Learning with Inevitable Noise: Enhance Distantly Supervised Relation Extraction with Dynamic Transition Matrices

Anonymous ACL submission

Abstract

Distant supervision significantly reduces human efforts in constructing training data for classification tasks by automatically labeling data using knowledge learned from seed examples. While promising, this technique often introduces noise into the dataset, which can severely affect the quality of the learned model. In this paper, we take a deep look at the application of distant supervision in relation extraction. We show that dynamic transition matrices can effectively model the noise in the training data generated through distant supervision. We propose to use curriculum learning to characterize the noise patterns, and control the behavior of noise to further improve the model robustness. We apply our approach to relation extraction at both the sentence and the bag levels. The experimental results show that our method can better handle the noise over the state of the art in distantly supervised relation extraction and improve extraction performance in various settings. TODO: I do not like the last sentence!! btw, shall we consider TIMERE as a contribution?

1 Introduction

In recent years, distant supervision (DS) is emerging as a viable means for supporting various classification tasks – from relation extraction (**?**) and sentiment classification (**?**) to cross-lingual semantic analysis (**?**). By automatically aligning facts from knowledge bases to text, this technique allows us to scale a small number of seeds to millions of instances.

While promising, DS does not guarantee perfect results and often introduces noise to the generated training data. In the context of relation extraction, DS could match the knowledge tuple <"Donald Trump", *born-in*, "New York"> in *false positive* contexts like "*Donald Trump worked in New York City*". Many prior works () show that DS often mistakenly labels real positive instances as negative (*false negative*) or versa vice (*false positive*), and there could also have confusions among positive labels too. These noises can severely affect the training procedure and lead to poorly-performing models.

There have been attempts to tackle the noisy data problem of DS. Previous works have tried to preprocess the data by removing sentences containing unreliable syntactic patterns (**?**), making the ***at-least-one*** assumption that at least one of the aligned sentences support the knowledge triple to reduce the influence of noise (**?**) and adding a set of variables in probabilistic graphic models to represent the true relation before????? the noisy relation labeled by distant supervision (**??**). These works represent a substantial leap forward towards making DS more practical. However, these methods either do not model the noise explicitly or just handle false positive or false negative noise. ——F: This argument is weak! can we say they either rely on predefined rules or extra labeled dataset?

In this paper, we show that while noise is inevitable, it is possible to model its pattern in a unified framework Our key insight is that the input data would typically contain useful clues about the noise pattern. For example, if we see an input sentence describing the work place of a person, we can reasonably assume that it has some chances to be erroneously labeled by distant supervision to express relation *born_in* (—F: can we use *live_place*??) . Accordingly, we propose to dynamically generate a transition matrix for each input datum to model its noise pattern. We ap-

ply our method to the relation extraction task. To tackle the difficulty of lack of direct guidance over the noise pattern, we employ curriculum learning to gradually model the noise pattern over time. To prevent TODO: xx, we use trace regularization to control the behavior of the transition matrix during training. We show that our novel technique can better model the noise pattern over the start-of-the-art, leading to improved learning performance.

We evaluate our approach by applying it to xxx?? and compare it against a state-of-the-art xx. Experimental results show that our approach can create a higher-quality relation extraction classifier with xx% better performance using the same set of training data. (—-F: Maybe we should change another type of representation? or list our contributions here?)

This paper makes the following specific contributions. It is the first to

- bla
- bla
- bla

2 Problem Definition

The task of distantly supervised relation extraction is to extract knowledge tuples, i.e.,<*subj*,*rel*,*obj*>, from free text using existing knowledge base, which can be applied 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 is based on the *at-least-one* assumption and takes a bag of sentences S as input where each sentence $s \in S$ contains both $subj$ and obj, and output the relation between $subj$ and obj expressed by this bag.

This work aims to improve the state of the art in DS for relation extraction. We do so by first modeling the noise in the DS generated training data using dynamic transition matrix, at both the sentence and the bag levels (Section 3), and then introducing a curriculum learning framework to control the behavior of noise under various training situations (Section **??**).

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

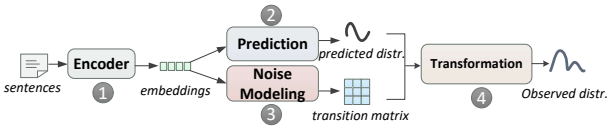


Figure 1: Overview of our approach

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 level of the DS-labeled input sentence (or the bag). Finally, the predicted distribution is multiplied by the transition matrix to produce the observed relation distribution, $\mathbf{o}$. This observed distribution $\mathbf{o}$ is used to capture the (possibly noisy) relation labels assigned by DS and update our model parameters during training, while the predicted relation distribution $\mathbf{p}$ is the output of our model during testing. One of the key challenges of our approach is on determining the weights of the transition matrix, $\mathbf{T}$, which will be described in Section **??**.

3.1 Sentence-level Modeling

Sentence Embedding and Prediction In this work, we use a piecewise convolutional neural network (**?**) for sentence encoding, but other sentence embedding models can also be used. To generate the predicted relation distribution, $\mathbf{p}$, we first feed the sentence embedding to a full connection layer, and then use *softmax* for relation classification. Again, other probabilistic prediction methods can also be used here.

Noise Modeling We use a fully connected layer in the neural network to perform noise modeling at the sentence level. Each sentence embedding, $\mathbf{x}$, generated by the encoder is passed to the full connection layer as a non-linearity to obtain the sentence embedding $\mathbf{x}_n$ used specifically for the noise modeling branch. We then use a *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 given sentence to be labeled as relation j by DS, but with i as the true relation, b a scalar bias, $|\mathbb{C}|$

the number of relations, $\mathbf{w}_{ij}$ is a shared weight vector to characterize the confusion between i and j.

—-F: Dynamic v.s. Static: 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 level for each sentence, with a few parameters only. In contrast, one could also produce one single transition matrix for all sentences, with much less computation, where one need not to compute $\mathbf{T}$ on the air.

Observed Distribution When we model the noise with a transition matrix, if the true relation of an input training instance is i, we can assume that this relation i could be labeled as relation j by DS with probability T_{ij}. Therefore, we can model the observed relation distribution, $\mathbf{o}$, by multiplying transition matrix, $\mathbf{T}$, and the predicted relation distribution, $\mathbf{p}$:

$$\mathbf{o} = \mathbf{T}^T \cdot \mathbf{p} \tag{2}$$

where $\cdot$ represents dot product, and we normalize the elements of $\mathbf{o}$ to ensure $\sum_i o_i = 1$.

Different from previous works that use the predicted relation distribution $\mathbf{p}$ to directly match the relation labeled by DS (**??**)??. We instead use $\mathbf{o}$ to match the noisy label during training and still use $\mathbf{p}$ as output during testing. By doing so, we can see that Equation **??** actually models 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 of the bag. In this work, we experiment two methods, namely average and attention aggregation (**?**). 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]:

$$\mathbf{s}_j = \sum_i^n a_{ij}\mathbf{x}_i \tag{3}$$

$$a_{ij} = \frac{exp(\mathbf{x}_i^T \mathbf{r}_j)}{\sum_i^n exp(\mathbf{x}_i^T \mathbf{r}_j)} \tag{4}$$

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 of relation j. Following (**?**), the resultant bag embedding $\mathbf{s}_j$ is fed to a *softmax* classifier to predict the probability of relation j. this is a binary classifier, right??? no, multi-class classifier

Noise Modeling Since the transition matrix models the transition distribution 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, we concentrate on each relation one by one and calculate its transition distribution respectively. Specifically, we calculate the bag embedding with respect to each relation using Equation **??** and **??** but with another set of relation embeddings $\mathbf{r}_t^j$. We then calculate the transition matrix, $\mathbf{T}$, by:

$$T_{ij} = \frac{exp(\mathbf{s}_i^T \mathbf{r}_t^j + b)}{\sum_{j=1}^{|\mathbb{C}|} exp(\mathbf{s}_i^T \mathbf{r}_t^j + b)} \tag{5}$$

where $\mathbf{s}_i$ is the bag embedding with respect to relation i, and $\mathbf{r}_t^j$ is the same embedding of relation j. do we need to clarify that avg and att share the same noise modeling method?

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 without any direct guidance and human involvement in the training data. A straightforward solution is to directly align the observed distribution, $\mathbf{o}$, with respect to the noisy annotations 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

[1] While (**?**) use bilinear function to calculate a_{ij} in their paper, we simply use dot product since we find these two functions perform similarly.

pattern; thus, both $\mathbf{T}$ and $\mathbf{p}$ are less reliable, which will make the training procedure be likely to trap into some poor local optimum. Therefore, what we would like to have is 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.

Fortunately, this is exactly what curriculum learning can do. The idea of curriculum learning () 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 advantange is to avoid falling into poor local optimum, when there is little knowledge about the noise pattern.

By exploring curriculum learning, our approach provides the flexibility to combine any prior knowledge of noise to improve the effectiveness of the transition matrix. We show that if one could break the dataset into reliable and less reliable parts, our approach can exploit the split as indirect supervision over the noise pattern to build more effective transition matrix to better model the noise. The sum is greater than its parts. As shown later in Section **??**, putting together these techniques enables us to build an adaptive scheme to better model the noise pattern over the state-of-the-art.

4.1 Trace Regularization

Before proceeding to training details, we first discuss how we characterize the noise level of the data by controling the trace of the 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 the trace of $\mathbf{T}$, $trace(\mathbf{T})$. 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 level of the data 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 be helpful to 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 local optimum. We accordingly design to minimize the following loss function:

$$Loss = \sum_{i=1}^{N} -((1-\alpha)log(o_{iy_i}) + \alpha log(p_{iy_i})) - \beta trace(\mathbf{T}^i) \quad (6)$$

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

Initially, we set $\alpha = 1$, 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 focusing 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 during training, 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 an indirect guidance over the noise patterns by helping the model to distinguish reliable data from less reliable ones (DO WE actually distinguish them??)yes, with trace reg, and indirectly with CL as well. 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 by using 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 the following loss function:

$$Loss = \sum_{m=1}^{M}\sum_{i=1}^{N_m} -log(o_{miy_{mi}}) - \beta_m trace(\mathbf{T}^{mi}) \quad (7)$$

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

For the reliable subset, we want $trace(\mathbf{T})$ to be large (positive β) so that the element values of $\mathbf{T}$ will be centralized to the diagonal and $trace(\mathbf{T})$ will be more similar to the identity matrix. As for the less reliable subset, we want the $trace(\mathbf{T})$ to be small (negative β) so that the element values of their transition matrices will be diffusive and the transition matrix will be less similar to the identity matrix. In other words, the transition matrix is encouraged to characterize the noise.

It is to note that this loss function only works for sentence level models. Since reliable sentences and less reliable ones are all aggregated into a sentence bag in the bag level models, we can not determine which bag is reliable and which is not. However, bag level models can still use the curriculum by changing the content of the bag, e.g., keeping reliable sentences in the bag first and gradually adding less reliable ones, and use Equation **??** for training. By doing so, it can 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 prior knowledge of the data quality is available as well as the situation where such prior knowledge is missing.

5.1 Datasets

We evaluate our approach on two datasets. The first one is TIMERE, constructed by ourself using DS. It is used to evaluate our approach on scenarios with and without prior knowledge about the data quality. The second dataset is the widely used ENTITYRE dataset (**?**), which provides no prior knowledge of the data quality.

TIMERE is automatically constructed by using DS to align time-related Wikidata(**?**) knowledge triples to Wikipedia text, containing 278,141 sentences with 12 types of relations between an entity mention and a time expression. We choose time-related relations in this dataset as time expressions speak for themselves in terms of reliability. That is, given a knowledge 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 highly 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 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 of the data 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

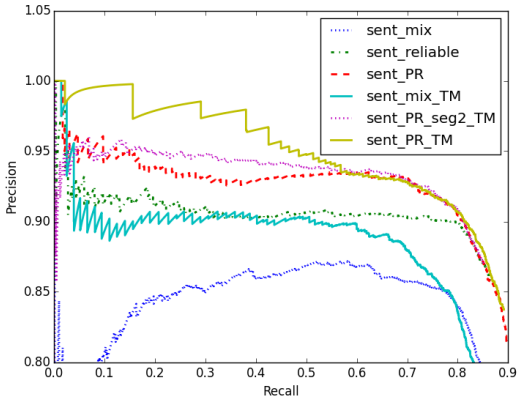


Figure 2: Sentence Level Results on TimeRE

`Year-Only`, can be seen as less reliable. Negative data are constructed heuristically that any *entity-time* pairs in a sentence without corresponding triples in Wikidata are considered as negative data. During training, we can access 184,579 negative instances and 77,777 positive instances, including 22,214 reliable ones, 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 instances and 5,143, 5,095 negative instances, respectively.

ENTITYRE is a widely-used entity relation extraction dataset, built by aligning triples in Freebase to the New York Times (NYT) corpus (**?**). 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. However, it is a good example to evaluate the generalization ability of our transition matrix approach. *Since the sentence level annotations in the test set of* ENTITYRE *are too noisy to serve as gold standard, we only evaluate bag-level models on* ENTITYRE*– a standard practice in previous works* (**???**).

5.2 Hyperparameters

Neural network components used in our algorithm are based on a convolution neural network with 200 convolution kernels and the window size is 3. To train the network, we use stochastic gradient descend (SGD) with a batch size of 20. The learning rates for sentence-level and bag-level models are 0.1 and 0.01, respectively.

Sentence level modeling is performed on the TIMERE dataset, where we use 100-dimensional word embeddings pre-trained using GloVe (**?**) on Wikipedia and Gigaword[2], and 20-dimensional 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 for the three subsets are $\beta_1 = 0.01$, $\beta_2 = -0.01$ and $\beta_3 = -0.1$, respectively, from the reliable to the most unreliable part, and the ratio of β_3 and β_2 is fixed to 10 or 5 during hyperparameter tuning.

Bag level modeling is performed on both TIMERE and ENTITYRE datasets. We use the same parameters as sentence level for TIMERE. For ENTITYRE, we use 50-dimensional word embeddings pre-trained on the NYT corpus using word2vec[3]. The size for distance embedding is 5. For both datasets, the linear combination parameter α is 1 and the trace regularization parameter β is initialized to 0.1. We tried various decay rates, $\{0.95, 0.9, 0.8\}$, and decay steps, $\{3, 5, 8\}$. We found that using a decay rate of 0.9 and step of 5 gives best performance in most cases.

5.3 Evaluation Metric

The performance is reported using the precision-recall (`PR`) curve, calculated according to the extraction results ranked decreasingly by their confidence scores. This is a standard evaluation metric in relation extraction.

5.4 Naming Conventions

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

For bag level modeling, we also evaluate the model performance using average aggregation (`_avg`) and attention aggregation (`_att`), as described in Section **??**.

[2] `catalog.ldc.upenn.edu/LDC2011T07`
[3] `code.google.com/p/word2vec/`

6 Experimental Results

6.1 Performances on TIMERE

Sentence Level Models The results of sentence level models on TIMERE are shown in Figure **??**. We can see that mixing all subsets together (sent_mix) gives the worst result. The performance of this strategy is 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 to be adopted at a wider scale. When getting help from our transition matrix during training, 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.

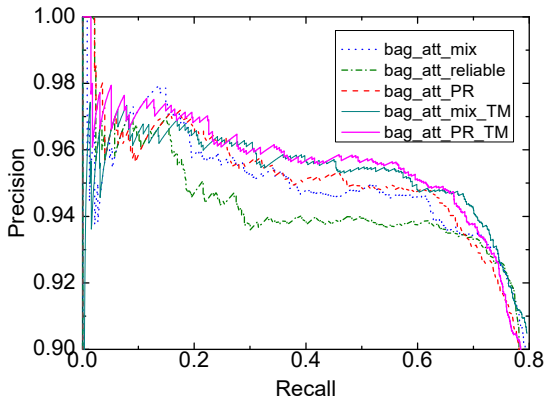
Let us consider the PR scenario where one can build a curriculum by first training on the reliable subset and then gradually moving to the space with both reliable and less reliable subsets. In this case, the curriculum learning based model (sent_PR) even outperforms *sent_reliable* significantly (F: does sent_PR have help from other components? e.g., TM?) No, indicating that the curriculum learning framework not only reduces the effect of noise, but also helps the model learns 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 the two less reliable subsets, our model (*sent_PR_seg2_TM*) further improves *sent_PR* by enabling it to use transition matrix to model the noise (I further rephrased this sentence.) 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[4]. TODO: ZW: I rephrase some of the sentences, but I still think this paragraph needs to be written.

Bag Level Models In this experiment, we first look at the performance of the bag level models with attention aggregation. The results are shown in Figure **??**. Consider a comparison between the model trained on the reliable subset only (bag_att_reliable) and the model trained on the mixed dataset (bag_att_mix). In contrast to the sentence level cases, bag_att_mix outperforms bag_att_reliable by a large margin. This is due to the fact that bag_att_mix has taken the noise within the bag into consideration through the attention aggregation mechanism, which can be seen as a denoising step within the bag. This may also be the reason that when we introduce either our transition matrix approach (bag_att_mix_TM) or the curriculum of using the reliable data first (bag_att_PR) into the bad level model, the improvement compared to bag_att_mix is not as significant as in the sentence level. However, when we utilize our transition matrix approach to model the noise with the curriculum of using the reliable data first (bag_att_PR_TM), the performance gets further improved. This is 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, we can see that 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.

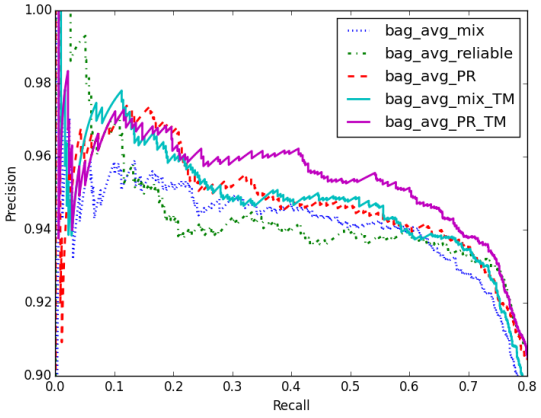
The results of the bag level models with average aggregation is shown in Figure **??**. The relative ranking of various settings is similar to those with the attention aggregation. One of the notable differences is that both bag_avg_PR and bag_avg_mix_TM improve bag_avg_mix with larger margins compared to that in the attention aggregation setting. The reasons may be that the average aggregation mechanism is not as good as the attention aggregation one in terms of denoising ability, which leaves more space for our transition matrix approach or curriculum learning with prior knolwdege of data quality to improve. We can also see that bag_avg_reliable performs best in the very-low-recall region but worst in general. This is because that it ranks highly the sentences expressing relation *birth-date* and *death-date* which are the simplest and the most common relations in the dataset. However, due to the less amout of data, this model performs worse in other relations and thus generates the worst results in general.

Global v.s. Dynamic Transition Matrix We also compare our dynamic transition matrix method with the global transition matrix method.

[4]We will use all three subsets for all *_PR* settings in the rest of the experiments.



(a) Attention Aggregation



(b) Average Aggregation

Figure 3: Bag Level Results on TimeRE

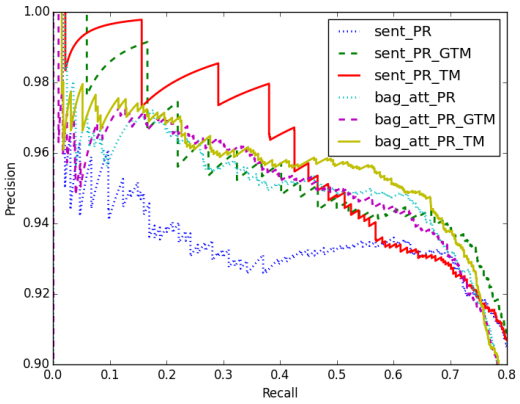


Figure 4: Single TM and Dynamic TM

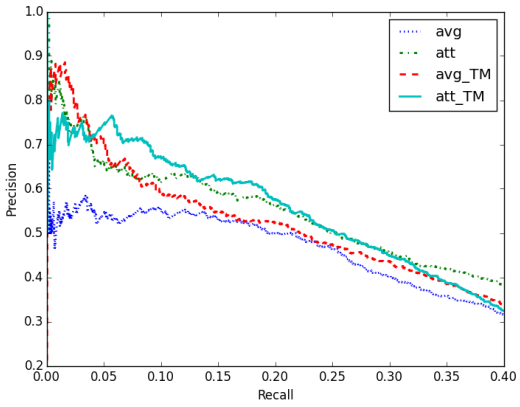


Figure 5: Results on Dataset of Riedel et.al.

Specifically, instead of dynamically generating a transition matrix for each datum, the global transition matrix method use a single transition matrix for all the traning data. First, we initialize an identity matrix $\mathbf{T}' \in \mathbb{R}^{k \times k}$ where k is the number of relations (including *no_relation*, (unify the notation)), then the global transition matrix is calculated with row-wise softmax:

$$T_{ij} = \frac{e^{T'_{ij}}}{\sum_{j=1}^{k} e^{T'_{ij}}} \tag{8}$$

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. The results are shown in Figure **??**. We can see that using only a global transition matrix (`_GTM`) is also beneficial and it improves both `sent_PR` and `bag_att_PR`. However, since the global transition matrix only captures global noise pattern and will therefore have incorrect behavior when the data is reliable, it performs worse than our dynamic transition matrix methods (`_TM`).

6.2 Performance on EntityRE

We also conduct experiments on the EntityRE dataset, where we can evaluate our bag level models only. TODO: ZW: Why bag-level models only? We need a an answer! F: I put a sentence in data preparation. We implemented the average aggregation method (`avg`), and the attention aggregation method (`att`) proposed by (**?**) as well as their corresponding transition matrix versions (`avg_TM` and `att_TM`). As we can see in Figure **??**, due to the inferior denoising ability of the average aggregation component, `avg` performs worst among all those models. what should we say about this figure???? When we inject our transition matrix approach into both *avg* and *att*, the resulting two models, both of which can clearly outperform

their basic extraction models. Again, because the attention aggregation model already has good ability in reducing the impact of sentence level noise and the bag level noise is less significant than the sentence level noise, the improvement of our transition matrix model is limited, which only improves the model on the low recall part.

6.3 Summary

TODO: ZW: This section is incredibly complex. I suggest to have a summary section to highlight the take away messages.

7 Related Work

Distant Supervision in NLP 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 (**?**), tweet named entity classifying (**?**). However, these early applications of DS do not address the issue of data noise. no cross lingual POS tagging?

Noise Reduction Some approaches have been proposed in recent years to reduce the influence of wrongly labeled data in relation extraction (RE). The work presented by (**?**) removes potential noisy sentences by identifying bad syntactic patterns at the pre-processing stage. (**?**) uses pseudo-relevance feedback to find possible false negative data. (**?**) proposes to alleviate the noise problem by considering the RE task as a multi-instance classification problem and assume that at least one of the retrieved sentences containing both *subj* and *obj* supports the corresponding <*subj*,*rel*,*obj*> triple (i.e. *at-least-one* assumption). Following this assumption, people further improves the original method using prababilistic graphic model mehtods (**??**), and neural network methods (**?**). Recently (**?**) further proposes to use attention mechanism to reduce the noise inside the sentence bag. In contrast to our work in noise modeling, these approaches all aim to identify either reliable or unreliable instances to reduce the bad influence of noise.

Noise Modeling The *at-least-one* assumption mentioned above 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. Recent works in the area include (**?**) and (**?**) that employ a set of latent variables to represent the true relation. Our approach differs from the these two approaches in two aspects. First, we target the neutral network framework while they target the probabilistic graphic model framework. Second, we advance their models by providing the capability to model the fine-grained transition from the true to the observed relation.

Recently, various approaches have been proposed in the computer vision domain for modeling the data noise using neural networks. (**?**) proposes to use a global transition matrix to transform the true label distribution to the observed label distribution. They also propose the idea of trace regularization but use weight decay for simplicity. (**?**) also uses a hidden layer to represent the true label distribution but try to force it to predict both the noisy label and the input. (**??**) first estimates the transition matrix on the clean data set and use it in the noisy data set. Our model shares similar spirit with (**?**) in that we all dynamically generate a transition matrix for each training data item. However, instead of using Expectation-Maximization (EM), we train our model with a curriculum learning framework and trace regularization. In the field of NLP, there is currently little work in neural-network-based nosie modeling. A recent work is conduct by (**?**), which uses a global transition matrix to model the noise introduced by cross-lingual projection of training data. Our work advances (**?**) through generating a transition matrix for each datum dynamically, which avoids the drawback of global transition matrix that it use a single transition matrix for both reliable and unreliable data.

8 Conclusion

TODO: re-write In this paper, we propose that the input data may contain useful clues that indicate the noise pattern introduced by distant supervision. We therefore propose to model the noise by dynamically generating a transition matrix for each training data. To overcome the lack direct guidance over the noise pattern, we propose to use curriculum learning to model the noise gradually and trace regularization to help control the behavior of the transition matrix. The experiments on two datasets show that our transition matrix method improves both the sentence level and bag level model in the situation with and without prior knowledge of data quality. We also find that the

prior knowledge of data quality can help the training of the transition matrix.

References

Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 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. *arXiv preprint arXiv:1607.01133* .

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*.

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

Bingfeng Luo, Yansong Feng, and Dongyan Zhao. 2016. Improving first order temporal fact extraction with unreliable data. In *Natural Language Processing and Chinese Computing*. Springer.

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 *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*.

Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. 2016. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels .

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, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1524–1534.

Alan Ritter, Luke Zettlemoyer, Oren Etzioni, et al. 2013. Modeling missing data in distant supervision for information extraction. *Transactions of the Association for Computational Linguistics* 1:367–378.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080* .

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*. Association for Computational Linguistics, pages 455–465.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 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 (2)*. pages 665–670.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. EMNLP.