

Noise Mitigation for Neural Entity Typing and Relation Extraction

Yadollah Yaghoobzadeh* and Heike Adel* and Hinrich Schütze

* *These authors contributed equally to this work*

Center for Information and Language Processing

LMU Munich, Germany

yadollah|heike@cis.lmu.de

Abstract

In this paper, we address two different types of noise in information extraction models: noise from distant supervision and noise from pipeline input features. Our target tasks are entity typing and relation extraction. For the first noise type, we introduce multi-instance multi-label learning algorithms using neural network models, and apply them to fine-grained entity typing for the first time. Our model outperforms the state-of-the-art supervised approach which uses global embeddings of entities. For the second noise type, we propose ways to improve the integration of noisy entity type predictions into relation extraction. Our experiments show that probabilistic predictions are more robust than discrete predictions and that joint training of the two tasks performs best.

1 Introduction

Knowledge bases (KBs) are important resources for natural language processing tasks like question answering and entity linking. However, KBs are far from complete (e.g., Socher et al. (2013)). Therefore, methods for automatic knowledge base completion (KBC) are beneficial. Two subtasks of KBC are *entity typing (ET)* and *relation extraction (RE)*. We address both tasks in this paper.

As in other information extraction tasks, obtaining labeled training data for ET and RE is challenging. The challenge grows as labels become more fine-grained. Therefore, distant supervision (Mintz et al., 2009) is widely used. It reduces the need for manually created resources. Distant supervision assumes that if an entity has a type (resp. two entities have a relationship) in a KB, then all sentences mentioning that entity (resp. those

two entities) express that type (resp. that relationship). However, that assumption is too strong and gives rise to many *noisy* labels. Different techniques to deal with that problem have been investigated. The main technique is multi-instance (MI) learning (Riedel et al., 2010). It relaxes the distant supervision assumption to the assumption that at least one instance of a bag (collection of all sentences containing the given entity/entity pair) expresses the type/relationship given in the KB. Multi-instance multi-label (MIML) learning is a generalization of MI in which one bag can have several labels (Surdeanu et al., 2012).

Most MI and MIML methods are based on hand crafted features. Recently, Zeng et al. (2015) introduced an end-to-end approach to MI learning based on neural networks. Their MI method takes the most confident instance as the prediction of the bag. Lin et al. (2016) further improved that method by taking other instances into account as well; they proposed MI learning based on selective attention as an alternative way of relaxing the impact of noisy labels on RE. In selective attention, a weighted average of instance representations is calculated first and then used to compute the prediction of a bag.

In this paper, we introduce two multi-label versions of MI. (i) *MIML-MAX* takes the maximum instance for each label. (ii) *MIML-ATT* applies, for each label, selective attention to the instances. We apply MIML-MAX and MIML-ATT to fine-grained ET. In contrast to RE, the ET task we consider contains a larger set of labels, with a variety of different granularities and hierarchical relationships. We show that MIML-ATT deals well with noise in corpus-level ET and improves or matches the results of a supervised model based on global embeddings of entities.

The second type of noise we address in this paper influences the integration of ET into RE. It has

been shown that adding entity types as features improves RE models (cf. Ling and Weld (2012), Liu et al. (2014)). However, noisy training data and difficulties of classification often cause wrong predictions of ET and, as a result, noisy inputs to RE. To address this, we propose a joint model of ET and RE and compare it with methods that integrate ET results in a strict pipeline. The joint model performs best. Among the pipeline models, we show that using probabilities instead of binary decisions better deals with noise (i.e., possible ET errors).

To sum up, our contributions are as follows. (i) We introduce new algorithms for MIML using neural networks. (ii) We apply MIML to fine-grained entity typing for the first time and show that it outperforms the state-of-the-art supervised method based on entity embeddings. (iii) We show that a novel way of integrating noisy entity type predictions into a relation extraction model and joint training of the two tasks lead to large improvements of RE performance.

We release code and data for future research.¹

2 Related Work

Noise mitigation for distant supervision. Distant supervision can be used to train information extraction systems, e.g., in relation extraction (e.g., Mintz et al. (2009), Riedel et al. (2010), Hoffmann et al. (2011), Zeng et al. (2015)) and entity typing (e.g., Ling and Weld (2012), Yogatama et al. (2015), Dong et al. (2015)). To mitigate the noisy label problem, multi-instance (MI) learning has been introduced and applied in relation extraction (Riedel et al., 2010; Ritter et al., 2013). Surdeanu et al. (2012) introduced multi-instance multi-label (MIML) learning to extend MI learning for multi-label relation extraction. Those models are based on manually designed features. Zeng et al. (2015) and Lin et al. (2016) introduced MI learning methods for neural networks. We introduce MIML algorithms for neural networks. In contrast to most MI/MIML methods, which are applied in relation extraction, we apply MIML to the task of fine-grained entity typing. Ritter et al. (2013) applied MI on a Twitter dataset with ten types. Our dataset has a larger number of classes or types (namely 102) and input examples, compared to that Twitter dataset and also to the most widely used datasets for evaluating MI (cf. Riedel et al. (2010)). This makes our setup more challenging because of dif-

ferent dependencies and the multi-label nature of the problem. Also, there seems to be a difference between how entity relations and entity types are expressed in text. Our experiments support that hypothesis.

Knowledge base completion (KBC). Most KBC systems focus on identifying triples $R(e_1, r, e_2)$ missing from a KB (Nickel et al., 2012; Bordes et al., 2013; Weston et al., 2013; Socher et al., 2013; Jiang et al., 2012; Riedel et al., 2013; Wang et al., 2014). Work on entity typing or unary relations for KBC is more recent (Yao et al., 2013; Neelakantan and Chang, 2015; Yaghoobzadeh and Schütze, 2015; Yaghoobzadeh et al., 2017). In this paper, we build a KBC system for unary and binary relations using contextual information of words and entities.

Named entity recognition (NER) and typing. NER systems (e.g., Finkel et al. (2005), Collobert et al. (2011)) used to consider only a small set of entity types. Recent work also addresses fine-grained NER (Yosef et al., 2012; Ling and Weld, 2012; Yogatama et al., 2015; Dong et al., 2015; Del Corro et al., 2015; Ren et al., 2016a; Ren et al., 2016b; Shimaoka et al., 2016). Some of this work (cf. Yogatama et al. (2015), Dong et al. (2015)) treats entity segment boundaries as given and classifies mentions into fine-grained types. We make a similar assumption, but in contrast to NER, we evaluate on the corpus-level entity typing task of Yaghoobzadeh and Schütze (2015); thus, we do not need test sentences annotated with context dependent entity types. This task was also used to evaluate embedding learning methods (Yaghoobzadeh and Schütze, 2016).

Entity types for relation extraction. Several studies have integrated entity type information into relation extraction – either coarse-grained (Hoffmann et al., 2011; Zhou et al., 2005) or fine-grained (Liu et al., 2014; Du et al., 2015; Augenstein et al., 2015; Vlachos and Clark, 2014; Yao et al., 2010; Ling and Weld, 2012) entity types. In contrast to most of this work, but similar to Yao et al. (2010), we do not incorporate binary entity type values, but probabilistic outputs. Thus, we allow the relation extraction system to compensate for errors of entity typing. Additionally, we compare this approach to various other possibilities, to investigate which approach performs best. Yao et al. (2010) found that joint training of entity typing and relation extraction is better than a pipeline

¹cistern.cis.lmu.de

model; we show that this result also holds for neural network models and when the number of entity types is large.

3 MIML Learning for Entity Typing

Entity typing (ET) is the task of finding, for each named entity, a set of types or classes that it belongs to, e.g., “author” and “politician” for “Obama”. Our goal is corpus-level prediction of entity types. We use the entity-type information from a KB and annotated contexts of entities in a corpus to estimate $P(t|e)$, the probability that entity e has type t .

More specifically, consider an entity e and $B = \{c_1, c_2, \dots, c_q\}$, the set of q contexts of e in the corpus. Each c_i is an instance of e and since e can have several labels, it is a multi-instance multi-label (MIML) learning problem. We address MIML using neural networks by representing each context as a vector $\vec{c}_i \in \mathbb{R}^h$, and learn $P(t|e)$ from the set of contexts of entity e . In the following, we first describe our MIML algorithms and then explain how \vec{c}_i is computed.

Notations and definitions. Lowercase letters (e.g., e) refer to variables. Lowercase letters with an upper arrow (e.g., \vec{e}) are vectors. We define BCE, binary cross entropy, as follows where y is a binary variable and \hat{y} is a real valued variable between 0 and 1.

$$\text{BCE}(y, \hat{y}) = -\left(y \log(\hat{y}) + (1-y)(1 - \log(\hat{y}))\right)$$

3.1 Algorithms

Distant supervision. The basic way to estimate $P(t|e)$ is based on distant supervision with learning the type probability of each c_i individually, by making the assumption that each c_i expresses all labels of e . Therefore, we define the context-level probability function as:

$$P(t|c_i) = \sigma(\vec{w}_t \vec{c}_i + b_t) \quad (1)$$

where $\vec{w}_t \in \mathbb{R}^h$ is the output weight vector and b_t is the bias scalar for type t . The cost function is defined based on binary cross entropy:

$$L(\theta) = \sum_c \sum_t \text{BCE}(y_t, P(t|c)) \quad (2)$$

where y_t is 1 if entity e has type t otherwise 0. To compute $P(t|e)$ at prediction time, i.e., $P_{\text{pred}}(t|e)$,

the context-level probabilities must be aggregated. Average is the usual way of doing that:

$$P_{\text{pred}}(t|e) = \frac{1}{q} \sum_{i=1}^q P(t|c_i) \quad (3)$$

Multi-instance multi-label. The distant supervision assumption is that *all* contexts of an entity with type t are contexts of t ; e.g., we label all contexts mentioning “Barack Obama” with all of his types. Obviously, the labels are incorrect or *noisy* for some contexts. Multi-instance multi-label (MIML) learning addresses this problem. We apply MIML to fine-grained ET for the first time. Our assumption is: if entity e has type t , then there is at least one context of e in the corpus in which e occurs as type t . So, we apply this assumption during training with the following estimation of the type probability of an entity:

$$P(t|e) = \max_{1 \leq i \leq q} P(t|c_i) \quad (4)$$

which means we take the **maximum** probability of type t over all contexts of entity e as $P(t|e)$. We call this approach **MIML-MAX**.

MIML-MAX picks the most confident context for t , ignoring the probabilities of all the other contexts. Apart from missing information, this can be especially harmful if the entity annotations in the corpus are the result of an entity linking system. In that case, the most confident context might be wrongly linked to the entity. So, it can be beneficial to leverage all contexts into the final prediction, e.g., by **averaging** the type probabilities of all contexts of entity e :

$$P(t|e) = \frac{1}{q} \sum_{i=1}^q P(t|c_i) \quad (5)$$

We call this approach **MIML-AVG**. We also propose a combination of the maximum and average, which uses MIML-MAX (Eq. 4) in training and MIML-AVG (Eq. 5) in prediction. We call this approach **MIML-MAX-AVG**.

MIML-AVG treats every context equally which might be problematic since many contexts are irrelevant for a particular type. A better way is to weight the contexts according to their similarity to the types. Therefore, we propose using selective **attention** over contexts as follows and call this approach **MIML-ATT**. MIML-ATT is the multi-label version of the selective attention method proposed in Lin et al. (2016). To compute the type

Model	Train	Prediction
MIML-MAX	MAX	MAX
MIML-AVG	AVG	AVG
MIML-MAX-AVG	MAX	AVG
MIML-ATT	ATT	ATT

Table 1: Different MIML algorithms for entity typing, and the aggregation function they use to get corpus-level probabilities.

probability for e , we define:

$$P(t|e) = \sigma(\vec{w}_t \vec{a}_t + b_t) \quad (6)$$

where $\vec{w}_t \in \mathbb{R}^h$ is the output weight vector and b_t the bias scalar for type t , and \vec{a}_t is the aggregated representation of all contexts c_i of e for type t , computed as follows:

$$\vec{a}_t = \sum_i \alpha_{i,t} \vec{c}_i \quad (7)$$

where $\alpha_{i,t}$ is the attention score of context c_i for type t and $\vec{a}_t \in \mathbb{R}^h$ can be interpreted as the representation of entity e for type t .

$\alpha_{i,t}$ is defined as:

$$\alpha_{i,t} = \frac{\exp(\vec{c}_i \mathbf{M} \vec{t})}{\sum_{j=1}^q \exp(\vec{c}_j \mathbf{M} \vec{t})} \quad (8)$$

where $\mathbf{M} \in \mathbb{R}^{h \times d_t}$ is a weight matrix that measures the similarity of \vec{c} and \vec{t} . $\vec{t} \in \mathbb{R}^{d_t}$ is the representation of type t .

Table 1 summarizes the differences of our MIML methods with respect to the aggregation function they use to get corpus-level probabilities. For optimization of all MIML methods, we use the binary cross entropy loss function,

$$L(\theta) = \sum_e \sum_t \text{BCE}(y_t, P(t|e)) \quad (9)$$

In contrast to the loss function of distant supervision in Eq. 2, which iterates over all *contexts*, we iterate over all *entities* here.

3.2 Context Representation

To produce a high-quality context representation \vec{c} , we use convolutional neural networks (CNNs).

The first layer of the CNN is a *lookup table* that maps each word in c to an embedding of size d . The output of the lookup layer is a matrix $E \in \mathbb{R}^{d \times s}$ (the embedding layer), where s is the context size (a fixed number of words).

The CNN uses n filters of different window widths w to narrowly convolve E . For each of the n filters $H \in \mathbb{R}^{d \times w}$, the result of applying H to matrix E is a feature map $\vec{m} \in \mathbb{R}^{s-w+1}$:

$$m[i] = g(E_{:,i:i+w-1} \odot H) \quad (10)$$

where g is the *relu* function, \odot is the Frobenius product, $E_{:,i:i+w-1}$ are the columns i to $i+w-1$ of E and $1 \leq w \leq k$ are the window widths we consider. Max pooling then gives us one feature for each filter and the concatenation of those features is the CNN representation of c .

As it is shown in the entity typing part of Figure 1, we apply the CNN to the left and right context of the entity mention and the concatenation $\vec{\phi}(c) \in \mathbb{R}^{2n}$ is fed into a multi-layer perceptron (MLP) to get the final context representation $\vec{c} \in \mathbb{R}^h$:

$$\vec{c} = \tanh(\mathbf{W}_h \vec{\phi}(c)) \quad (11)$$

4 Type-aware Relation Extraction

Relation extraction (RE) is mostly defined as finding relations between pairs of entities, for instance, finding the relation “president-of” between “Obama” and “USA”. Given a set of q contexts for an entity pair z , $B = \{c_1, c_2, \dots, c_q\}$ in the corpus, we learn $P(r|z)$, which is the probability of relation r for z . We assume that each z has one relation $r(z)$. Each c_i is represented by a vector $\vec{c}_i \in \mathbb{R}^h$, which is our type-aware representation of context described in Section 4.1.

To learn $P(r|z)$, we use the multi-instance (MI) learning method of Zeng et al. (2015):

$$P(r|c_i) = \text{softmax}(\mathbf{W}_{\text{out}} \vec{c}_i), \quad (12)$$

$$P(r|z) = \max_{1 \leq i \leq q} P(r|c_i)$$

where $P(r|c_i)$ is the probability of relation r for context c_i . The cost function we optimize is:

$$L(\theta) = - \sum_z \log P(r(z)|z)$$

4.1 Context Representation

Similar to our entity typing system, we apply CNNs to compute the context representation $\vec{\phi}(c)$. In particular, we use Adel et al. (2016)’s CNN. It uses an input representation designed for RE. Each sentence is split into three parts: left of the relation arguments, between the relation arguments

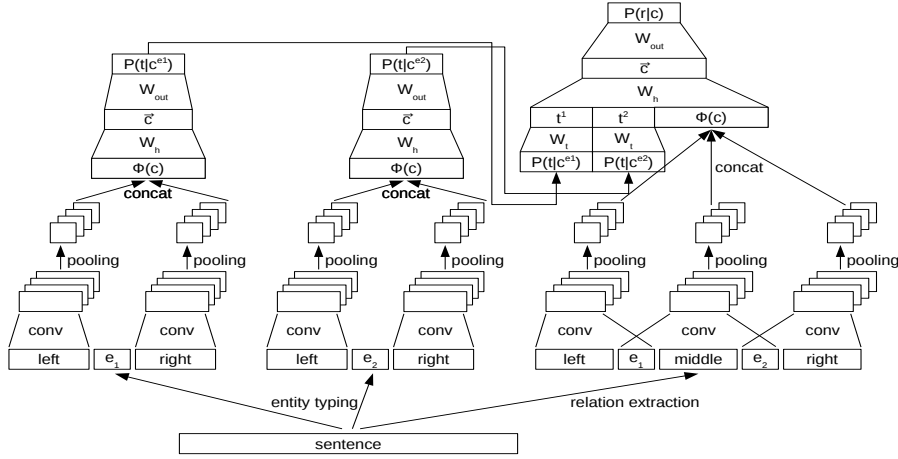


Figure 1: Our architecture for joint entity typing and relation extraction

and right of the relation arguments. The parts “overlap”, i.e., the left (resp. right) argument is included in both left (resp. right) and middle parts. For each of the three parts, convolution and 3-max pooling (Kalchbrenner et al., 2014) is performed. The context representation $\vec{\phi}(c) \in \mathbb{R}^{3 \cdot 3 \cdot n}$ is the concatenation of the pooling results.

4.1.1 Integration of Entity Types

We concatenate the entity type representations $t^1 \in \mathbb{R}^\tau$ and $t^2 \in \mathbb{R}^\tau$ of the relation arguments to the CNN representation of the context, $\vec{\phi}(c)$:

$$\vec{\phi}(c)' = [\vec{\phi}(c) : t^1 : t^2] \quad (13)$$

Our context representation \vec{c} is then:

$$\vec{c} = \tanh(\mathbf{W}_h \vec{\phi}(c)') \quad (14)$$

where $\mathbf{W}_h \in \mathbb{R}^{h \times (3 \cdot 3 \cdot n + 2\tau)}$ is the weight matrix. This is also depicted in Figure 1, right column, third layer from the top: $t^1, t^2, \vec{\Phi}(c)$. We calculate t^1 and t^2 from the predictions of the entity typing model with the following transformation:

$$t^k = f(\mathbf{W}_t [P(t_1|c^{e_k}) \dots P(t_T|c^{e_k})]) \quad (15)$$

where c^{e_k} is the context of e_k , $\mathbf{W}_t \in \mathbb{R}^{\tau \times T}$ is a weight matrix (learned from corpus or during training) and f is a function (identity or tanh). With the transformation \mathbf{W}_t , the model can combine predictions for different types to learn better internal representations t^1 and t^2 . The choices of \mathbf{W}_t and f depend on the different representations we investigate and describe in the following.

(1) Pipeline: We integrate entity types into the RE model, using the output of ET in a pipeline

model (see Eq. 15). We test the following representations of t^k , $k \in \{1, 2\}$. **PREDICTED-HIDDEN:** \mathbf{W}_t from Eq. 15 is learned during training and f is tanh. That means that a hidden layer learns representations based on the predictions $P(t_1|c^{e_k}) \dots P(t_T|c^{e_k})$. **BINARY-HIDDEN:** This is the binarization of the input of PREDICTED-HIDDEN, i.e., each probability estimate is converted to 0 or 1 (with a threshold of 0.5). **BINARY:** t^k is the binary vector itself (used by Ling and Weld (2012)). **WEIGHTED:** The columns of matrix \mathbf{W}_t from Eq. 15 are the distributional embeddings of types trained on the corpus (see Section 5.1). f is the identity function.

(2) Joint model: As an alternative to the pipeline model, we investigate integrating entity typing into RE by jointly training both models. We use the architecture depicted in Figure 1. The key difference to the pipeline model PREDICTED-HIDDEN is that we learn $P(t|c)$ and $P(r|c)$ jointly, called **JOINT-TRAIN**. We compare JOINT-TRAIN to other models, including the pipeline models.

During training of JOINT-TRAIN, we compute the cost of the ET model for typing the first entity $L_1(\theta_T)$, the cost for typing the second entity $L_2(\theta_T)$ and the cost of the RE model for assigning a relation to the two entities $L(\theta_R)$. Then, we combine those costs with a weight γ which is tuned on the development set:

$$L(\theta) = \sum_z (L_1(\theta_T) + L_2(\theta_T) + \gamma \cdot L(\theta_R)),$$

$$L_i(\theta_T) = \sum_t \text{BCE}(y_t^{e_i}, P(t|c^{e_i})),$$

$$L(\theta_R) = -\log P(r(z)|z)$$

GOV.GOV_agency.jurisdiction	PPL.PER.children
GOV.us_president.vice_president	PPL.PER.nationality
PPL.deceased.PER.place_of_death	PPL.PER.religion
ORG.ORG.place_founded	PPL.PER.place_of_birth
ORG.ORG_founder.ORGs_founded	NA (no relation)
LOC.LOC.containedby	

Table 2: Selected relations for relation extraction; PPL = people, GOV = government

$P(r|z)$ is computed based on Eq. 12.

Note that based on this equation, the ET parameters are optimized on the contexts of the RE examples, which are a subset of all training examples of ET. However in the pipeline models, ET is trained on the whole training set used for typing. Also note that in JOINT-TRAIN we do not use MIML for the ET part but a distant supervised cost function.

5 Experimental Data, Setup and Results

For entity typing, we use CF-FIGMENT (URL, 2016b), a dataset published by Yaghoobzadeh and Schütze (2015). CF-FIGMENT is derived from a version of ClueWeb (URL, 2016c) in which Freebase entities are annotated using FACC1 (URL, 2016d; Gabrilovich et al., 2013). CF-FIGMENT contains 200,000 Freebase entities that were mapped to 102 FIGER types (Ling and Weld, 2012), divided into train (50%), dev (20%) and test (30%); and a set of 4,300,000 sentences (contexts) containing those entities.

For relation extraction, we first select the ten most frequent relations (plus NA for no relation according to Freebase) of entity pairs in CF-FIGMENT. We ensure that the entity pairs have at least one context in CF-FIGMENT. This results in 5815, 3054 and 6889 unique entity pairs for train, dev and test.² Dev and test set sizes are 124,462 and 556,847 instances. For the train set, we take a subsample of 135,171 sentences. The entity and sentence sets of CF-FIGMENT were constructed to ensure that entities in the entity test set do not occur in the sentence train and dev sets; that is, a sentence was assigned to the train set only if all entities it contains are train entities.¹

5.1 Word, Entity and Type Embeddings

We use 100-dimensional word embeddings to initialize the input layer of ET and RE. Embeddings

²We only assign those entity pairs to test (resp. dev, resp. train) for which both constituting entities are in the ET test (resp. dev, resp. train) set.

are kept fixed during training. Since we need embeddings for words, entities and types in the same space, we process ClueWeb+FACC1 (corpus with entity information) as follows. For each sentence s , we add two copies: s itself, and a copy in which each entity is replaced with its notable type, the most important type according to Freebase. We process train, dev and test this way, but do not replace test entities with their notable type because the types of test entities are unknown in our application scenario. We run word2vec (Mikolov et al., 2013) on the resulting corpus to learn embeddings for words, entities and types. Note that our application scenario is that we are given an unannotated input corpus and our system then extracts entity types and relations from this input corpus to enhance the KB.

5.2 Entity Typing Experiments

Entity context setup. We use a window size of 5 on each side of the entity mentions. Following Yaghoobzadeh and Schütze (2015), we replace other entities occurring in the context with their Freebase notable type mapped to FIGER.

Models. Yaghoobzadeh and Schütze (2015) applied a multi-layer perceptron (MLP) architecture to create context representations. Therefore, we use an MLP baseline to compute the context representation $\vec{\phi}(c)$. The input to the MLP model is a concatenation of context word embeddings. As an alternative to MLP, we also train a CNN (see Section 3.2) to compute context representations. We run experiments with MLP and CNN, each trained with standard distant supervision and with MIML.

EntEmb and FIGMENT baselines. Following Yaghoobzadeh and Schütze (2015), we also learn entity embeddings and classify those embeddings to types, i.e., instead of distant supervision, we classify entities based on aggregated information represented in entity embeddings. An MLP with one hidden layer is used as classifier. We call that model EntEmb. We join the results of EntEmb with our best model (line 13 in Table 3), similar to the joint model (FIGMENT) in Yaghoobzadeh and Schütze (2015).

We use the same **evaluation measures** as Ling and Weld (2012), Yaghoobzadeh and Schütze (2015) and Neelakantan and Chang (2015) for entity typing: precision at 1 ($P@1$), which is the accuracy of picking the most confident type for each entity, micro average F_1 of all entity-type

	$P@1$	F_1	F_1	F_1	MAP
	all	all	head	tail	
1 MLP	74.3	69.1	74.8	52.5	42.1
2 MLP+MIML-MAX	74.7	59.2	50.7	46.8	41.3
3 MLP+MIML-AVG	77.2	70.6	74.9	56.2	45.0
4 MLP+MIML-MAX-AVG	75.2	71.2	76.4	56.0	47.1
5 MLP+MIML-ATT	81.0	72.0	76.9	59.1	48.8
6 CNN	78.4	72.2	77.3	56.3	47.6
7 CNN+MIML-MAX	78.6	62.2	53.5	49.7	46.6
8 CNN+MIML-AVG	80.8	73.5	77.7	59.2	50.4
9 CNN+MIML-MAX-AVG	79.9	74.3	79.2	59.8	53.3
10 CNN+MIML-ATT	83.4	75.1	79.4	62.2	55.2
11 EntEmb	80.8	73.3	79.9	57.4	56.6
12 FIGMENT	81.6	74.3	80.3	60.1	57.0
13 CNN+MIML-ATT+EntEmb	85.4	78.2	83.3	66.2	64.8

Table 3: $P@1$, Micro F_1 for all, head and tail entities and MAP results for entity typing.

assignments and mean average precision (MAP) over types. We could make assignment decisions based on the standard criterion $p > \theta$, $\theta = 0.5$, but we found that tuning θ improves results. For each probabilistic classifier and each type, we set θ to the value that maximizes performance on dev.

Results. Table 3 shows results for $P@1$, micro F_1 and MAP. For F_1 , we report separate results for all, head (frequency higher than 100) and tail (frequency less than 5) entities.

Discussion. The improvement of CNN (6) compared to MLP (1) is not surprising considering the effectiveness of CNNs in finding position independent local features, compared to the flat representation of MLP. Lines 2-5 and 7-10 show the results of different MIML algorithms for MLP and CNN, respectively. Considering micro F1 for all entities as the most importance measure, the trend is similar in both MLP and CNN for MIML methods: ATT > MAX-AVG > AVG > MAX.

MAX is worse than even basic distant supervised models, especially for micro F1. MAX predictions are based on only one context of each entity (for each type), and the results suggest that this is harmful for entity typing. This is in contradiction with the previous results in RE (cf. Zeng et al. (2015)) and suggests that there might be a significant difference between expressing types of entities and relations between them in text. Related to this, MAX-AVG which averages the type probabilities at prediction time improves MAX by a large margin. Averaging the context probabilities seems to be a way to smooth the entity type probabilities. MAX-AVG models are also better than the corresponding models with AVG that train and predict with averaging. This is due to the fact that AVG gives equal weights to all context probabilities both in training and prediction. ATT uses

	ATT			MAX		
	person	author	doctor	person	author	doctor
... /m/024g5w , and DOCTOR into disease will be ...						
... whooping cough , and kidney disease (/m/024g5w 's disease ...						
In 7 , DOCTOR and /m/024g5w write Elements of the ...						
book but his catarrhal bronchitis turned to /m/024g5w 's disease and ...						
It has cured /m/024g5w 's disease that could be traced to ...						
two clinical wards so /m/024g5w can carry on intensive study ...						
/m/024g5w , who once explored LOCATION-COUNTRY and wrote up his ...						
... is /m/024g5w , who is collecting and painstakingly recording ...						

Figure 2: MIML-ATT and MIML-MAX scores for the example entity /m/024g5w.

weighted contexts in both training and prediction and that is probably the reason for its effectiveness over all other MIML algorithms. Overall, using attention (ATT) significantly improves the results of both MLP and CNN models.

CNN+MIML-ATT (10) performs comparable to EntEmb (11), with better micro F1 on all and tail entities and worse MAP and micro F1 on head entities. These two models have different properties, e.g., MIML is also able to type each mention of entities while EntEmb works only for corpus-level typing of entities. (See Yaghoobzadeh and Schütze (2015) for more differences) It is important to note that MIML can also be applied to any entity typing architecture or model that is trained by distant supervision. Due to the lack of large annotated corpora, distant supervision is currently the only viable approach to fine-grained entity typing; thus, our demonstration of the effectiveness of MIML is an important finding for entity typing.

Joining the results of CNN+MIML-ATT with EntEmb (line 13) gives large improvements over each of the single models. It is also consistently better (by more than 3% in all measures) than our baseline FIGMENT (12), which is basically MLP+EntEmb. This improvement is achieved by using CNN instead of MLP for context representation and integrating MIML-ATT. EntEmb is improved by Yaghoobzadeh et al. (2017) by using entity names. We leave the integration of that model to future work.

Example. To show the behavior of MIML-MAX and MIML-ATT, we extract the scores that each method assigns to the labels for each context. A comparison for the example entity “Richard Bright” (MID: /m/024g5w) who is a PERSON, DOCTOR and AUTHOR is shown in Figure 2. Note

that the weights from MIML-ATT (Eq. 8) sum to 1 for each label because of the applied softmax function while the scores from MIML-MAX (Eq. 1) do not. For both methods, the scores for the type PERSON are more equally distributed than for the other types which makes sense since the entity has the PERSON characteristics in every sentence. For the other types, both models seem to be influenced by other entities occurring in the context (e.g., an occurrence with another DOCTOR could indicate that the entity is also a DOCTOR) but also by trigger words such as “write” or “book” for the type AUTHOR or “disease” for the type DOCTOR.

5.3 Relation Extraction Experiments

Models. In our experiments, we compare two state-of-the-art RE architectures: piecewise CNN (Zeng et al., 2015) and contextwise CNN (Adel et al., 2016). We use the publicly available implementation for the piecewise CNN (URL, 2016a) and our own implementation for the contextwise CNN. Both CNNs represent the input words with embeddings and split the contexts based on the positions of the relation arguments. The contextwise CNN splits the input before convolution, the piecewise CNN after convolution. Also, while the piecewise CNN applies a softmax layer directly after pooling, the contextwise CNN feeds the pooling results into a fully-connected hidden layer first. For both models, we use MI learning to mitigate the noise from distant supervision.

Results. The precision recall (PR) curves in Figure 3 show that the contextwise CNN outperforms the piecewise CNN on our RE dataset. We also compare them to a baseline model that does not learn context features but uses only the embeddings of the relation arguments as an input and feeds them into an MLP (similar to the EntEmb baseline for ET). The results confirm that the context features which the CNNs extract are very important, not only for ET but also for RE. Note that the PR curves are calculated on the corpus level and not on the sentence-level, i.e., after aggregating the predictions for each entity pair. Following Ritter et al. (2013), we compute the area A under the PR curves which supports this trend (EntEmb: $A = 0.34$, piecewise CNN: $A = 0.48$, contextwise CNN: $A = 0.50$).

Pipeline vs. joint training. Since the contextwise CNN outperforms the piecewise CNN, we use the contextwise CNN for integrating en-

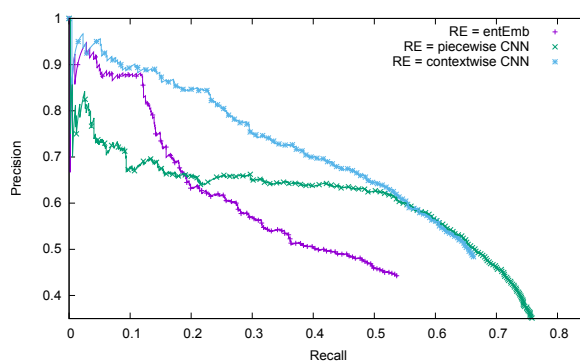


Figure 3: PR curves: relation extraction models

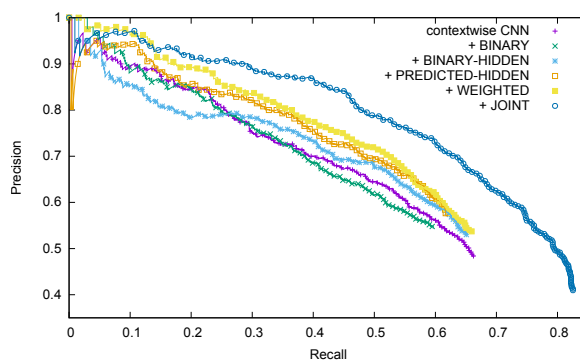


Figure 4: PR curves: type-aware relation extraction models

tity types. Figure 4 shows that the performance on the RE dataset increases when we integrate entity type information into the CNN. The main trend of the PR curves and the areas under them shows the following order of model performances: JOINT-TRAIN $>$ WEIGHTED $>$ PREDICTED-HIDDEN $>$ BINARY-HIDDEN $>$ BINARY.

Discussion. The better performance of our approaches of integrating type predictions into the contextwise CNN (PREDICTED-HIDDEN, WEIGHTED) compared to baseline type integrations (BINARY, BINARY-HIDDEN) shows that probabilistic predictions of an entity typing system can be a valuable resource for RE. With binary types, it is not possible to tell whether one of the selected types had a higher probability than another or whether a type whose binary value is 0 just barely missed the threshold. Probabilistic representations preserve this information. Thus, using probabilistic representations, the RE system can compensate for noise in ET predictions.

WEIGHTED with access to the distributional type embeddings learned from the corpus works better than all other pipeline models. This shows

that our type embeddings can be valuable for RE. JOINT-TRAIN performs better than all pipeline models, even though the ET part in the pipelines is trained on more data. The area of JOINT-TRAIN under the PR curve is $A = 0.66$. A plausible reason is the mutual dependencies of those two tasks which a joint model can better learn than a pipeline model. We can also relate it to better noise mitigation of jointed ET, compared to isolated models.³

Analysis of joint training. In this paragraph, we investigate the joint training in more detail. In particular, we evaluate different variants of it by combining relation extraction with other entity typing approaches: EntEmb and FIGMENT. For joint training with ET-EntEmb, we do not use the context for predicting the types of the relation arguments but only their embeddings. Then, we feed those embeddings into an MLP which computes a representation that we use for the type prediction. This corresponds to the EntEmb model presented in Table 3 (line 11). For joint training with ET-FIGMENT, we compute two different cost functions for entity typing: one for typing based on entity embeddings (see ET-EntEmb above) and one for typing based on an MLP context model. This does not correspond exactly to the FIGMENT model from Table 3 (line 12) which combines an entity embedding and MLP context model as a postprocessing step but comes close. In addition to those two baseline ET models, we also train a version in which both entity typing and relation extraction use EntEmb as their only input features. Figure 5 shows the PR curves for those models. The curve for the model that uses only entity embedding features for both entity typing and relation extraction is much worse than the other curves. This emphasizes the importance of our context model for RE (see also Figure 3), also in combination with joint training. Similarly, the curve for the model with EntEmb as entity typing component has more precision variations than the curves for the other models which use context features for ET. Thus, joint training does not help per se but it is important which models are trained together. The areas under the PR curves show the following model trends: joint with ET-FIGMENT \approx joint as in Figure 1 $>$ joint with ET-EntEmb $>$ joint with ET-EntEmb and RE-EntEmb.

Most improved relations. To identify which

³On the joint dataset, joint training improves MAP for entity typing by about 20% compared to the best isolated model.

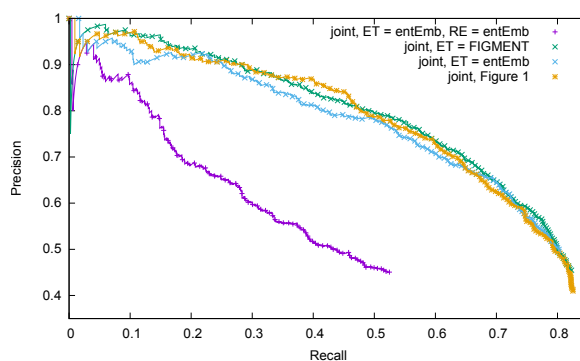


Figure 5: Variants of joint training

relations are improved the most when entity types are integrated, we compare the relation specific F_1 scores of CNN, CNN+WEIGHTED and CNN+JOINT-TRAIN. With WEIGHTED, the relations PPL.deceased_PER.place_of_death and LOC.LOC.containedby are improved the most (from 36.13 to 53.73 and 49.04 to 64.19 F_1 , resp.). JOINT-TRAIN has the most positive impact on PPL.deceased_PER.place_of_death, ORG.ORG.place_founded and GOV.GOV_agency.jurisdiction (from 36.13 to 67.10, 42.38 to 58.51 and 62.26 to 70.41 resp.).

6 Conclusion

In this paper, we addressed different types of noise in two information extraction tasks: entity typing and relation extraction. We presented the first multi-instance multi-label methods for entity typing and showed that it helped to alleviate the noise from distant supervised labels. This is an important contribution because most of the current fine-grained entity typing systems are trained by distant supervision. Our best model sets a new state of the art in corpus-level entity typing. For relation extraction, we mitigated noise from using predicted entity types as features. We compared different pipeline approaches with each other and with our proposed joint type-relation extraction model. We observed that using type probabilities is more robust than binary predictions of types, and joint training gives the best results.

Acknowledgments

This work was supported by DFG (SCHU2246/8-2) and by a Google European Doctoral Fellowship granted to Heike Adel.

References

- Heike Adel, Benjamin Roth, and Hinrich Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838, San Diego, California, June. Association for Computational Linguistics.
- Isabelle Augenstein, Andreas Vlachos, and Diana Maynard. 2015. Extracting relations between non-standard entities using distant supervision and imitation learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 747–757, Lisbon, Portugal, September. Association for Computational Linguistics.
- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Irreflexive and hierarchical relations as translations. In *ICML 2013 Workshop on Structured Learning: Inferring Graphs from Structured and Unstructured Inputs*.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 868–878, Lisbon, Portugal, September. Association for Computational Linguistics.
- 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 Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1243–1249.
- Lan Du, Anish Kumar, Mark Johnson, and Massimiliano Ciaramita. 2015. Using entity information from a knowledge base to improve relation extraction. In *Australasian Language Technology Association Workshop 2015*, pages 31–38.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 363–370, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Evgeniy Gabrilovich, Michael Ringgaard, and Amarag Subramanya. 2013. Facc1: Freebase annotation of cluweb corpora.
- Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Xueyan Jiang, Volker Tresp, Yi Huang, and Maximilian Nickel. 2012. Link prediction in multi-relational graphs using additive models. In *Proceedings of the International Workshop on Semantic Technologies meet Recommender Systems & Big Data, Boston, USA, November 11, 2012*, pages 1–12.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.
- Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany, August. 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*, Toronto, Ontario, Canada., July.
- Yang Liu, Kang Liu, Liheng Xu, and Jun Zhao. 2014. Exploring fine-grained entity type constraints for distantly supervised relation extraction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2107–2116, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at 1st International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, May.
- 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.
- Arvind Neelakantan and Ming-Wei Chang. 2015. Inferring missing entity type instances for knowledge

- base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525, Denver, Colorado, May–June. Association for Computational Linguistics.
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing YAGO: scalable machine learning for linked data. In *World Wide Web Conference*, pages 271–280.
- Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016a. 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.
- Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016b. 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1825–1834.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.
- Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia, June. Association for Computational Linguistics.
- Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. 2013. Modeling missing data in distant supervision for information extraction. *TACL*, 1:367–378.
- 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.
- Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pages 926–934, Lake Tahoe, Nevada, United States., December.
- Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, Jeju Island, Korea, July. Association for Computational Linguistics.
- URL. 2016a. Ds_pcnnns (piecewise cnn) code (kang liu). http://www.nlpr.ia.ac.cn/cip/~liukang/liukangPageFile/code/ds_pcnnns-master.zip.
- URL. 2016b. Fimgent data set. <http://cistern.cis.lmu.de/fimgent>.
- URL. 2016c. Lemur project: Clueweb. <http://lemurproject.org/clueweb12>.
- URL. 2016d. Lemur project: Faccl. <http://lemurproject.org/clueweb12/FACCL>.
- Andreas Vlachos and Stephen Clark. 2014. Application-driven relation extraction with limited distant supervision. In *Proceedings of the AAAI Workshop on Information Discovery in Text, Dublin, Ireland, August 23 2014*, pages 1–6.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, October. Association for Computational Linguistics.
- Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal, September. Association for Computational Linguistics.
- Yadollah Yaghoobzadeh and Hinrich Schütze. 2016. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 236–246, Berlin, Germany, August. Association for Computational Linguistics.
- Yadollah Yaghoobzadeh, , and Hinrich Schütze. 2017. Multi-level representations for fine-grained typing of knowledge base entities. In *EACL*, Valencia, Spain.

- Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023, Cambridge, MA, October. Association for Computational Linguistics.
- Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, AKBC '13*, pages 79–84, San Francisco, California, USA, October.
- 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 Hofmann, 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.
- Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal, September. Association for Computational Linguistics.
- GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan, June. Association for Computational Linguistics.