Procesamiento del Lenguaje Natural, Revista nº 68, marzo de 2022, pp. 71-83 recibido 19-12-2021 revisado 07-02-2022 aceptado 10-02-2022

Risks of misinterpretation in the evaluation of Distant Supervision for Relation Extraction

Riesgos de interpretación errónea en la evaluación de la Supervisión Distante para la Extracción de Relaciones

Juan-Luis García-Mendoza[1], **Luis Villaseñor-Pineda**[1], **Felipe Orihuela-Espina**[1,2]
[1]Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, Mexico
{juanluis,villasen,f.orihuela-espina}@inaoep.mx
[2]University of Birmigham, Birmingham, United Kingdom

Abstract: Distant Supervision is frequently used for addressing Relation Extraction. The evaluation of Distant Supervision in Relation Extraction has been attempted through Precision-Recall curves and/or calculation of Precision at N elements. However, such evaluation is challenging because the labeling of the instances results from an automatic process that can introduce noise into the labels. Consequently, the labels are not necessarily correct, affecting the learning process and the interpretation of the evaluation results. Therefore, this research aims to show that the performance of the methods measured with the mentioned evaluation strategies varies significantly if the correct labels are used during the evaluation. Besides, based on the preceding, the current interpretation of the results of these measures is questioned. To this end, we manually labeled a subset of a well-known data set and evaluated the performance of 6 traditional Distant Supervision approaches. We demonstrate quantitative differences in the evaluation scores when considering manually versus automatically labeled subsets. Consequently, the ranking of performance among distant supervision methods is different with both labeled.

Keywords: Relation Extraction. Distant Supervision evaluation. Precision-Recall curves. Precision at N.

Resumen: La Supervisión Distante se utiliza con frecuencia para abordar la extracción de relaciones. La evaluación de la Supervisión Distante en la Extracción de Relaciones se ha realizado mediante curvas de Precisión-Cobertura y/o el cálculo de la Precisión en N elementos. Sin embargo, dicha evaluación es un desafío porque el etiquetado de las instancias es el resultado de un proceso automático. En consecuencia, las etiquetas no son necesariamente correctas, afectando no solo el proceso de aprendizaje sino también la interpretación de los resultados de la evaluación. El objetivo de esta investigación es mostrar que el desempeño de los métodos medido con las estrategias de evaluación mencionadas varía de manera significativa si se utilizan las etiquetas correctas durante la evaluación. Además, basado en lo anterior, se cuestiona la interpretación actual de los resultados de estas medidas. Con este fin, etiquetamos manualmente un subconjunto de un conjunto de datos y evaluamos el desempeño de 6 enfoques tradicionales de Supervisión Distante. Demostramos diferencias cuantitativas en los puntajes de evaluación al considerar subconjuntos etiquetados manualmente versus automáticamente. En consecuencia, el orden de desempeño entre los métodos de Supervisión Distante es diferente con ambos etiquetados.

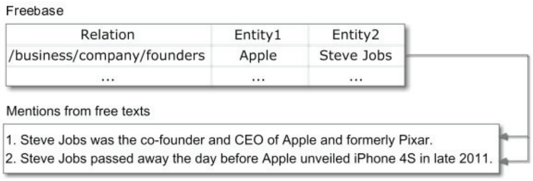
Palabras clave: Extracción de Relaciones. evaluación de la Supervisión Distante. curvas de Precisión-Cobertura. Precisión en N.

ISSN 1135-5948 DOI 10.26342/2022-68-5

1 Introduction

Relation Extraction (RE) is concerned with detecting and classifying predefined relations between entities identified in text (Piskorski and Yangarber, 2013). The traditional RE approach uses a supervised method to create the classifier(s) necessary to identify relations between pairs of named entities (Hearst, 1992; Agichtein and Gravano, 2000; Bunescu and Mooney, 2005). However, this process is slow and expensive; hence an alternative is the use of Distant Supervision (DS).

DS consists of automatically labeling the relations between each pair of named entities in a text using some pre-existing Knowledge Base (KB) (Mintz et al., 2009). For the automatic annotation of the data set with labeled relations, Mintz et al. (2009) assumed that given two entities that participate in a relation, *all* sentences in the data set that include these two entities express that relation (see Figure 1). However, it is common that a pair of entities in a sentence does not necessarily express a relation or may express several relations (see Figure 1). Hence, the assumtion proposed by Mintz et al. (2009) is too strong and often introduces false positives (which basically is noise in the labels) in the train and test sets. Later, Riedel et al. (2010) relaxed this assumption, assuming that "if two entities participate in a relation, *at least one* sentence that mentions these two entities might express that relation". This relaxation alleviates the problem of false positives in the *automatically* generated labels, but it does not fully fix it.



Freebase

Relation	Entity1	Entity2
/business/company/founders	Apple	Steve Jobs
...	...	...

Mentions from free texts

1. Steve Jobs was the co-founder and CEO of Apple and formerly Pixar.
2. Steve Jobs passed away the day before Apple unveiled iPhone 4S in late 2011.

Figure 1: In this example, two sentences with the same pair of entities are automatically labeled with the same relation. Considering the *founders* relation, the first one will be correctly labeled while the second will not (Zeng et al., 2015).

Unfortunately, the evaluation of DS methods is complicated because there is no set correctly labeled to check their performance. Considering this, alternative evaluation methods have been proposed, such as the Precision-Recall (PR) curves or Precision at N (P@N) elements (Mintz et al., 2009). However, these measures are calculated using data labeled with the same automatic process; that is, the labels are not necessarily correct, impairing the calculation of the evaluation results.

This paper[1] aims to analyze the use of these evaluation measures showing that when the methods are evaluated using a correctly labeled set, the performance of the algorithms for DS reported so far varies substantially, thus questioning the current interpretation of the evaluation methods. We assessed the performance of 6 DS algorithms with PR curves and P@N analysis, with a correctly labeled set and with automatically generated labels, and compared the outcomes.

Our contributions can be summarized as follows:

- PR curves and P@N performance measures are critically revisited under competing scenarios of *manual* and *automatic* labeling.
- All sentences with a relation other than NA from the *New York Times* (NYT2010)[2] data set proposed by (Riedel, Yao, and McCallum, 2010) was crowd-labeled using MTurk[3]. So far, the manually annotated datasets for the task do not include all these sentences, which is a strength of this research. We argued that this affords better guarantees over the performance assessment in this task.
- We show that under current practice, performance measures for DS in RE may be misinterpreted when evaluation is carried out over *automatic* potentially noisy labeling.

In general, these contributions can positively impact the DS task evaluation. So far, the evaluation of this task is performed on *automatically* labeled partitions that may introduce incorrect labels. With the *manual* review of the test partition of well-known data set in DS, the performance comparisons of different methods are more reliable. In addition, precision, recall and F1 measures can

[1] Source available at https://github.com/juanluis17/distant-supervision-dataset-evaluation

[2] http://iesl.cs.umass.edu/riedel/ecml/

[3] Mechanical Turk, MTurk, is a human annotation service provided by Amazon.

be incorporated.

2 Related Work

The state-of-the-art in DS includes several solutions using different Deep Learning architectures. One of the first networks was the *Piecewise Convolutional Neural Networks* (PCNN) proposed by Zeng et al. (2015) based on *Convolutional Neural Networks* (CNN) (Zeng et al., 2014). This network incorporates bags of sentences to handle the noise on the labels. A bag of sentences contains sentences that have the same entities pair. Also, it includes a piecewise max-pooling layer "to capture structural information between two entities". Later, different attention mechanisms were incorporated into CNN and PCNN. In (Lin et al., 2016; Ji et al., 2017) an attention mechanism at sentences level (CNN_ATT and PCNN_ATT) in multiple instances was proposed to use the information of all sentences in the bag. Also, in (Ji et al., 2017) description about entities was included. Zhou et al. (2018) select from the bag several instances related to the label to predict the relations and use a word-level attention mechanism to highlight essential parts of the sentence dynamically. Besides, in (Jat, Khandelwal, and Talukdar, 2018), the *Bidirectional Gated Recurrent Unit* architecture was proposed with an attention mechanism over words to identify which key phrases are used (BGWA). Ye and Ling (Ye and Ling, 2019) used intra-bag and inter-bag attention mechanisms while in (Lin et al., 2016; Ji et al., 2017) it is only performed intra-bag, which ignores when all sentences in the bag are false positives. Moreover, Vashishth et al. (2018) propose to RESIDE that uses knowledge base information such as the entity type and relations alias to predict the correct relation. In addition, *Convolutional Graph Networks* (Defferrard, Bresson, and Vandergheynst, 2016) are used over dependency tree for modeling the syntactic information and capturing long-range dependencies. This information and the words and positions embeddings are used to encode the entire sentence. Finally, Bastos et al. (2021) proposed a method using an aggregator that obtains a homogeneous representation with a Graph Neural Network. This representation merges information from the sentence, relation, and the two entities (considering attributes like entity label, entity alias, entity description and the entity type).

Many of these methods have been evaluated with the test partition of the NYT2010 data set. This partition was automatically labeled under some heuristics, and consequently, some instances have been associated with an incorrect label. Given the absence of an adequate gold standard, precision, recall, and F1 measures have not been used to evaluate these methods. Mintz et al. (2009) used, for the first time, the PR Curves and P@N measures in an attempt to evaluate the DS task. These authors stated that PR curves "gives a rough measure of precision without requiring expensive human evaluation, making it useful for parameter setting". In such a case, "rough" is not an accurate statement. Therefore, performance measured with PR curves is dependent on the amount and distribution of noise in the labels. These curves constructed from *automatic* labels are a simple approximation of the performance of DS methods. Despite this problem, several authors continued using PR curves to evaluate and compare the performance of the proposed DS methods, probably leading to misinterpretations (Surdeanu et al., 2012; Zeng et al., 2015; Lin et al., 2016; Jat, Khandelwal, and Talukdar, 2018; Vashishth et al., 2018; Wu, Fan, and Zhang, 2019; Xu and Barbosa, 2019; Ye and Ling, 2019; Bastos et al., 2021; Nadgeri et al., 2021). In addition, P@N has been used in DS with 10, 30, 100, 200, 300, and 500 as the value of N. In P@N, the first N elements represent the most reliable answers of the classifier based on the ranking score. Lin et al. (2016), and Liu et al. (2017) reported P@100, P@200, and P@300 by randomly extracting one sentence for each pair of entities, two sentences or using them all. This evaluation, like in (Mintz et al., 2009), must be done manually on each execution because of the noise inherent to the *automatic* labels. Unfortunately, many works did not explicitly report whether and how the review was done manually (Lin et al., 2016; Liu et al., 2017; Wu, Fan, and Zhang, 2019; Vashishth et al., 2018; Ye and Ling, 2019).

Because of the noise that *automatic* labeling introduces, several efforts have been made to build a *gold standard* to evaluate the DS task. First, Mintz et al. (2009) used MTurk service for manual evaluation of P@N. The first 100 instances of each of the top 10 relations were sent to MTurk. Hoffmann et al.

(2011) manually labeled 1000 sentences from the NYT2010 data set to report the results of their method. These authors stated that "These results provide a good approximation to the true precision but can overestimate the actual recall since we did not manually check the much larger set of sentences where no approach predicted extractions". Based on these 1000 annotated instances, in (Ren et al., 2017) 395 were used as test partition. However, in these instances, there is no more than one sentence per entity pair (Jia et al., 2019). Later, Jiang et al. (2018) label 2040 randomly chosen instances of the NYT2010 data set, including the relation NA. In (Jiang et al., 2018), the performance of 4 DS methods is compared with the automatically annotated NYT2010 data set and the manually annotated data sets proposed by Hoffmann et al. (2011) and Jiang et al. (2018). However, a disadvantage of these data sets is that they do not include the entire NYT2010 test partition. Furthermore, in these papers, the measures of the DS task (i.e., PR curves and P@N) were not studied except for [1], which includes PR curves only. Besides, statistical validations were not carried out, nor were the selection criteria of the instances expressed. Finally, precision, recall, and F1 measures were not reported in most DS papers. Only Hoffmann et al. (2011) reported these measures on the 1000 annotated instances.

3 Background

3.1 Precision-Recall curves

PR curves are frequently used in binary classification (Davis and Goadrich, 2006) and, within this generic problem, in Information Retrieval (IR) (Manning, Raghavan, and Schütze, 2008). PR curves plot precision versus recall for a varying decision threshold parameter in binary classification (Keilwagen, Grosse, and Grau, 2014). These curves are calculated from the (assumed) true label and a score given by the classifier. This analysis is closely related to the Receiver-Operator Curve (ROC) analysis (Davis and Goadrich, 2006) widely used in statistics. However conveniently, for IR purposes, the PR curves can be built without the true negatives (TN). To get a scalar score, the area under PR curves (AUC) can be calculated by using the composite trapezoidal method (Davis and Goadrich, 2006).

Let Γ be a threshold set defined over classifier scores, and Ψ be a vector of descending ordered scores given by a classifier. The Precision and Recall for a threshold $\gamma \in \Gamma$ are calculated using the equations 1 and 2 respectively $\forall \psi \in \Psi \mid \psi > \gamma$.

$$P_\gamma = \frac{TP_\gamma}{TP_\gamma + FP_\gamma} \qquad \gamma \in \Gamma \tag{1}$$

$$R_\gamma = \frac{TP_\gamma}{TP_\gamma + FN_\gamma} \qquad \gamma \in \Gamma \tag{2}$$

where TP are positive examples correctly labeled as positives, FP are negative examples mislabelled as positives and FN are positive examples incorrectly labeled as negative.

To obtain the set of pairs (R_γ, P_γ) in the PR curve, we iterate over Γ as per Equation 3:

$$PR\_Curve(\gamma) = \{(R_\gamma, P_\gamma) : \gamma \in \Gamma\} \tag{3}$$

3.2 Precision at N

The P@N in Equation 4 measures the number of correct elements in a window of N elements (Manning, Raghavan, and Schütze, 2008).

$$P@N = \frac{|TP \cap R_N|}{N} \tag{4}$$

The TP (positive examples correctly labeled as positives) is calculated by manual evaluation. The P@N is frequently used in IR to measure the precision in a subset of retrieved elements R_N, with N the cardinality of the set. According to (Manning, Raghavan, and Schütze, 2008), it has the advantage of not requiring any estimate of the size of the set of relevant elements. P@N has been used in DS by multiple authors, but in most cases, this has been on the automatically labeled data set (with noisy labels) (Zeng et al., 2015; Lin et al., 2016; Ji et al., 2017; He et al., 2018; Wang et al., 2018; Wu, Fan, and Zhang, 2019; Ye and Ling, 2019; Bastos et al., 2021; Nadgeri et al., 2021).

4 Methodology

4.1 Dataset preparation

In order to establish whether there are risks of misinterpreting the evaluation measures, we compared the performance of 6 DS methods assessed over *manually-generated* labels and *automatically-generated* labels. We depart from the NYT2010 data set for the DS task. This data set includes 53 relations types, including $\mathcal{NA}$, when there is no relation. Originally, this data set was labeled *automatically*. The *train* partition has 522611 instances (sentence that may or may not contain a relation), 279226 unique entity pairs and 154929 instances with a relation other than $\mathcal{NA}$. We use this training partition, with the *automatically* generated labels, to train the algorithms. In turn, the test partition has 172448 instances, 96678 unique entity pairs and 6444 instances with a relation other than $\mathcal{NA}$. From this last partition, two test partitions with *manual* labels were built and used in this work.

In the first test partition, 430 instances were selected for *manual* revision. The instances selection to be reviewed was made by choosing one instance from each relation at random during 20 iterations. During *manual* revision, 88 duplicate instances and 18 that have unclear relations were found and removed. Thus, the remaining 324 instances were revised *manually* and constitute our first test partition (named *test_1*). Considering the 324 instances of the *test_1* partition, 158 (48.8%) changed their *automatic* label after their/the review, i.e., they were considered by a human to hold incorrect labels.

In the second test partition, the complete 6444 instances different from the relation $\mathcal{NA}$ were selected for *manual* revision. First, we curated the 6444 instances by removing invalid instances. An instance is considered invalid when the defined entities are not found in the sentence. A total of 6431 were found valid. Then, from the 6431 valid instances, we further eliminated 579 duplicate instances (containing the same sentence, entity pair, and relation). We publish the remaining 5852 instances on the MTurk for review by three reviewers. The reviewers only determined whether the sentence explicitly expressed the associated relation.

Finally, we consider an instance as noisy if at least two of the three judges decided that the relations were not expressed. 4801 instances did not vary their *automatic* label but 1051 did (17.9%). This partition was named *test_2*.

4.2 Selection of DS methods for comparison

The following DS methods were compared in their performance:

- PCNN (Zeng et al., 2015) and CNN: The authors used both PR curves and P@N for evaluation, and the labeling was performed manually. This was one of the first architectures to be used in DS.
- PCNN_ATT (Lin et al., 2016) and CNN_ATT: The authors incorporated an attention mechanism over instances. They used PR curves to determine the performance of the attention mechanism compared to other methods. Finally, P@N was calculated on automatically generated *automatic* labels.
- BGWA (Jat, Khandelwal, and Talukdar, 2018): It incorporates an attention mechanism over words and entities. Only the PR curves were used as a measure to compare the performance of BGWA concerning the rest.
- RESIDE (Vashishth et al., 2018): It combines syntactic information with entity types and relations aliases. Like (Lin et al., 2016), P@N was calculated automatically on *automatic* labels.

These methods were chosen because they use three different architectures. On the one hand, CNN and PCNN use a convolutional architecture to which an attention mechanism is then incorporated (CNN_ATT and PCNN_ATT). On the other hand, RESIDE uses Graph Convolution Networks and Bidirectional Gated Recurrent Unit (the latter used by BGWA) and incorporates information about entities and relations. The execution of these methods was done in the same way as defined in Github[4] without using the gradient descent optimizer. To compare the evaluation measures, we trained these methods with the NYT2010 train partition proposed by (Riedel, Yao, and McCallum, 2010). Then, we evaluate them with the *test_1* and *test_2* partitions on the *automatic* and *manual* labels (see Figure 2).

4.3 Experimental design

In order to fairly evaluate the performance obtained, replications are necessary to ensure that chance does not play a role in our results. The number of replications (sample size) was determined using power analysis. Power analysis refers to the estimation of the probability of correctly rejecting a false null hypothesis when a particular alternative hypothesis is true (Howell, 2012).

[4] https://github.com/malllabiisc/RESIDE

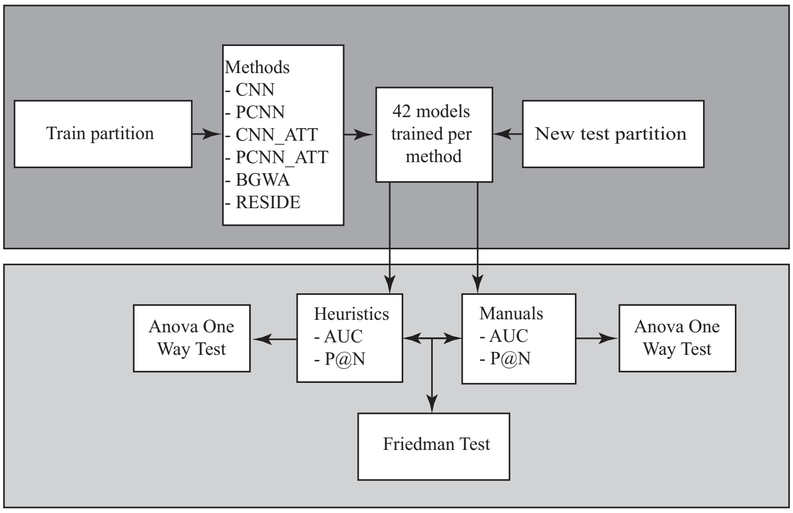


Figure 2: This diagram depicts the methodology followed in the current research. The top box illustrates the experiment design. The bottom box summarizes the statistical hypothesis testing followed.

The analysis depends on four factors: statistical significance, effect size, sample size and the statistical power itself. Fixing any three, yields the fourth for a given hypothesis model. The power analysis was estimated using the ANOVA One Way test for a desired significance level of 0.05, statistical power of $\beta = 0.95$ and assuming an effect size of Cohen's $d = 0.4$. As a result, 42 repetitions per treatment (i.e., algorithm to be compared) was obtained as the required sample size. The samples number here represents the number of executions for each method, that is, the replications required to detect an effect of the assumed size in the experiment.

From the results of the replications, the Friedman test was used to determine if there were differences in the ranking of the methods using automatic labels concerning manual labels. First, the Friedman test is used for one-way repeated measures analysis of variance by ranks (Friedman, 1940). This test only considers the number that each method occupies in the ranking and not the measure values. This is because the measure values are only used to determine ranking. Then, the ANOVA One Way test is applied on *automatic* and *manual* labels to know if there are significant differences between the results achieved by the methods. The ANOVA One Way test is used to test for differences among at least three groups, with the two-group case covered by the simpler t-test (Student, 1908; Howell, 2012). Finally, if there were significant differences, pairwise comparisons were made to observe which pair of methods showed differences. The two-by-two comparisons were made with t-test and Holm Correction (Holm, 1979). The significance threshold was set at $p < 0.05$.

5 *Experiments*

5.1 Precision-Recall curves

Performance on *test_1* partition

The Table 1 summarizes the AUC of the tested methods PR curves with *automatic* and *manual* labels on *test_1*. All methods increased their AUC with the *manual* labels with regards to their performances using the *automatic* ones, pointing to a systematic overall underestimation. Further, and more critically here, the order of the methods in terms of their performance varied significantly (Friedman: $\chi^2(2) = 373.46$, $p < 2.2e^{-16}$), i.e., they are all underestimated but not in the same extent. This suggests that using PR curves with *automatic* labels might not conferring the direct message one would expect otherwise in the DS evaluation task, and that for this scenario, such bias has to be considered during interpretation. Besides, significant differences were found with either *automatics* (ANOVA: $F(5, 246) = 746.9, p < 2e^{-16}$) and *manual* labels (ANOVA: $F(5, 246) = 520.8, p < 2e^{-16}$). In the case of pairwise comparisons, BGWA presents significant differences from the other methods for both labels.

The Figures 3a and 3b show the PR curves obtained by BGWA, RESIDE, PCNN,

Automatic labels		*Manual* labels	
Model	AUC	Model	AUC
BGWA	0.412 ± 0.026^a	BGWA	0.440 ± 0.023^a
CNN_ATT	0.194 ± 0.022^b	CNN_ATT	0.239 ± 0.031^b
CNN	0.193 ± 0.027^b	CNN	0.235 ± 0.027^c
RESIDE	0.191 ± 0.013^b	PCNN	0.209 ± 0.028^d
PCNN	0.158 ± 0.023^c	RESIDE	0.199 ± 0.020^d
PCNN_ATT	0.151 ± 0.025^d	PCNN_ATT	0.197 ± 0.029^d

[a] differences with rest of methods***.
[b] differences with BGWA***, PCNN*** and PCNN_ATT***.
[c] differences with rest of methods*** except PCNN_ATT.)
[d] differences with rest of methods*** except PCNN.
*, **, *** to indicate $p < 0.05$, $p < 0.01$ and $p < 0.001$ respectively.

[a] differences with rest of methods***.
[b] differences with rest of methods*** except CNN.
[c] differences with rest of methods*** except CNN_ATT.
[d] differences with BGWA***, CNN*** and CNN_ATT***.

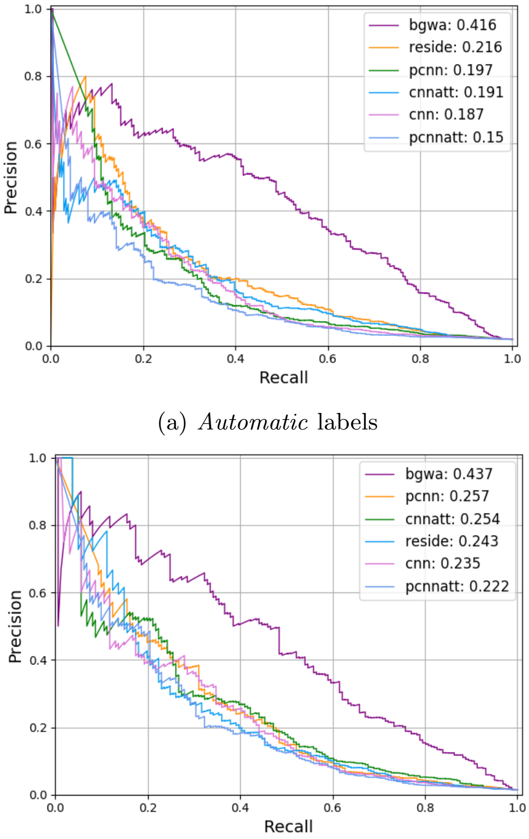
Table 1: AUC of the PR curves after 42 replications with *automatic* and *manual* labels on *test_1*.

PCNN_ATT, CNN and CNN_ATT in one execution made with *automatic* and *manual* labels respectively on *test_1*. It can be appreciated that the ordering of the algorithms according to their performance in terms of AUC varies when using the *manual* labels concerning the *automatic* ones (previously validated with Friedman test and multiples executions).

Performance on *test_2* partition

As with the *test_1* partition, the AUC values of the PR curves with *automatic* and *manual* labels on *test_2* were obtained (see Table 2). In these tables, similar values are observed with both labels. However, as in *test_1*, the order of the methods varied significantly (Friedman: $\chi^2(2) = 785.37$, $p < 2.2e^{-16}$. Similarly, significant differences were found with *automatics* labels (ANOVA: $F(5, 246) = 2097, p < 2e^{-16}$). Analogously, significant differences were found (ANOVA: $F(5, 246) = 1553, p < 2e^{-16}$) on *manual* labels. As in *test_1*, BGWA presents significant differences from the other methods for both labels in pairwise comparisons. However, there were no differences between PCNN_ATT, CNN_ATT and PCNN for *automatic* labels. Besides, no differences were found in the *manual* labels between PCNN_ATT and PCNN methods.

The Figures 4a and 4b show the PR curves in one execution made with *automatic* and *manual* labels respectively on *test_2*.



(a) *Automatic* labels

(b) *Manual* labels

Figure 3: PR curves corresponding to evaluation of the DS algorithms over *test_1* (one execution) set pick for verification in (a) *automatic* labels and (b) *manual* labels. The AUC of the PR curves is indicated beside each label in the legend.

5.2 Precision at N

Performance on *test_1* partition

The P@25 and P@50 subsets from the *test_1* partition were established in addition to all the instances (P@All). Table 3 shows that the order of the models remains the same for the first three models by increasing N, unlike the last three positions. The same happens with Table 4 where, in this case, the first two models are kept. The order of the models, as with the AUC, varied significantly for

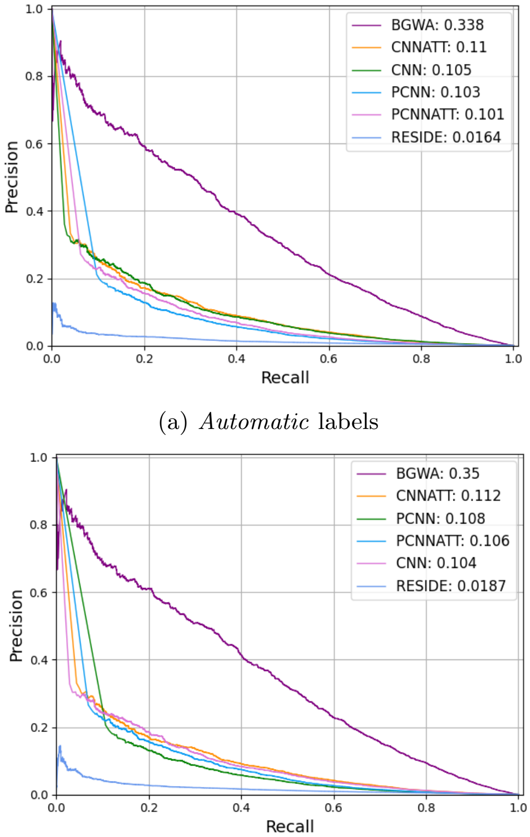
Automatic labels		*Manual* labels	
Model	AUC	Model	AUC
BGWA	0.339 ± 0.016^a	BGWA	0.345 ± 0.021^a
PCNN_ATT	0.112 ± 0.015^b	PCNN_ATT	0.118 ± 0.017^b
CNN_ATT	0.105 ± 0.017^c	PCNN	0.109 ± 0.020^c
PCNN	0.105 ± 0.018^c	CNN_ATT	0.106 ± 0.018^d
CNN	0.098 ± 0.016^d	CNN	0.098 ± 0.017^e
RESIDE	0.021 ± 0.006^c	RESIDE	0.028 ± 0.011^f

[a] differences with rest of methods***.
[b] differences with BGWA*** and CNN***.
[c] differences with BGWA***.
[d] differences with BGWA*** and PCNN_ATT***.

[a] differences with rest of methods***.
[b] differences with CNN*** and CNN_ATT*.
[c] differences with BGWA*** and CNN*.
[d] differences with BGWA*** and PCNN_ATT***.
[e] differences with BGWA***, PCNN_ATT*** and PCNN*.
[f] differences with BGWA***

*, **, *** to indicate $p < 0.05$, $p < 0.01$ and $p < 0.001$ respectively.

Table 2: AUC of the PR curves after 42 replications with *automatic* and *manual* labels on *test_2*.



(a) *Automatic* labels

(b) *Manual* labels

Figure 4: PR curves corresponding to evaluation of the DS algorithms over *test_2* (one execution) set pick for verification in (a) *automatic* labels and (b) *manual* labels. The AUC of the PR curves is indicated beside each label in the legend.

the *automatic* and *manual* labels on P@All (Friedman: $\chi^2(2) = 382.28$, $p < 2.2e^{-16}$). Similarly, there are significant differences in the performance of methods with *automatic* (ANOVA: $F(5, 246) = 210.8, p < 2e^{-16}$) and *manual* (ANOVA: $F(5, 246) = 255.6, p < 2e^{-16}$) labels. Then, two-by-two comparisons with Holm Correction (Holm, 1979) show significant differences with *automatic* labels between the BGWA and RESIDE models and the rest. Similarly, two-by-two comparisons show significant differences with manual labels between the BGWA model and the rest. In addition, PCNN_ATT has significant differences with the other models except for PCNN (in reverse order, it also happens). In this case, RESIDE only shows significant differences with BGWA, PCNN and PCNN_ATT.

Performance on *test_2* partition

In the same way as with *test_1*, the subsets P@25 and P@50 were established together with P@All, which includes the entire set. With both labeled, only two methods did not vary their order in the three subsets, BGWA and RESIDE (see Tables 5 and 6). In addition, the order of the methods using the P@All results varied significantly concerning the *automatic* and *manual* labels (Friedman: $\chi^2(2) = 369.55$, $p < 2.2e^{-16}$)[5]. Similarly, significant differences were found in the performance of the methods with *automatic* (ANOVA: $F(5, 246) = 1610, p < 2e^{-16}$) and *manual* (ANOVA: $F(5, 246) = 1265, p < 2e^{-16}$) labels. Then, in two-by-two comparisons with Holm Correction (Holm, 1979) there are no significant differences only between the CNN and CNN_ATT and PCNN and PCNN_ATT methods with both labeled.

[5] It should be noted that in all cases the Friedman test is used on the ranking of each execution, not only on the final results.

Model	P@25	Model	P@50	Model	P@All
BGWA	0.819±0.062	BGWA	0.730±0.041	BGWA	0.558±0.029
CNN	0.587±0.087	CNN	0.489±0.062	CNN	0.386±0.036
CNN_ATT	0.580±0.089	CNN_ATT	0.486±0.064	CNN_ATT	0.375±0.045
PCNN	0.554±0.087	PCNN_ATT	0.461±0.055	PCNN	0.362±0.037
RESIDE	0.552±0.074	PCNN	0.459±0.060	PCNN_ATT	0.351±0.040
PCNN_ATT	0.550±0.079	RESIDE	0.433±0.054	RESIDE	0.325±0.035

Table 3: P@25, P@50 and P@All after 42 replications with *automatic* labels on *test_1*.

Model	P@25	Model	P@50	Model	P@All
BGWA	0.715±0.079	BGWA	0.677±0.044	BGWA	0.585±0.033
RESIDE	0.555±0.075	RESIDE	0.489±0.043	RESIDE	0.376±0.037
CNN	0.551±0.089	CNN_ATT	0.465±0.061	CNN	0.370±0.035
CNN_ATT	0.544±0.089	CNN	0.459±0.059	CNN_ATT	0.370±0.044
PCNN	0.486±0.093	PCNN	0.401±0.062	PCNN	0.328±0.044
PCNN_ATT	0.458±0.096	PCNN_ATT	0.399±0.066	PCNN_ATT	0.325±0.041

Table 4: P@25, P@50 and P@All after 42 replications with *manual* labels on *test_1*.

6 Discussion

Our results indicate that the ranking of the methods, in terms of the AUC of the PR curves on *test_1* and *test_2* partition, differ depending on the labeling. This justifies our claim that the interpretation of the PR curves must be reconsidered when used for evaluating DS algorithms. PR curves using *automatic* labels as a reference is not an optimal way to compare methods performance in DS because it breaks a premise of the PR curves construction; that *true* labels are available. Several authors have based the comparison of their method on the PR curves on these labels (Riedel, Yao, and McCallum, 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Zeng et al., 2015; Lin et al., 2016; Jiang et al., 2016; Liu et al., 2017; Vashishth et al., 2018; Ru et al., 2018; Zhou et al., 2018; Wang et al., 2018; Jat, Khandelwal, and Talukdar, 2018; Wu, Fan, and Zhang, 2019; Xu and Barbosa, 2019; Ye and Ling, 2019; Bastos et al., 2021; Nadgeri et al., 2021). The classical interpretation does not provide guarantees as to which method is performing better or which one is more tolerant to noise in the labels.

The Section 5.2 has also confirmed that P@N is not being interpreted correctly in DS either. This is critical for the task at hand considering the unbalance in the data sets, variability among the relations, selection criteria, among others. There is no clear selection criterion that guarantees to choose the same instances for evaluating each of the methods. In other words, it is not guaranteed that the first instances chosen to evaluate one method are the same for another method. If the selection is based on the classifier's score, it varies from one execution to another. The same happens if the selection is random. For example, the first N instances can be of the same relation for a method. This indicates how good this method is for that relation. However, for the rest, its performance is not known. Also, sometimes, the P@N is calculated over *automatic* labels, whereas some works do it over *manual* labels. This is the case of the 6 methods used in this work. This further confuses P@N's interpretation. Furthermore, dispersion values are not reported in the previous works, which mathematically renders those works uninformative.

What was expressed above shows that PR curves and P@N measures are not currently being interpreted properly in DS due to the presence of noisy labels. Currently, we believe there are no reliable statistics regarding the actual performance of the DS methods. While the community agrees on a mathematically correct interpretation in this context, or new statistics are proposed for evaluating the performance of DS methods, a possible strategy to circumvent the deadlock is what was done here. That is, selecting multiple instances of the evaluation data set while maintaining its distribution (*test_1* partition). Then, perform a manual review of these instances using multiple raters. The main limitations of *test_1* partition are the instances number selected. This

Model	P@25	Model	P@50	Model	P@All
BGWA	0.804 ± 0.082	BGWA	0.762 ± 0.064	BGWA	0.019 ± 0.000
CNNATT	0.360 ± 0.112	CNN	0.357 ± 0.084	CNN	0.015 ± 0.000
CNN	0.346 ± 0.111	CNNATT	0.341 ± 0.087	CNNATT	0.015 ± 0.000
PCNNATT	0.273 ± 0.089	PCNNATT	0.268 ± 0.067	PCNN	0.014 ± 0.000
PCNN	0.252 ± 0.106	PCNN	0.233 ± 0.070	PCNNATT	0.014 ± 0.000
RESIDE	0.115 ± 0.095	RESIDE	0.129 ± 0.076	RESIDE	0.010 ± 0.000

Table 5: P@25, P@50 and P@All after 42 replications with *automatic* labels on *test_2*.

Model	P@25	Model	P@50	Model	P@All
BGWA	0.017 ± 0.000	BGWA	0.795 ± 0.083	BGWA	0.0168 ± 0.000
CNN	0.014 ± 0.000	CNNATT	0.343 ± 0.120	CNN	0.0137 ± 0.000
CNNATT	0.014 ± 0.000	CNN	0.320 ± 0.117	CNNATT	0.0135 ± 0.000
PCNN	0.013 ± 0.000	PCNNATT	0.255 ± 0.104	PCNN	0.0130 ± 0.000
PCNNATT	0.013 ± 0.000	PCNN	0.230 ± 0.099	PCNNATT	0.0129 ± 0.000
RESIDE	0.010 ± 0.000	RESIDE	0.150 ± 0.094	RESIDE	0.0103 ± 0.000

Table 6: P@25, P@50 and P@All after 42 replications with *manual* labels on *test_2*.

is why the *test_2* partition was labeled with multiple raters using MTurk. The advantage of this partition concerning *test_1* and those proposed by (Hoffmann et al., 2011), (Ren et al., 2017) and (Jiang et al., 2018) is that it is made up of all the instances of the NYT2010 data set test partition (only those different from NA were labeled with MTurk). From the *test_2* partition, the methods can be compared with precision, recall and F1 using the traditional interpretation. Besides, in (Jiang et al., 2018), although the performance of the CNN, PCNNN, CNN_ATT and PCNNN_ATT methods are analyzed, the P@N measure, the BGWA and RESIDE methods and statistical validations are not included. One limitation of this evaluation alternative is that *manual* labeling of the test partition is expensive but only done once. In addition, experts in the area are needed for this labeling in most cases. However, in this work, it has been shown that the performance of the methods using *automatic* labeling can be misinterpreted.

7 Conclusions

Significant differences were found in the ranking of the methods regarding their performances when the performance is established according to the AUC of the PR curves between the evaluation using the *automatic* labels and the same data set with the *manual* labels. The largest AUCs were obtained using *manual* labels which speaks well of the capacity of the DS methods to handle noisy data as it is their core intention. Our results suggest that PR curves are currently not being interpreted correctly in DS. Furthermore, they suggest that the PR curves calculated using the *automatically* labeled data should not be used to compare the performance of DS methods. In addition, manual evaluation of the first N instances (P@N) does not cover the entire data set. The existing selection criteria for the instances to be manually reviewed are not deterministic, suggesting multiple executions of the method and the dispersion report. Besides, as they are being used, these measures are inconclusive as to the performance of those methods. Finally, we provided a partition that allows you to evaluate this task using labels manually reviewed by multiple raters. This partition also allows the use of precision, recall and F1 measures and will be available for use by the area community. In future work, we will analyze various DS methods using these two partitions and the traditional precision, recall, and F1 measures. In addition, we will continue to work on the DS evaluation methods.

Acknowledgments

The present work was supported by CONACyT/México (scholarship 937210 and grant CB-2015-01-257383). Additionally, the authors thank CONACYT for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies.

References

Agichtein, E. and L. Gravano. 2000. Snowball: Extracting Relations from large Plain-Text Collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Bastos, A., A. Nadgeri, K. Singh, I. O. Mulang', S. Shekarpour, J. Hoffart, and M. Kaul. 2021. RECON: Relation Extraction using Knowledge Graph Context in a Graph Neural Network. In *Proceedings of the Web Conference 2021*, pages 1673–1685, Ljubljana.

Bunescu, R. C. and R. J. Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 724–731, Vancouver,. Association for Computational Linguistics.

Davis, J. and M. Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, pages 233–240, Pittsburgh, Pennsylvania, USA. ACM Press.

Defferrard, M., X. Bresson, and P. Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in neural information processing systems*, pages 3844–3852.

Friedman, M. 1940. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.

He, Z., W. Chen, Z. Li, M. Zhang, W. Zhang, and M. Zhang. 2018. SEE: Syntax-Aware Entity Embedding for Neural Relation Extraction. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5795–5802. Association for the Advancement of Artificial Intelligence.

Hearst, M. A. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING-92)*, pages 539–545, Nantes.

Hoffmann, R., C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting ofthe Association for Computational Linguistics*, pages 541–550, Portland, Oregon. Association for Computational Linguistics.

Holm, S. 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Howell, D. C. 2012. *Statistical Methods for Psychology.* Cengage Learning ALL.

Jat, S., S. Khandelwal, and P. Talukdar. 2018. Improving Distantly Supervised Relation Extraction using Word and Entity Based Attention. *arXiv*, [cs.CL](1804.06987v1), apr.

Ji, G., K. Liu, S. He, and J. Zhao. 2017. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3060–3066.

Jia, W., D. Dai, X. Xiao, and H. Wu. 2019. ARNOR: Attention Regularization based Noise Reduction for Distant Supervision Relation Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1399–1408, Florence. Association for Computational Linguistics.

Jiang, T., J. Liu, C.-Y. Lin, and Z. Sui. 2018. Revisiting distant supervision for relation extraction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Jiang, X., Q. Wang, P. Li, and B. Wang. 2016. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480, Osaka.

Keilwagen, J., I. Grosse, and J. Grau. 2014. Area under Precision-Recall Curves for Weighted and Unweighted Data. *PLoS ONE*, 9(3):e92209, mar.

Lin, Y., S. Shen, Z. Liu, H. Luan, and M. Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting ofthe Association for Computational Linguistics*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Liu, T., K. Wang, B. Chang, and Z. Sui. 2017. A Soft-label Method for Noise-tolerant Distantly Supervised Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790–1795, Copenhagen, Denmark.

Manning, C. D., P. Raghavan, and H. Schütze. 2008. *Introduction to information retrieval.* Cambridge University Press.

Mintz, M., S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL*, pages 1003–1011, Suntec, Singapore.

Nadgeri, A., A. Bastos, K. Singh, I. O. Mulang', J. Hoffart, S. Shekarpour, and V. Saraswat. 2021. KGPool: Dynamic Knowledge Graph Context Selection for Relation Extraction. *arXiv*, 2106.00459, jun.

Piskorski, J. and R. Yangarber. 2013. Information extraction: Past, Present and Future. In *Multi-source, Multilingual Information Extraction and Summarization 11.* Springer-Verlag Berlin Heidelberg, pages 23–49.

Ren, X., Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han. 2017. CoType. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024, Perth, apr. International World Wide Web Conferences Steering Committee.

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

Ru, C., J. Tang, S. Li, S. Xie, and T. Wang. 2018. Using semantic similarity to reduce wrong labels in distant supervision for relation extraction. *Information Processing Management*, 54(4):593–608, jul.

Student. 1908. The Probable Error of a Mean. *Biometrika*, 6(1):1–25.

Surdeanu, M., J. Tibshirani, R. Nallapati, and C. 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. Association for Computational Linguistics.

Vashishth, S., R. Joshi, S. S. Prayaga, C. Bhattacharyya, and P. Talukdar. 2018. Reside: Improving Distantly-Supervised Neural Relation Extraction using Side Information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266, Brussels, Belgium. Association for Computational Linguistics.

Wang, G., W. Zhang, R. Wang, Y. Zhou, L. Chen, W. Zhang, H. Zhu, and H. Chen. 2018. Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2246–2255.

Wu, S., K. Fan, and Q. Zhang. 2019. Improving Distantly Supervised Relation Extraction with Neural Noise Converter and Conditional Optimal Selector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7273–7280, nov.

Xu, P. and D. Barbosa. 2019. Connecting Language and Knowledge with Heterogeneous Representations for Neural Relation Extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3201–3206, Minneapolis, Minnesota. Association for Computational Linguistics.

Ye, Z.-X. and Z.-H. Ling. 2019. Distant Supervision Relation Extraction with Intra-Bag and Inter-Bag Attentions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Hu-*

man Language Technologies, pages 2810–2819, Minneapolis, Minnesota. Association for Computational Linguistics.

Zeng, D., K. Liu, Y. Chen, and J. 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. Association for Computational Linguistics.

Zeng, D., K. Liu, S. Lai, G. Zhou, and J. Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland.

Zhou, P., J. Xu, Z. Qi, H. Bao, Z. Chen, and B. Xu. 2018. Distant supervision for relation extraction with hierarchical selective attention. *Neural Networks*, 108:240–247.