the datasets.

4.2 Evaluation settings

4.2.1 Baselines

We compared the proposed model with state-of-the-art entity classification methods²: (1) **FIGER** (Ling and Weld, 2012); (2) **HYENA** (Yosef et al., 2012); (3) **AFET-NoCo** (Ren et al., 2016): AFET without data based label-label correlation modeled in loss function; (4) **AFET-CoH** (Ren et al., 2016): AFET with hierarchy based label-label correlation modeled in loss function; (5) **AFET** (Ren et al., 2016); (6) **Attentive** (Shimaoka et al., 2016): An attentive neural network based model.

We compare these baselines with variants of our proposed model: (1) **our**: complete model; (2) **our-AIIC** assuming all mentions are *clean*; (3) **our-NoM** without mention representation.

4.2.2 Experimental setup

We use Accuracy or Strict-F1 score, Macro-averaged F1 score, and Micro-averaged F1 score as metrics for evaluation. Existing methods for FETC use same measures (Ling and Weld, 2012; Yogatama et al., 2015; Shimaoka et al., 2016; Ren et al., 2016). We removed entity mentions that do not have any label in training as well as test set. We also remove entity mentions that have spurious indices (i.e entity mention length of 0).³ 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 same hyperparameters five times and report their performance in terms of micro-F1 score on the development set as

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

²Whenever possible, the baselines result are reported from (Ren et al., 2016), otherwise we re-implemented baseline methods based on description available in corresponding papers.

³The code to replicate the work is available at <https://github.com/abhipef/fnet>

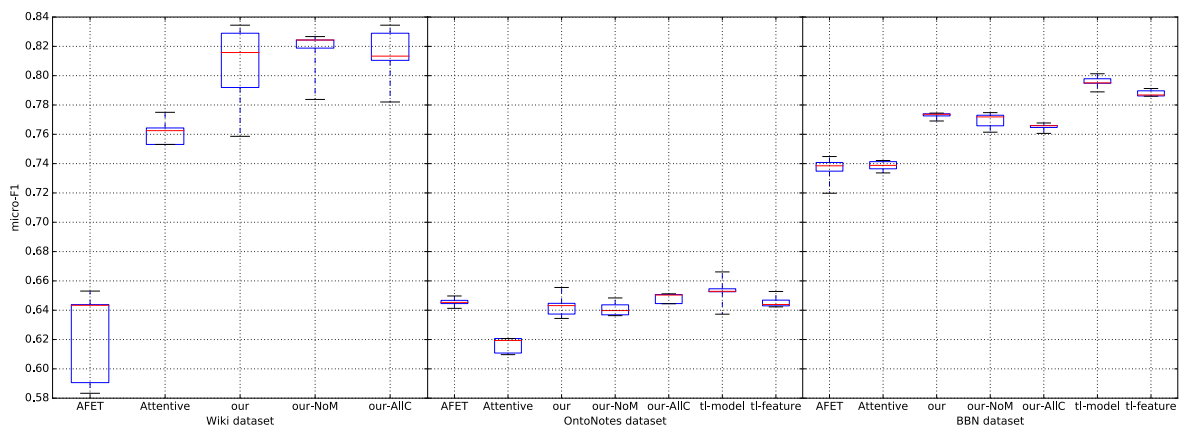


Figure 5: These box-plots show the performance of different baselines on validation set. The red line, boxes and whiskers indicate the median, quartiles and range.

shown in Figure 5. On Wiki dataset, we observed a large variance in performance as compared to other two datasets. This might be because of the fact that Wiki 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 neural network based models in this paper used 300 dimensional pre-trained word embeddings distributed by Pennington et al. (2014). The hidden-layer size of word level bi-directional LSTM was 100, and that of character level LSTM was 200. Vectors for character embeddings were randomly initialized and were of size 200. We use dropout with the probability of 0.5 on the output of LSTM encoders. The embedding dimension used was 500. We use Adam (Kingma and Ba, 2014) as optimization method with learning rate of 0.0005-0.001 and mini-batch size in the range of 800 to 1500. The proposed model and some of the baselines were implemented using TensorFlow⁴ framework.

4.3 Transfer learning

In feature level transfer learning, we use the best performing proposed model trained on Wiki dataset to generate representations that is D_f dimensional vector for every entity mention present in the train, development, and test set of the BBN and the OntoNotes dataset. Figure 4 illustrates an example for 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 hyper-parameters were tuned on the development set. These results are shown in table 2 as **feature level transfer-learning**.

In model level transfer learning, we use the learnt weights of LSTM encoders from the best performing proposed model trained on Wiki dataset and initialize the LSTM encoders of the same model with these weights while training on BBN and OntoNotes datasets. These results are shown in table 2 as **model level transfer learning**.

4.4 Performance comparison and analysis

Table 2 shows the results of the proposed method, its variants and the baseline methods.

Comparison with other feature learning methods: The proposed model and its variants (**our-AIIC**, **our-NoM**) perform better than the existing feature learning method by Shimaoka et al. (2016) (**Attentive**), consistently on all datasets. This indicates 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 Micro-F1 and 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 Wiki and BBN dataset in terms of Micro-F1 evaluation metric. This indicates benefits of feature learning as well as data driven label-label correlation. We do a type-wise perfor-

⁴<http://tensorflow.org/>

Typing methods	Wiki/FiGER(GOLD)			OntoNotes			BBN		
	Acc.	Ma-F1	Mi-F1	Acc.	Ma-F1	Mi-F1	Acc.	Ma-F1	Mi-F1
FIGER [*] (Ling and Weld, 2012)	0.474	0.692	0.655	0.369	0.578	0.516	0.467	0.672	0.612
HYENA [*] (Yosef et al., 2012)	0.288	0.528	0.506	0.249	0.497	0.446	0.523	0.576	0.587
AFET-NoCo [*] (Ren et al., 2016)	0.526	0.693	0.654	0.486	0.652	0.594	0.655	0.711	0.716
AFET-CoH [*] (Ren et al., 2016)	0.433	0.583	0.551	0.521	0.680	0.609	0.657	0.703	0.712
AFET [*] (Ren et al., 2016)	0.533	0.693	0.664	0.551	0.711	0.647	0.670	0.727	0.735
AFET ^{†‡} (Ren et al., 2016)	0.509	0.689	0.653	0.553	0.712	0.646	0.683	0.744	0.747
Attentive [†] (Shimaoka et al., 2016)	0.581	0.780	0.744	0.473	0.655	0.586	0.484	0.732	0.724
our-AIIC [†]	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 2: Performance analysis of entity classification methods on the three datasets.

mance comparison on OntoNotes dataset in subsection 4.5.

Comparison with variants of our model: The proposed model performs better on all dataset as compared to **our-AIIC** in terms of micro-F1 score. However, we find the performance difference on Wiki and OntoNotes dataset is not statistically significant. We investigated it further and found that across all three datasets, there exist only few entity types for which more than 85% of entity mentions are noisy. These types consist of approximately 3-4% of test set, and our model fails on these types (zero micro-F1 score). However, **our-AIIC** performs relatively well on these types. Examples of such types are: */building*, */person/political_figure*, */GPE/STATE_PROVINCE*. This indicates two limitations of the proposed model. First, the separating of clean and noisy mentions based on the hierarchy has its own 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. We will try to address these limitations in our future work. Compared with **our-NoM**, the proposed model performs slightly better across all datasets in terms of micro-F1 score.

Feature level transfer learning analysis: We observed 4.5% performance increase in micro-F1 score of AFET on BBN dataset, after replacing hand-crafted features with feature representations generated by the proposed model. This indicates usefulness of the learnt feature representations. However, if we repeat the same process with OntoNotes dataset, there is only a subtle change in performance. This is majorly because of the data distribution of OntoNotes dataset is different from

that of Wiki dataset. This issue is discussed in the next subsection.

Model level transfer learning analysis: In model level transfer learning, sharing knowledge from similar dataset (Wiki to BBN) increases the performance by 3.8% in terms of micro-F1 score. However, sharing knowledge from Wiki to OntoNotes dataset slightly increases the performance by 0.4% in terms of micro-F1 score.

4.5 Case analysis: OntoNotes dataset

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

Statistics of the dataset (Table 1) indicates that presence of pronominal or other kinds of mentions are relatively higher in OntoNotes (6.78% in test set) than the other two datasets (0% in test set). Examples of such mentions are *100 people*, *It*, *the director*, etc. Table 3 shows 20 randomly sampled entity mentions from test set of OntoNotes datasets. Some of these mentions are very generic and likely to be dependent on

^{*}These results are from (Ren et al., 2016) that also uses 10% of the test set as development set and the remaining for evaluation.

[‡]We used the publicly available code distributed by Ren et al. (2016).

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

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 test set increases the performance of all feature learning methods by around 3%.

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: 20 randomly sampled entity mentions present in the test set of OntoNotes dataset.

Next we analyse where the proposed model is failing as compared to AFET. For this, we look at type-wise performance for the top-10 most frequent types in the OntoNotes test dataset. Results are shown in Table 4. 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 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*. 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 2a, the use of label-label correlation depends on appropriate values of parameters which vary from one dataset to another.

Label type	Support	our			AFET		
		Prec.	Rec.	F-1	Prec.	Rec.	F-1
<i>/other</i>	42.6%	0.838	0.809	0.823	0.774	0.962	0.858
<i>/organization</i>	11.0%	0.588	0.490	0.534	0.903	0.273	0.419
<i>/person</i>	9.9%	0.559	0.467	0.508	0.669	0.352	0.461
<i>/organization/company</i>	7.8%	0.932	0.166	0.282	1.0	0.127	0.225
<i>/location</i>	7.5%	0.687	0.796	0.737	0.787	0.609	0.687
<i>/organization/government</i>	2.1%	0	0	0	0	0	0
<i>/location/country</i>	2.0%	0.783	0.614	0.688	0.838	0.498	0.625
<i>/other/legal</i>	1.8%	0	0	0	0	0	0
<i>/location/city</i>	1.8%	0.919	0.610	0.733	0.816	0.637	0.715
<i>/person/political_figure</i>	1.6%	0	0	0	0	0	0

Table 4: Performance analysis of the proposed model and AFET on top 10 (in terms of type frequency) types present in OntoNotes dataset.

5 Conclusion and Future Work

In this paper, we propose a neural network based model for the task of fine-grained entity classification. The proposed model learns representations for entity mention, its context and incorporate label noise information in a variant of 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, OntoNotes dataset has a different distribution of entity mentions compared with other two datasets. Second, if 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 label types that are very noisy.

Future work could analyse the effect of label noise reduction techniques on the proposed model, revisiting the definition of clean and noisy labels and modeling label-label correlation in a principled way that is not dependent on dataset specific parameters.

Acknowledgments

We thank the anonymous reviewers for their invaluable and insightful comments. Abhishek is supported by MHRD fellowship, Government of India. We acknowledge the use of computing resources made available from the Board of Research in Nuclear Science (BRNS), Dept. of Atomic Energy (DAE), Govt. of India sponsored project (No.2013/13/8-BRNS/10026) by Dr. Aryabartta Sahu at Department of Computer Science and Engineering, IIT Guwahati.

References

- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Semantic Web Conference*, ISWC’07/ASWC’07, pages 722–735, Berlin, Heidelberg, Springer-Verlag.
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A

- collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.
- Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.
- Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press.
- Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA. ACM.
- Xin Dong, Evgeniy Gabilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmman, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 601–610, New York, NY, USA. ACM.
- Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1243–1249. AAAI Press.
- Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*.
- Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780, November.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901, Doha, Qatar, October. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.
- Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Thomas Lin, Mausam, and Oren Etzioni. 2012. No noun phrase left behind: Detecting and typing un-linkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 893–903, Jeju Island, Korea, July. Association for Computational Linguistics.
- Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 94–100. AAAI Press.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June. Association for Computational Linguistics.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. 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*, pages 1003–1011, Suntec, Singapore, August. Association for Computational Linguistics.
- T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2302–2310. AAAI Press.
- Lili Mou, Ran Jia, Yan Xu, Ge Li, Lu Zhang, and Zhi Jin. 2016. Distilling word embeddings: An encoding approach. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, CIKM '16, pages 1977–1980, New York, NY, USA. ACM.

- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Lorien Y. Pratt. 1993. Discriminability-based transfer between neural networks. In *Advances in Neural Information Processing Systems 5*, pages 204–211, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado, June. Association for Computational Linguistics.
- Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378, Austin, Texas, November. Association for Computational Linguistics.
- Lei Shi, Rada Mihalcea, and Mingjun Tian. 2010. Cross language text classification by model translation and semi-supervised learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1057–1067, Cambridge, MA, October. Association for Computational Linguistics.
- Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 69–74, San Diego, CA, June. Association for Computational Linguistics.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, pages 697–706, New York, NY, USA. ACM.
- Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. 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*, pages 477–487, Montréal, Canada, June. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Dong Wang and Thomas Fang Zheng. 2015. Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1225–1237. IEEE.
- Ralph Weischedel and Ada Brunstein. 2005. BBN Pronoun Coreference and Entity Type Corpus LDC2005T33. *Linguistic Data Consortium, Philadelphia*, 112.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.
- Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. 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)*, pages 291–296, Beijing, China, July. Association for Computational Linguistics.
- Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. HYENA: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India, December. The COLING 2012 Organizing Committee.