Learning with Noise: Enhance Distantly Supervised Relation Extraction with Dynamic Transition Matrix

Bingfeng Luo[1], **Yansong Feng**[*1], **Zheng Wang**[2], **Zhanxing Zhu**[3],
Songfang Huang[4], **Rui Yan**[1] and **Dongyan Zhao**[1]
[1]ICST, Peking University, China
[2]School of Computing and Communications, Lancaster University, UK
[3]Peking University, China
[4]IBM China Research Lab, China
{bf_luo,fengyansong,zhanxing.zhu,ruiyan,zhaody}@pku.edu.cn
z.wang@lancaster.ac.uk
huangsf@cn.ibm.com

Abstract

Distant supervision significantly reduces human efforts in building training data for many classification tasks. While promising, this technique often introduces noise to the generated training data, which can severely affect the model performance. In this paper, we take a deep look at the application of distant supervision in relation extraction. We show that the dynamic transition matrix can effectively characterize the noise in the training data built by distant supervision. The transition matrix can be effectively trained using a novel curriculum learning based method without any direct supervision about the noise. We thoroughly evaluate our approach under a wide range of extraction scenarios. Experimental results show that our approach consistently improves the extraction results and outperforms the state-of-the-art in various evaluation scenarios.

1 Introduction

Distant supervision (DS) is rapidly emerging as a viable means for supporting various classification tasks – from relation extraction (Mintz et al., 2009) and sentiment classification (Go et al., 2009) to cross-lingual semantic analysis (Fang and Cohn, 2016). By using knowledge learned from seed examples to label data, DS automatically prepares large scale training data for these tasks.

While promising, DS does not guarantee perfect results and often introduces noise to the generated data. In the context of relation extraction, DS works by considering sentences containing both the subject and object of a <*subj*, rel, *obj*> triple as its supports. However, the generated data are not always perfect. For instance, DS could match the knowledge base (KB) triple, <*Donald Trump*, born-in, *New York*> in *false positive* contexts like *Donald Trump worked in New York City*. Prior works (Takamatsu et al., 2012; Ritter et al., 2013) show that DS often mistakenly labels real positive instances as negative (*false negative*) or versa vice (*false positive*), and there could be confusions among positive labels as well. These noises can severely affect training and lead to poorly-performing models.

Tackling the noisy data problem of DS is non-trivial, since there usually lacks of explicit supervision to capture the noise. Previous works have tried to remove sentences containing unreliable syntactic patterns (Takamatsu et al., 2012), design new models to capture certain types of noise or aggregate multiple predictions under the *at-least-one assumption* that at least one of the aligned sentences supports the triple in KB (Riedel et al., 2010; Surdeanu et al., 2012; Ritter et al., 2013; Min et al., 2013). These approaches represent a substantial leap forward towards making DS more practical. however, are either tightly couple to certain types of noise, or have to rely on manual rules to filter noise, thus unable to scale. Recent breakthrough in neural networks provides a new way to reduce the influence of incorrectly labeled data by aggregating multiple training instances attentively for relation classification, without explicitly characterizing the inherent noise (Lin et al., 2016; Zeng et al., 2015). Although promising, modeling noise within neural network architectures is still in its early stage and much remains to be done.

In this paper, we aim to enhance DS noise modeling by providing the capability to explicitly characterize the noise in the DS-style training data

within neural networks architectures. We show that while noise is inevitable, it is possible to characterize the noise pattern in a unified framework along with its original classification objective. Our key insight is that the DS-style training data typically contain useful clues about the noise pattern. For example, we can infer that since some people work in their birthplaces, DS could wrongly label a training sentence describing a working place as a born-in relation. Our novel approach to noisy modeling is to use a dynamically-generated transition matrix for each training instance to (1) characterize the possibility that the DS labeled relation is confused and (2) indicate its noise pattern. To tackle the challenge of no direct guidance over the noise pattern, we employ a curriculum learning based training method to gradually model the noise pattern over time, and utilize trace regularization to control the behavior of the transition matrix during training. Our approach is flexible – while it does not make any assumptions about the data quality, the algorithm can make effective use of the data-quality prior knowledge to guide the learning procedure when such clues are available.

We apply our method to the relation extraction task and evaluate under various scenarios on two benchmark datasets. Experimental results show that our approach consistently improves both extraction settings, outperforming the state-of-the-art models in different settings.

Our work offers an effective way for tackling the noisy data problem of DS, making DS more practical at scale. Our main contributions are to (1) design a *dynamic* transition matrix structure to characterize the noise introduced by DS, and (2) design a curriculum learning based framework to adaptively guide the training procedure to learn with noise.

2 Problem Definition

The task of distantly supervised relation extraction is to extract knowledge triples, <*subj*, rel, *obj*>, from free text with the training data constructed by aligning existing KB triples with a large corpus. Specifically, given a triple in KB, DS works by first retrieving all the sentences containing both *subj* and *obj* of the triple, and then constructing the training data by considering these sentences as support to the existence of the triple. This task can be conducted in both the sentence and the bag levels. The former takes a sentence s containing both $subj$ and obj as input, and outputs the relation expressed by the sentence between $subj$ and obj. The latter setting alleviates the noisy data problem by using the ***at-least-one assumption*** that at least one of the retrieved sentences containing both $subj$ and obj supports the <*subj*, rel, *obj*> triple. It takes a bag of sentences S as input where each sentence $s \in S$ contains both $subj$ and obj, and outputs the relation between $subj$ and obj expressed by this bag.

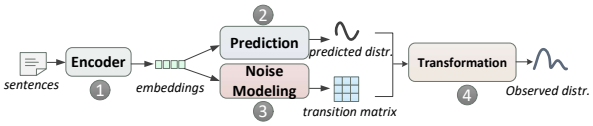


Figure 1: Overview of our approach

3 Our approach

In order to deal with the noisy training data obtained through DS, our approach follows four steps as depicted in Figure 1. First, each input sentence is fed to a sentence encoder to generate an embedding vector. Our model then takes the sentence embeddings as input and produce a predicted relation distribution, $\mathbf{p}$, for the input sentence (or the input sentence bag). At the same time, our model dynamically produces a transition matrix, $\mathbf{T}$, which is used to characterize the noise pattern of sentence (or the bag). Finally, the predicted distribution is multiplied by the transition matrix to produce the observed relation distribution, $\mathbf{o}$, which is used to match the noisy relation labels assigned by DS while the predicted relation distribution $\mathbf{p}$ serves as output of our model during testing. One of the key challenges of our approach is on determining the element values of the transition matrix, which will be described in Section 4.

3.1 Sentence-level Modeling

Sentence Embedding and Prediction In this work, we use a piecewise convolutional neural network (Zeng et al., 2015) for sentence encoding, but other sentence embedding models can also be used. We feed the sentence embedding to a full connection layer, and use *softmax* to generate the predicted relation distribution, $\mathbf{p}$.

Noise Modeling First, each sentence embedding $\mathbf{x}$, generated b sentence encoder, is passed to a full connection layer as a non-linearity to obtain the sentence embedding $\mathbf{x}_n$ used specifically for noise modeling. We then use *softmax* to calculate the

transition matrix $\mathbf{T}$, for each sentence:

$$T_{ij} = \frac{exp(\mathbf{w}_{ij}^T \mathbf{x}_n + b)}{\sum_{j=1}^{|\mathbb{C}|} exp(\mathbf{w}_{ij}^T \mathbf{x}_n + b)} \quad (1)$$

where T_{ij} is the conditional probability for the input sentence to be labeled as relation j by DS, given i as the true relation, b is a scalar bias, $|\mathbb{C}|$ is the number of relations, $\mathbf{w}_{ij}$ is the weight vector characterizing the confusion between i and j.

Here, we dynamically produce a transition matrix, $\mathbf{T}$, specifically for each sentence, but with the parameters ($\mathbf{w}_{ij}$) shared across the dataset. By doing so, we are able to adaptively characterize the noise pattern for each sentence, with a few parameters only. In contrast, one could also produce a global transition matrix for all sentences, with much less computation, where one need not to compute $\mathbf{T}$ on the fly (see Section 6.1).

Observed Distribution When we characterize the noise in a sentence with a transition matrix $\mathbf{T}$, if its true relation is i, we can assume that i might be erroneously labeled as relation j by DS with probability T_{ij}. We can therefore capture the observed relation distribution, $\mathbf{o}$, by multiplying $\mathbf{T}$ and the predicted relation distribution, $\mathbf{p}$:

$$\mathbf{o} = \mathbf{T}^T \cdot \mathbf{p} \quad (2)$$

where $\mathbf{o}$ is then normalized to ensure $\sum_i o_i = 1$.

Rather than using the predicted distribution $\mathbf{p}$ to directly match the relation labeled by DS (Zeng et al., 2015; Lin et al., 2016), here we utilize $\mathbf{o}$ to match the noisy labels during training and still use $\mathbf{p}$ as output during testing, which actually captures the procedure of how the noisy label is produced and thus protects $\mathbf{p}$ from the noise.

3.2 Bag Level Modeling

Bag Embedding and Prediction One of the key challenges for bag level model is how to aggregate the embeddings of individual sentences into the bag level. In this work, we experiment two methods, namely average and attention aggregation (Lin et al., 2016). The former calculates the bag embedding, $\mathbf{s}$, by averaging the embeddings of each sentence, and then feed it to a *softmax* classifier for relation classification.

The attention aggregation calculates an attention value, a_{ij}, for each sentence i in the bag with respect to each relation j, and aggregates to the bag level as $\mathbf{s}_j$, by the following equations[1]:

$$\mathbf{s}_j = \sum_i^n a_{ij}\mathbf{x}_i;\ a_{ij} = \frac{exp(\mathbf{x}_i^T \mathbf{r}_j)}{\sum_{i'}^n exp(\mathbf{x}_{i'}^T \mathbf{r}_j)} \quad (3)$$

where $\mathbf{x}_i$ is the embedding of sentence i, n the number of sentences in the bag, and $\mathbf{r}_j$ is the randomly initialized embedding for relation j. In similar spirit to (Lin et al., 2016), the resulting bag embedding $\mathbf{s}_j$ is fed to a *softmax* classifier to predict the probability of relation j for the given bag.

Noise Modeling Since the transition matrix addresses the transition probability with respect to each true relation, the attention mechanism appears to be a natural fit for calculating the transition matrix in bag level. Similar to attention aggregation above, we calculate the bag embedding with respect to each relation using Equation 3, but with a separate set of relation embeddings $\mathbf{r'}_j$. We then calculate the transition matrix, $\mathbf{T}$, by:

$$T_{ij} = \frac{exp(\mathbf{s}_i^T \mathbf{r'}_j + b_i)}{\sum_{j=1}^{|\mathbb{C}|} exp(\mathbf{s}_i^T \mathbf{r'}_j + b_i)} \quad (4)$$

where $\mathbf{s}_i$ is the bag embedding regarding relation i, and $\mathbf{r'}_j$ is the embedding for relation j.

4 Curriculum Learning based Training

One of the key challenges of this work is on how to train and produce the transition matrix to model the noise in the training data without any direct guidance and human involvement. A straightforward solution is to directly align the observed distribution, $\mathbf{o}$, with respect to the noisy labels by minimizing the sum of the two terms: $CrossEntropy(\mathbf{o}) + Regularization$. However, doing so does not guarantee that the prediction distribution, $\mathbf{p}$, will match the true relation distribution. The problem is at the beginning of the training, we have no prior knowledge about the noise pattern, thus, both $\mathbf{T}$ and $\mathbf{p}$ are less reliable, making the training procedure be likely to trap into some poor local optimum. Therefore, we require a technique to guide our model to gradually adapt to the noisy training data, e.g., learning something simple first, and then trying to deal with noises.

[1] While (Lin et al., 2016) use bilinear function to calculate a_{ij}, we simply use dot product since we find these two functions perform similarly in our experiments.

Fortunately, this is exactly what curriculum learning can do. The idea of curriculum learning (Bengio et al., 2009) is simple: starting with the easiest aspect of a task, and leveling up the difficulty gradually, which fits well to our problem. We thus employ a curriculum learning framework to guide our model to gradually learn how to characterize the noise. Another advantage is to avoid falling into poor local optimum.

With curriculum learning, our approach provides the flexibility to combine prior knowledge of noise, e.g., splitting a dataset into reliable and less reliable subsets, to improve the effectiveness of the transition matrix and better model the noise.

4.1 Trace Regularization

Before proceeding to training details, we first discuss how we characterize the noise level of the data by controlling the trace of its transition matrix. Intuitively, if the noise is small, the transition matrix $\mathbf{T}$ will tend to become an identity matrix, i.e., given a set of annotated training sentences, the observed relations and their true relations are almost identical. Since each row of $\mathbf{T}$ sums to 1, the similarity between the transition matrix and the identity matrix can be represented by its trace, $trace(\mathbf{T})$. The larger the $trace(\mathbf{T})$ is, the larger the diagonal elements are, and the more similar the transition matrix $\mathbf{T}$ is to the identity matrix, indicating a lower level of noise. Therefore, we can characterize the noise pattern by controlling the expected value of $trace(\mathbf{T})$ in the form of regularization. For example, we will expect a larger $trace(\mathbf{T})$ for reliable data, but a smaller $trace(\mathbf{T})$ for less reliable data. Another advantage of employing trace regularization is that it could help reduce the model complexity and avoid overfitting.

4.2 Training

To tackle the challenge of no direct guidance over the noise patterns, we implement a curriculum learning based training method to first train the model without considerations for noise. In other words, we first focus on the loss from the prediction distribution $\mathbf{p}$, and then take the noise modeling into account gradually along the training process, i.e., gradually increasing the importance of the loss from the observed distribution $\mathbf{o}$ while decreasing the importance of $\mathbf{p}$. In this way, the prediction branch is roughly trained before the model managing to characterize the noise, thus avoids being stuck into poor local optimum. We thus design to minimize the following loss function:

$$L = \sum_{i=1}^{N} -((1-\alpha)log(o_{iy_i}) + \alpha log(p_{iy_i})) - \beta trace(\mathbf{T}^i) \quad (5)$$

where $0<\alpha\leq 1$ and $\beta>0$ are two weighting parameters, y_i is the relation assigned by DS for the i-th instance, N the total number of training instances, o_{iy_i} is the probability that the observed relation for the i-th instance is y_i, and p_{iy_i} is the probability to predict relation y_i for the i-th instance.

Initially, we set α=1, and train our model completely by minimizing the loss from the prediction distribution $\mathbf{p}$. That is, we do not expect to model the noise, but focus on the prediction branch at this time. As the training progresses, the prediction branch gradually learns the basic prediction ability. We then decrease α and β by $0<\rho<1$ ($\alpha^*=\rho\alpha$ and $\beta^*=\rho\beta$) every τ epochs, i.e., learning more about the noise from the observed distribution $\mathbf{o}$ and allowing a relatively smaller $trace(\mathbf{T})$ to accommodate more noise. The motivation behind is to put more and more effort on learning the noise pattern as the training proceeds, with the essence of curriculum learning. This gradually learning paradigm significantly distinguishes from prior work on noise modeling for DS seen to date. Moreover, as such a method does not rely on any extra assumptions, it can serve as our default training method for $\mathbf{T}$.

With Prior Knowledge of Data Quality On the other hand, if we happen to have prior knowledge about which part of the training data is more reliable and which is less reliable, we can utilize this knowledge as guidance to design the curriculum. Specifically, we can build a curriculum by first training the prediction branch on the reliable data for several epochs, and then adding the less reliable data to train the full model. In this way, the prediction branch is roughly trained before exposed to more noisy data, thus is less likely to fall into poor local optimum.

Furthermore, we can take better control of the training procedure with trace regularization, e.g., encouraging larger $trace(\mathbf{T})$ for reliable subset and smaller $trace(\mathbf{T})$ for less relaibale ones. Specifically, we propose to minimize:

$$L = \sum_{m=1}^{M}\sum_{i=1}^{N_m} -log(o_{mi,y_{mi}}) - \beta_m trace(\mathbf{T}^{mi}) \quad (6)$$

where β_m is the regularization weight for the m-th data subset, M is the total number of subsets, N_m the number of instances in m-th subset, and $\mathbf{T}^{mi}$, y_{mi} and $o_{mi,y_{mi}}$ are the transition matrix, the relation labeled by DS and the observed probability of this relation for the i-th training instance in the m-th subset, respectively. Note that different from Equation 5, this loss function does not need to initiate training by minimizing the loss regarding the prediction distribution $\mathbf{p}$, since one can easily start by learning from the most reliable split first.

We also use trace regularization for the most reliable subset, since there are still some noise annotations inevitably appearing in this split. Specifically, we expect its $trace(\mathbf{T})$ to be large (using a positive β) so that the elements of $\mathbf{T}$ will be centralized to the diagonal and $\mathbf{T}$ will be more similar to the identity matrix. As for the less reliable subset, we expect the $trace(\mathbf{T})$ to be small (using a negative β) so that the elements of the transition matrix will be diffusive and $\mathbf{T}$ will be less similar to the identity matrix. In other words, the transition matrix is encouraged to characterize the noise.

Note that this loss function only works for sentence level models. For bag level models, since reliable and less reliable sentences are all aggregated into a sentence bag, we can not determine which bag is reliable and which is not. However, bag level models can still build a curriculum by changing the content of a bag, e.g., keeping reliable sentences in the bag first, then gradually adding less reliable ones, and training with Equation 5, which could benefit from the prior knowledge of data quality as well.

5 Evaluation Methodology

Our experiments aim to answer two main questions: (1) is it possible to model the noise in the training data generated through DS, even when there is no prior knowledge to guide us? and (2) whether the prior knowledge of data quality can help our approach better handle the noise.

We apply our approach to both sentence level and bag level extraction models, and evaluate in the situations where we do not have prior knowledge of the data quality as well as where such prior knowledge is available.

5.1 Datasets

We evaluate our approach on two datasets.

TIMERE We build TIMERE by using DS to align time-related Wikidata (Vrandečić and Krötzsch, 2014) KB triples to Wikipedia text. It contains 278,141 sentences with 12 types of relations between an entity mention and a time expression. We choose to use time-related relations because time expressions speak for themselves in terms of reliability. That is, given a KB triple $<e$, rel, $t>$ and its aligned sentences, the finer-grained the time expression t appears in the sentence, the more likely the sentence supports the existence of this triple. For example, a sentence containing both *Alphabet* and *October-2-2015* is very likely to express the inception-time of *Alphabet*, while a sentence containing both *Alphabet* and *2015* could instead talk about many events, e.g., releasing financial report of 2015, hiring a new CEO, etc. Using this heuristics, we can split the dataset into 3 subsets according to different granularities of the time expressions involved, indicating different levels of reliability. Our criteria for determining the reliability are as follows. Instances with full date expressions, i.e., Year-Month-Day, can be seen as the most reliable data, while those with partial date expressions, e.g., Month-Year and Year-Only, are considered as less reliable. Negative data are constructed heuristically that any *entity-time* pairs in a sentence without corresponding triples in Wikidata are treated as negative data. During training, we can access 184,579 negative and 77,777 positive sentences, including 22,214 reliable, 2,094 and 53,469 less reliable ones. The validation set and test set are randomly sampled from the reliable (full-date) data for relatively fair evaluations and contains 2,776, 2,771 positive sentences and 5,143, 5,095 negative sentences, respectively.

ENTITYRE is a widely-used entity relation extraction dataset, built by aligning triples in Freebase to the New York Times (NYT) corpus (Riedel et al., 2010). It contains 52 relations, 136,947 positive and 385,664 negative sentences for training, and 6,444 positive and 166,004 negative sentences for testing. Unlike TIMERE, this dataset does not contain any prior knowledge about the data quality. Since the sentence level annotations in ENTITYRE are too noisy to serve as gold standard, we only evaluate bag-level models on ENTITYRE, a standard practice in previous works (Surdeanu et al., 2012; Zeng et al., 2015; Lin et al., 2016).

5.2 Experimental Setup

Hyper-parameters We use 200 convolution kernels with widow size 3. During training, we use stochastic gradient descend (SGD) with batch size 20. The learning rates for sentence-level and bag-level models are 0.1 and 0.01, respectively.

Sentence level experiments are performed on TIMERE, using 100-d word embeddings pre-trained using GloVe (Pennington et al., 2014) on Wikipedia and Gigaword (Parker et al., 2011), and 20-d vectors for distance embeddings. Each of the three subsets of TIMERE is added after the previous phase has run for 15 epochs. The trace regularization weights are $\beta_1 = 0.01$, $\beta_2 = -0.01$ and $\beta_3 = -0.1$, respectively, from the reliable to the most unreliable, with the ratio of β_3 and β_2 fixed to 10 or 5 when tuning.

Bag level experiments are performed on both TIMERE and ENTITYRE. For TIMERE, we use the same parameters as above. For ENTITYRE, we use 50-d word embeddings pre-trained on the NYT corpus using word2vec (Mikolov et al., 2013), and 5-d vectors for distance embedding. For both datasets, α and β in Eq. 5 are initialized to 1 and 0.1, respectively. We tried various decay rates, {0.95, 0.9, 0.8}, and steps, {3, 5, 8}. We found that using a decay rate of 0.9 with step of 5 gives best performance in most cases.

Evaluation Metric The performance is reported using the precision-recall (PR) curve, which is a standard evaluation metric in relation extraction. Specifically, the extraction results are first ranked decreasingly by their confidence scores, then the precision and recall are calculated by setting the threshold to be the score of each extraction result one by one.

Naming Conventions We evaluate our approach under a wide range of settings for sentence level (sent_) and bag level (bag_) models: (1) _mix: trained on all three subsets of TIMERE mixed together; (2) _reliable: trained using the reliable subset of TIMERE only; (3) _PR: trained with prior knowledge of annotation quality, i.e., starting from the reliable data and then adding the unreliable data; (4) _TM: trained with dynamic transition matrix; (5) _GTM: trained with a global transition matrix. In bag level, we also investigate the performance of average aggregation (_avg) and attention aggregation (_att).

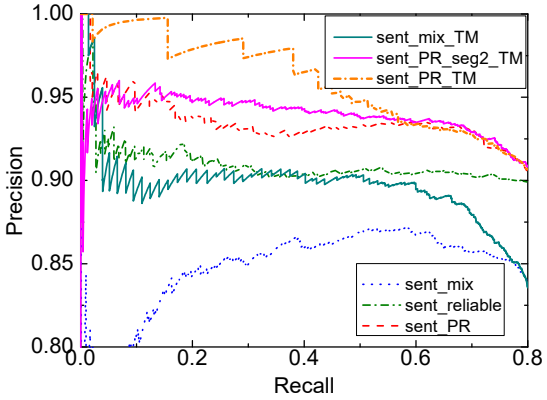


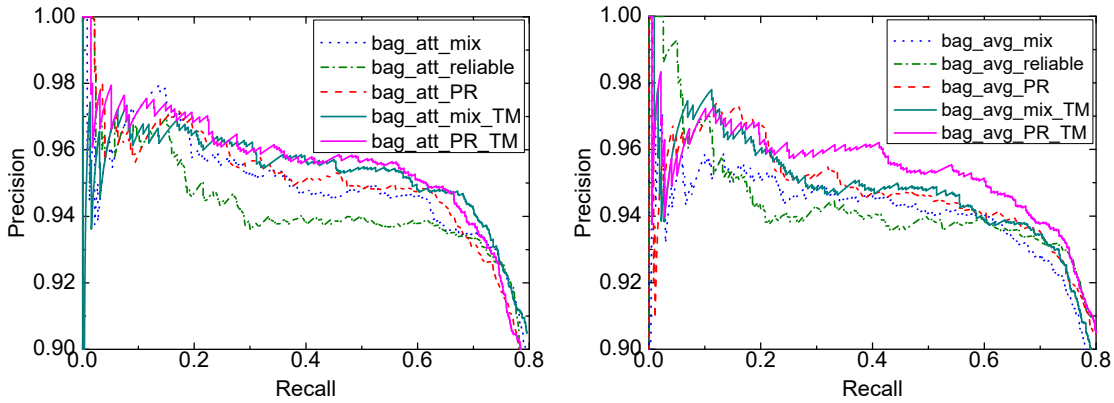
Figure 2: Sentence Level Results on TIMERE

6 Experimental Results

6.1 Performance on TIMERE

Sentence Level Models The results of sentence level models on TIMERE are shown in Figure 2. We can see that mixing all subsets together (sent_mix) gives the worst performance, significantly worse than using the reliable subset only (sent_reliable). This suggests the noisy nature of the training data obtained through DS and properly dealing with the noise is the key for DS for a wider range of applications. When getting help from our dynamic transition matrix, the model (sent_mix_TM) significantly improves sent_mix, delivering the same level of performance as sent_reliable in most cases. This suggests that our transition matrix can help to mitigate the bad influence of noisy training instances.

Now let us consider the PR scenario where one can build a curriculum by first training on the reliable subset, then gradually moving to both reliable and less reliable data. We can see that, this simple curriculum learning based model (sent_PR) further outperforms sent_reliable significantly, indicating that the curriculum learning framework not only reduces the effect of noise, but also helps the model learn from noisy data. When applying the transition matrix approach into this curriculum learning framework using one reliable subset and one unreliable subset generated by mixing our two less reliable subsets, our model (sent_PR_seg2_TM) further improves sent_PR by utilizing the dynamic transition matrix to model the noise. It is not surprising that when we use all three subsets separately, our model (sent_PR_TM) significantly outperforms all other models by a large margin.



(a) Attention Aggregation (b) Average Aggregation

Figure 3: Bag Level Results on TIMERE

Bag Level Models In this setting, we first look at the performance of the bag level models with attention aggregation. The results are shown in Figure 3(a). Consider the comparison between the model trained on the reliable subset only (`bag_att_reliable`) and the one trained on the mixed dataset (`bag_att_mix`). In contrast to the sentence level, `bag_att_mix` outperforms `bag_att_reliable` by a large margin, because `bag_att_mix` has taken the *at-least-one assumption* into consideration through the attention aggregation mechanism (Eq. 3), which can be seen as a denoising step within the bag. This may also be the reason that when we introduce either our dynamic transition matrix (`bag_att_mix_TM`) or the curriculum of using prior knowledge of data quality (`bag_att_PR`) into the bag level models, the improvement regarding `bag_att_mix` is not as significant as in the sentence level.

However, when we apply our dynamic transition matrix into the curriculum built upon prior knowledge of data quality (`bag_att_PR_TM`), the performance gets further improved. This happens especially in the high precision part compared to `bag_att_PR`. We also note that the bag level's *at-least-one assumption* does not always hold, and there are still false negative and false positive problems. Therefore, using our transition matrix approach with or without prior knowledge of data quality, i.e., `bag_att_mix_TM` and `bag_att_PR_TM`, both improve the performance, and `bag_att_PR_TM` performs slightly better.

The results of bag level models with average aggregation are shown in Figure 3(b), where the relative ranking of various settings is similar to those with attention aggregation. A notable difference

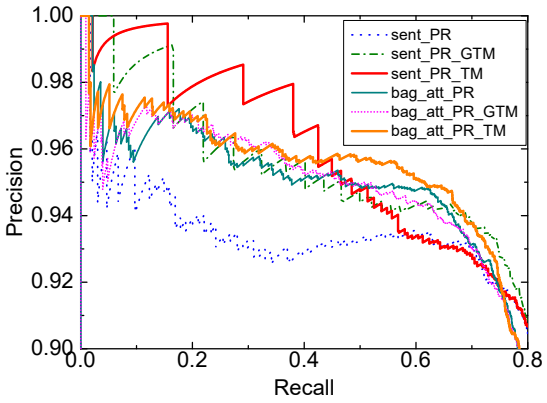


Figure 4: Global TM v.s. Dynamic TM

is that both `bag_avg_PR` and `bag_avg_mix_TM` improve `bag_avg_mix` by a larger margin compared to that in the attention aggregation setting. The reason may be that the average aggregation mechanism is not as good as the attention aggregation in denoising within the bag, which leaves more space for our transition matrix approach or curriculum learning with prior knowledge to improve. Also note that `bag_avg_reliable` performs best in the very-low-recall region but worst in general. This is because that it ranks higher the sentences expressing either `birth-date` or `death-date`, the simplest but the most common relations in the dataset, but fails to learn other relations with limited or noisy training instances, given its relatively simple aggregation strategy.

Global v.s. Dynamic Transition Matrix We also compare our dynamic transition matrix method with the global transition matrix method, which maintains only one transition matrix for all training instances. Specifically, instead of dynam-

ically generating a transition matrix for each datum, we first initialize an identity matrix $\mathbf{T}' \in \mathbb{R}^{|\mathbb{C}|\times|\mathbb{C}|}$, where $|\mathbb{C}|$ is the number of relations (including `no-relation`). Then the global transition matrix $\mathbf{T}$ is built by applying *softmax* to each row of $\mathbf{T}'$ so that $\sum_j \mathbf{T}_{ij} = 1$:

$$T_{ij} = \frac{e^{T'_{ij}}}{\sum_{j=1}^{|\mathbb{C}|} e^{T'_{ij}}} \quad (7)$$

where T_{ij} and T'_{ij} are the elements in the i^{th} row and j^{th} column of $\mathbf{T}$ and $\mathbf{T}'$. The element values of matrix $\mathbf{T}'$ are also updated via backpropagation during training. As shown in Figure 4, using one global transition matrix (`_GTM`) is also beneficial and improves both the sentence level (`sent_PR`) and bag level (`bag_att_PR`) models. However, since the global transition matrix only captures the global noise pattern, it fails to characterize individuals with subtle differences, resulting in a performance drop compared to the dynamic one (`_TM`).

Case Study We find our transition matrix method tends to obtain more significant improvement on noisier relations. For example, *time_of_spacecraft_landing* is noisier than *time_of_spacecraft_launch* since compared to the launching of a spacecraft, there are fewer sentences containing the landing time of a spacecraft that talks directly about the landing. Instead, many of these sentences tend to talk about the activities of the crew. Our `sent_PR_TM` model improves the F1 of *time_of_spacecraft_landing* and *time_of_spacecraft_launch* over `sent_PR` by 9.09% and 2.78%, respectively. The transition matrix makes more significant improvement on *time_of_spacecraft_landing* since there are more noisy sentences for our method to handle, which results in more significant improvement on the quality of the training data.

6.2 Performance on ENTITYRE

We evaluate our bag level models on ENTITYRE. As shown in Figure 5, it is not surprising that the basic model with attention aggregation (`att`) significantly outperforms the average one (`avg`), where `att` in our bag embedding is similar in spirit to (Lin et al., 2016), which has reported the-state-of-the-art performance on ENTITYRE. When injected with our transition matrix approach, both `att_TM` and `avg_TM` clearly outperform their basic versions.

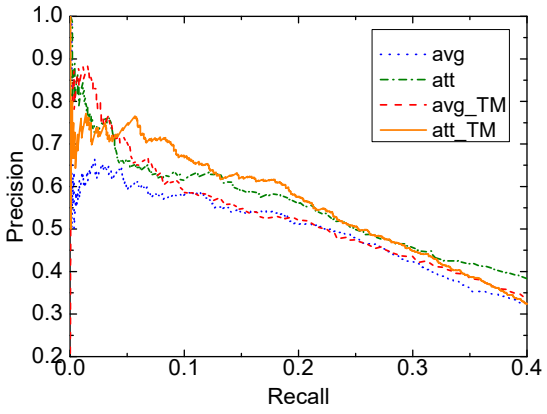


Figure 5: Results on ENTITYRE

Method	P@R_10	P@R_20	P@R_30
Mintz	39.88	28.55	16.81
MultiR	60.94	36.41	-
MIML	60.75	33.82	-
avg	58.04	51.25	42.45
avg_TM	58.56	52.35	43.59
att	61.51	56.36	**45.63**
att_TM	**67.24**	**57.61**	44.90

Table 1: Comparison with feature-based methods. P@R_10/20/30 refers to the precision when recall equals 10%, 20% and 30%.

Similar to the situations in TIMERE, since `att` has taken the *at-least-one assumption* into account through its attention-based bag embedding mechanism, thus the improvement made by `att_TM` is not as large as by `avg_TM`.

We also include the comparison with three feature-based methods: `Mintz` (Mintz et al., 2009) is a multiclass logistic regression model; `MultiR` (Hoffmann et al., 2011) is a probabilistic graphical model that can handle overlapping relations; `MIML` (Surdeanu et al., 2012) is also a probabilistic graphical model but operates in the multi-instance multi-label paradigm. As shown in Table 1, although traditional feature-based methods have reasonable results in the low recall region, their performances drop quickly as the recall goes up, and `MultiR` and `MIML` did not even reach the 30% recall. This indicates that, while human-designed featurs can effectively capture certain relation patterns, their coverage is relatively low. On the other hand, neural network models have more stable performance across different recalls, and `att_TM` performs generally better than other models, indicating again the effectiveness of our transition matrix method.

7 Related Work

In addition to relation extraction, distant supervision (DS) is shown to be effective in generating training data for various NLP tasks, e.g., tweet sentiment classification (Go et al., 2009), tweet named entity classifying (Ritter et al., 2011), etc. However, these early applications of DS do not well address the issue of data noise.

In relation extraction (RE), recent works have been proposed to reduce the influence of wrongly labeled data. The work presented by (Takamatsu et al., 2012) removes potential noisy sentences by identifying bad syntactic patterns at the preprocessing stage. (Xu et al., 2013) use pseudo-relevance feedback to find possible false negative data. (Riedel et al., 2010) make the *at-least-one assumption* and propose to alleviate the noise problem by considering RE as a multi-instance classification problem. Following this assumption, people further improves the original paradigm using probabilistic graphic models (Hoffmann et al., 2011; Surdeanu et al., 2012), and neural network methods (Zeng et al., 2015). Recently, (Lin et al., 2016) propose to use attention mechanism to reduce the noise within a sentence bag. Instead of characterizing the noise, these approaches only aim to alleviate the effect of noise.

The *at-least-one assumption* is often too strong in practice, and there are still chances that the sentence bag may be false positive or false negative. Thus it is important to model the noise pattern to guide the learning procedure. (Ritter et al., 2013) and (Min et al., 2013) try to employ a set of latent variables to represent the true relation. Our approach differs from them in two aspects. We target noise modeling in neutral networks while they target probabilistic graphic models. We further advance their models by providing the capability to model the fine-grained transition from the true relation to the observed, and the flexibility to combine indirect guidance.

Outside of NLP, various methods have been proposed in computer vision to model the data noise using neural networks. (Sukhbaatar et al., 2015) utilize a global transition matrix with weight decay to transform the true label distribution to the observed. (Reed et al., 2014) use a hidden layer to represent the true label distribution but try to force it to predict both the noisy label and the input. (Chen and Gupta, 2015; Xiao et al., 2015) first estimate the transition matrix on a clean dataset and apply to the noisy data. Our model shares similar spirit with (Misra et al., 2016) in that we all dynamically generate a transition matrix for each training instance, but, instead of using vanilla SGD, we train our model with a novel curriculum learning training framework with trace regularization to control the behavior of transition matrix. In NLP, the only work in neural-network-based noise modeling is to use one single global transition matrix to model the noise introduced by cross-lingual projection of training data (Fang and Cohn, 2016). Our work advances them through generating a transition matrix dynamically for each instance, to avoid using one single component to characterize both reliable and unreliable data.

8 Conclusions

In this paper, we investigate the noise problem inherent in the DS-style training data. We argue that the data speak for themselves by providing useful clues to reveal their noise patterns. We thus propose a novel transition matrix based method to dynamically characterize the noise underlying such training data in a unified framework along the original prediction objective. One of our key innovations is to exploit a curriculum learning based training method to gradually learn to model the underlying noise pattern without direct guidance, and to provide the flexibility to exploit any prior knowledge of the data quality to further improve the effectiveness of the transition matrix. We evaluate our approach in two learning settings of the distantly supervised relation extraction. The experimental results show that the proposed method can better characterize the underlying noise and consistently outperform start-of-the-art extraction models under various scenarios.

Acknowledgement

This work is supported by the National High Technology R&D Program of China (2015AA015403); the National Natural Science Foundation of China (61672057, 61672058); KLSTSPI Key Lab. of Intelligent Press Media Technology; the UK Engineering and Physical Sciences Research Council under grants EP/M01567X/1 (SANDeRs) and EP/M015793/1 (DIVIDEND); and the Royal Society International Collaboration Grant (IE161012).

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*. ACM, pages 41–48.

Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *ICCV*. pages 1431–1439.

Meng Fang and Trevor Cohn. 2016. Learning when to trust distant supervision: An application to low-resource pos tagging using cross-lingual projection. In *CONLL*. pages 178–186.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1(12).

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 ACL*. pages 541–550.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*. volume 1, pages 2124–2133.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. pages 3111–3119.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*. pages 777–782.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*. pages 1003–1011.

Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. 2016. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *CVPR*. pages 2930–2939.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. Technical report, Linguistic Data Consortium, Philadelphia.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* .

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pages 148–163.

Alan Ritter, Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP*. Association for Computational Linguistics, pages 1524–1534.

Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. 2013. Modeling missing data in distant supervision for information extraction. *TACL* 1:367–378.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training convolutional networks with noisy labels. In *ICLR*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP-CoNLL*. pages 455–465.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *ACL*. pages 721–729.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM* 57(10):78–85.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *CVPR*. pages 2691–2699.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *ACL*. pages 665–670.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*. pages 1753–1762.