Evaluation of a new representation for noise reduction in distant supervision

García-Mendoza, Juan Luis; Villaseñor-Pineda, Luis; Buscaldi, Davide; Bustio-Martínez, Lázaro; Orihuela-Espina, Felipe

DOI:
10.1007/978-3-031-19496-2_8

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
García-Mendoza, JL, Villaseñor-Pineda, L, Buscaldi, D, Bustio-Martínez, L & Orihuela-Espina, F 2022, Evaluation of a new representation for noise reduction in distant supervision. in O Pichardo Lagunas, B Martínez Seis & J Martínez-Miranda (eds), *Advances in Computational Intelligence: 21st Mexican International Conference on Artificial Intelligence, MICAI 2022, Proceedings, Part II.* 1 edn, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13613 LNAI, Springer, pp. 101-113, 21st Mexican International Conference on Artificial Intelligence, MICAI 2022, Monterrey, Mexico, 24/10/22. https://doi.org/10.1007/978-3-031-19496-2_8

Evaluation of a New Representation for Noise Reduction in Distant Supervision

Juan-Luis García-Mendoza[1], Luis Villaseñor-Pineda[1], Davide Buscaldi[2], Lázaro Bustio-Martínez[3], and Felipe Orihuela-Espina[1,4]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México
{juanluis,villasen}@inaoep.mx
[2] Université Sorbonne Paris Nord, LIPN, Villetaneuse, France
davide.buscaldi@lipn.univ-paris13.fr
[3] Universidad Iberoamericana, DEII, CDMX, México
lbustio@ibero.mx
[4] University of Birmigham, Birmingham, United Kingdom
f.orihuela-espina@bham.ac.uk

Abstract. Distant Supervision is a relation extraction approach that allows automatic labeling of a dataset. However, this labeling introduces noise in the labels (e.g., when two entities in a sentence are automatically labeled with an invalid relation). Noise in labels makes difficult the relation extraction task. This noise is precisely one of the main challenges of this task. Until now, the methods that incorporate a previous noise reduction step do not evaluate the performance of this step. This paper evaluates the noise reduction using a new representation obtained with autoencoders. In addition, it was incoporated more information to the input of the autoencoder proposed in the state-of-the-art to improve the representation over which the noise is reduced. Also, three methods were proposed to select the instances considered as real. As a result, it was obtained the highest values of the area under the ROC curves using the improved input combined with state-of-the-art anomaly detection methods. Moreover, the three proposed selection methods significantly improve the existing method in the literature.

Keywords: Noise Reduction · Distant Supervision · Adversarial Autoencoders · Data representation

1 Introduction

The goal of the Relation Extraction (RE) task is the extraction and classification of the relations existing between two entities of interest in a sentence [24]. Several approaches for solving this task have been proposed, which can be consulted in [8,13,29,21,6].

One of these approaches is Distant Supervision (DS) [21]. DS allows the automatic labeling of a dataset based on existing knowledge about a specific domain [28]. This knowledge is generally stored in knowledge bases such as Freebase[5].

[5] https://developers.google.com/freebase/

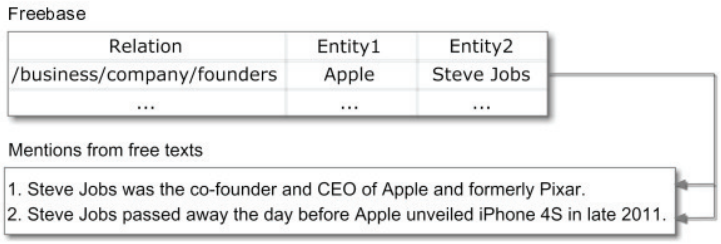


Fig. 1: In this example, two sentences with the same pair of entities are automatically labeled with the same relation. Considering the *founders* relation, the first one will be correctly labeled while the second will not [33].

The labeling is performed following the idea proposed in [21]. Mintz et al. expressed that "if two entities participate in a relation, any sentence that contains those two entities might express that relation". That is, given two entities in a sentence, these entities are searched in the knowledge base. If there is a relation between these entities, the sentence is labeled with this relation. Otherwise, it is labeled with "Not a relation" ($\mathcal{NA}$) (see Figure 1). This idea is the most commonly used in automatic labeling.

In [25], the idea proposed by Mintz et al. in [21] was relaxed because all sentences with the same pair of entities do not necessarily express the relation. Therefore, Reidel et al. concluded that "if two entities participate in a relation, at least one sentence that mentions these two entities might express that relation". Despite this, it may happen that no sentence expresses the relation. This idea is frequently used in the heuristics of methods for solving DS task.

Automatic labeling in DS introduces instances with noise in the labels due to following the idea proposed by Mintz et al. in [21]. These noisy instances are considered false positives. For example, in Figure 1, the second sentence is considered as a false positive. This is because this sentence is labeled with the relation *founder* and it does not actually express such relation. This noise is precisely one of the problems of DS [28]. The late problem has been addressed in two ways. The first one is the inclusion of a noise tolerance mechanism within the proposed methods [21,25,33,16,12,31,32]. While the second one includes a previous noise reduction step [30,26,7]. However, none of the previous work that incorporates a previous noise reduction step evaluates the performance of that step.

Therefore, the aim of this paper is to evaluate the noise reduction on the representations obtained with autoencoders (AE). This evaluation is performed considering the area under the ROC curves (AUC). The ROC curves are obtained as a result of applying state-of-the-art anomaly detection methods to the obtained representations. For this purpose, a dataset was used for the RE task because the noise can be controlled. In DS datasets, the noise introduced by automatic labeling is not known a priori and cannot be controlled.

In addition, it was proposed to add more information to the input of the AE reported in [7] to improve the representation used in noise reduction. Finally, three methods were proposed to select instances used in the adversarial autoencoders (AAE) to obtain the a priori known distribution.

2 Related Work

According to [28], DS can be divided, into three categories. However, these methods are not exclusive to each category. In this work, noise handling is used as a division criterion. Based on this criterion, two groups are formed: *noisy label tolerant methods* [25,33,16,12,31,32] and *noisy label cleaning methods* [30,26,7]. On the one hand, *noisy label tolerant methods* incorporate a mechanism to handle noise within the method itself. For example, in the approach reported in [21], features from several sentences were combined into an enriched vector that was able to tolerate noise. Based on the idea proposed in [25], a multi-instance learning approach was used in several neural networks [33,16,12,32]. The main idea is to consider a bag of instances containing the same pair of entities. In addition, some mechanisms have been added to these networks at the word level [11], at the sentence level [16], at the entity level [12], at the intra-bag level [34] and the intra-bag level and inter-bag level [32]. Finally, several papers included information from knowledge bases such as entity type and relations alias [31] and entity label, entity alias, entity description and the entity type [2].

On the other hand, *noisy label cleaning methods* incorporate a previous noise reduction step. In [30], negative patterns were used to remove noisy labels. Elements such as the syntactic tree path between the two entities are considered if it does not exceed 4 steps. Later, in [26], an algorithm calculates the semantic similarity between text fragments is used to reduce noisy labels. The idea here is to compute the semantic similarity that exists between the triplet stored in the knowledge base representing the relation and the dependency phrase between the two entities. Finally, in [7], architectures based on classical and adversarial AE are used to obtain data representations that allow noise reduction. These representations are obtained by training different AE for each relation. After the noise reduction step, new datasets with less noise were obtained, which can be used by classifiers and will obtain better performance/result. In the revised state-of-the-art papers, the classifiers performance is evaluated using precision-recall curves and precision at N elements obtained. Nevertheless, the performance of the representations used in noise reduction was not evaluated. In [7], the input of the AE and AAE is the vector of the complete sentence. This vector is calculated with pretrained embeddings proposed in [4,1].

AAE is one of the most common AE to obtain representations using unsupervised approaches. AAE uses an adversarial training procedure to force the generated vectors to fit a known prior distribution [20]. The known prior distribution is generated in [7] from randomly selected instances.

3 Methodology

Anomaly detection, according to [22], "is referred to as the process of detecting data instances that significantly deviate from the majority of data instances". In the following, we formally define this problem in the DS task.
Let:

- A set of sentences $\mathcal{S} = \{s_i | i = 1 \dots |S|\}$.
- A set of entities $\mathcal{E} = \{e_z | z = 1 \dots |\mathcal{E}|\}$.
- A set of relations $\mathcal{R} = \{r_j | j = 1 \dots J\}$. One of these relations is the $\mathcal{NA}$ relation.
- A set of observations $\mathcal{X} = \{x_k | x_k = (s_i, e_h, e_t, r_j) \in \mathcal{S} \times \mathcal{E} \times \mathcal{E} \times \mathcal{R}\}$.
- A subset of observations $\mathcal{X}_n \subseteq \mathcal{X}$ where the relation r_j is noisy (the relation is not expressed).
- An *encoder* function to obtain data representation where $\mathcal{V} = \{v_i | v_i \in \mathbb{R}^n\}$ is the vector representation of each sentence s_i.

$$\begin{aligned} encoder : & \mathcal{S} \rightarrow \mathcal{V} \\ & (s_i, v_i) \end{aligned} \tag{1}$$

- A *noisy* function that determines whether the sentence s_i is noisy or not from its v_i representation.

3.1 Dataset

One of the main datasets in DS task, New York Times 2010 (NYT2010)[6] was automatically labeled by Riedel et al. [25]. This labeling results in instances with noise in the labels that are not known a priori. Because of this, noise cannot be controlled during the experiments. An alternative to this problem is to use a dataset of the RE task. SemEval-2010 Task 8 (*semeval2010*) [9] dataset was released as part of Task 8 of the SemEval-2010 event [9]. It has 10 relations, including "Other" which represents $\mathcal{NA}$. Nine of these relations are represented in a bidirectional way becoming 18 relations, which when adding $\mathcal{NA}$ are a total of 19. In the training and test partitions there are 8000 and 2717 instances respectively. We take as *inlier* all instances with a relation different from $\mathcal{NA}$ (6590 instances in train partition and 2263 in test). Those belonging to $\mathcal{NA}$ are taken as *outliers* or *noisy* (1410 instances in train partition and 454 in test).

3.2 Baseline of representation learning methods

In this section it is defined the baseline of representation learning methods. These methods are the *encoder* function defined in the Equation 1.
Bag-of-words based methods In this approach it was used bag-of-words (BoW) and bag-of-characters (BoC) methods to represent the texts using a vector. In both cases it was used unigrams and bigrams. These functions were named

[6] avalaible in http://iesl.cs.umass.edu/riedel/ecml/

f_ bow_ 1 and *f_ boc_ 1* for unigrams. For the case of bigrams, the functions were named as *f_ bow_ 2* and *f_ boc_ 2*. Finally, the union of unigrams and bigrams were named *f_ bow_ 12* and *f_ boc_ 12*. For the comparisson, it were selected the 10 000 most frequent terms.

TF-IDF based methods The frequency-based methods (TF-IDF) of words and characters was used for representing texts as a vector. It was calculated the TF-IDF representation for unigrams and bigrams. These functions were named as *f_ tfidf_ 1* and *f_ tfidf_ char_ 1* for unigrams. For the case of bigrams, they were named as *f_ tfidf_ 2* and *f_ tfidf_ char_ 2*. Finally, the union of unigrams and bigrams was named as *f_ tfidf_ 12* and *f_ tfidf_ char_ 12*. As with BoW and BoC, it were selected the 10 000 most frequent terms.

Pretrained embeddings based methods Pretrained embeddings were used to obtain a vector from all text. The pretrained embeddings used in this work were RoBERTa [18], DAN [4], TRANSF [4] and LASER [1]. These functions were named as *f_ roberta*, *f_ dan*, *f_ transf* and *f_ laser* respectively.

3.3 Unsupervised representation learning methods

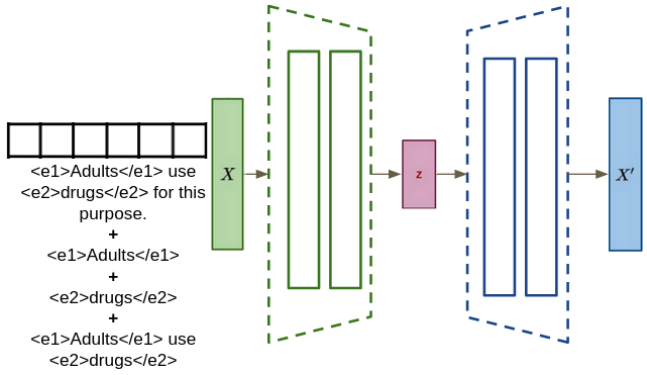
Inputs used in unsupervised methods For the input of these methods, one input representation is used and another is proposed. Both inputs use the functions *f_ dan*, *f_ transf*, *f_ laser* and *f_ roberta* as pretrained embeddings. The inputs *f_ dan*, *f_ transf* and *f_ laser* are used in [7], while *f_ roberta* is included in this work. The inputs are:

- original input: This input was proposed in [7]. It consists of the vector obtained with pretrained embeddings from all the text.
- improved input: This input is proposed in this research. The improved input consists of a concatenation of the vectors obtained from the entities, the text between the two entities including them, and the full text. As in the original input, these vectors are obtained with pretrained embeddings.

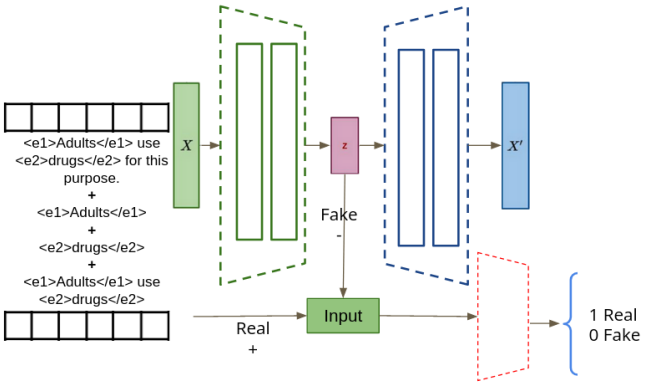
Autoencoder based methods The use of AE was proposed to obtain unsupervised text representations (see Figure 2(a)). This method is based on [7]. The architecture is composed of two dense layers in the encoder and the decoder, with 768 dense units and a ReLu-like activation function. These AE-based functions were named *f_ ae_ roberta*, *f_ ae_ dan*, *f_ ae_ transf* and *f_ ae_ laser* according to the function with which the input vectors are obtained.

Adversarial Autoencoders based methods AAE, as AE, was used to obtain unsupervised text representations (see Figure 2(b)). It was proposed as *encoder* AAE under the assumption that if an observation (s_i, r) is noisy, then the observation will not fit the distribution of the rest of the observations, and it will remain far away. As in the AE, this paper is based on the architecture proposed in [7]. The AAE's input is composed of two elements: the *instance representations* and *data distribution*. On the one hand, the instance representations are the vectors obtained from the $\mathcal{X}_{train}$ partition with original or improved input. On the other hand, *data distribution* is an essential element of the AAE. This

distribution is obtained from instances considered as real. They are represented by vectors obtained using an AE with the same architecture as the AAE encoder. The real instances are essential because the latent space z of *instance representations* is tried to fit the distribution obtained from them. These AE-based functions were named as *f_aae_roberta*, *f_aae_dan*, *f_aae_transf* and *f_aae_laser* according to the function with which the input vectors are obtained.



(a) AE with dense layers and improved input



(b) AAE with dense layers and improved input

Fig. 2: Architectures with dense layers and improved input.

In [7], the authors randomly selected one-third of the total of instances as real. In this work, three methods are proposed to select these instances.

- *Random*: It consists of selects randomly the 30% of the total number of instances as real. This method was proposed in [7].
- *Gaussian*: This method generates random instances fitted to the Gaussian distribution as real.

- *k-Means* clustering algorithm [19]: This algorithm was trained on the complete train partition to create 2 clusters. Then, those that belong to the cluster with the largest number of instances were selected as real instances. This decision is given because the number of noisy instances is generally a small percent of the total number of instances. The representation $\mathcal{V}$ used to train this algorithm was the functions *f_ roberta*, *f_ dan*, *f_ transf* and *f_ laser* output.
- *DBScan* [5]: This algorithm constructs the groups based on the density of the points. It only requires defining the number of points (*min_pts*) to consider a region as dense and the distance (*eps*) to consider the neighborhood of a point. The points considered real (real instances) are the cores of each built group. In this work, we try to get the number of core points between 10 and 50 of the total number of points.

3.4 Anomaly detection methods

Anomaly detection methods can be grouped into *proximity-based*, *linear model*, *ensembles* and *neural networks* among other categories [22]. The functions *noisy* of the *proximity-based* group used were Local Outlier Factor (*lof*) [3], k Nearest Neighbors (*knn*) [22] and Subspace Outlier Detection (*sod*) [15]. In addition, in the *linear model* group are Principal Component Analysis Outlier Detector (*pca_ od*) [27]. Also, Variational Autoencoder Outlier Detector (*vae_ od*) [14] belong to the *neural networks* group. Finally, Isolation Forest (*iforest*) [17] and Lightweight On-line Detector of Anomalies (*loda*) [23] are in group *ensembles*.

3.5 Experimental design

First experiment To determine the best $< encoder, noisy >$ pair for each group of representation learning methods, it were performed 5 iterations. The 10 best *encoder* functions with their associated *noisy* functions were chosen from these results. Then, the performance of these functions was evaluated based on 20 replications of each of these pairs $< encoder, noisy >$. The number of replications (sample size) was determined using ANOVA One Way test for a desired significance level of 0.05, statistical power of $\beta = 0.95$ and assuming an effect size of F distribution $= 0.4$. From the results of the replications, the ANOVA One Way test is applied to know if there are significant differences between the results achieved by the pairs. Finally, if there were significant differences, pairwise comparisons were made to observe which pair showed differences. The two-by-two comparisons were made with t-test and Holm Correction [10]. The significance threshold was set at $\alpha = 5\%$. Figure 3 summarizes the above experimental design.

Second experiment In addition, it was analyzed the performance of the original input proposed in [7] with respect to the improved input proposed in this work. To analyze the performance, the 10 best $< encoder, noisy >$ pairs were considered, and run 5 replications with each input. Then, it was analyzed if there were variations in the ranking for each replication with Friedman test.

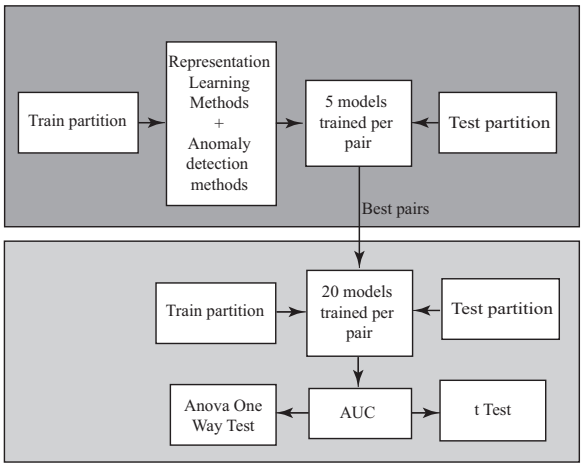


Fig. 3: Methodology followed in the current experiment.

Third experiment Finally, the performance of the four methods was evaluated to select the instances considered as real in the AAE. To do so, all results with the same method were pooled to select without considering the pretrained embeddings with which the input vectors are obtained. The *noisy* functions are also not taken into account.

4 Experiments and evaluation

First experiment The first 10 $< encoder, noisy >$ pairs with highest AUC after 5 iterations in the *semeval2010* dataset were selected. Table 1 summarizes the AUC of this 10 best $< encoder, noisy >$ pairs after 20 iterations in the *semeval2010*. Significant differences were found between all pairs (ANOVA: $F(9, 190) = 24.34, p < 2e^{-16}$). In the case of pairwise comparisons with t-test, the pair $< f\_ae\_transf, knn >$ presents significant differences with the rest, except with $< f\_ae\_roberta, lof >$ and $< f\_ae\_dan, knn >$.

The obtained results indicate that architectures based on AE and AAE combined with anomaly detection methods obtain the highest AUC. Eigth of the 10 best pairs have an *encoder* functions based on AE and AAE. Only the functions *f_tfidf_12* and *f_dan* are not based on these architectures. Among these first pairs, architectures *f_ae_transf* and *f_ae_dan* stand out, appearing twice each with different anomaly detection methods. Also, the architecture *f_aae_roberta* with the instance selection methods *gaussian*, *kmeans* and *random* are present. This suggests that the AAE is able to adjust the latent space z independently of the instances considered as real. Among the anomaly detection methods, *lof* appears the half of the time.

Second experiment Table 2 summarizes the AUC with the improved input and original input in the *semeval2010* dataset. The *encoder* functions *f_tfidf_12* and

Table 1: AUC of the 10 best $< encoder, noisy >$ pairs using the *semeval2010* dataset after 20 iterations.

encoder	*noisy*	AUC
f_ae_transf	knn	0.580 ± 0.007
f_ae_roberta	lof	0.576 ± 0.005
f_ae_dan	knn	0.573 ± 0.011
f_aae_roberta (gaussian)	lof	0.561 ± 0.021
f_ae_transf	sod	0.555 ± 0.012
f_ae_dan	sod	0.554 ± 0.014
f_tfidf_12	lof	0.552 ± 0.000
f_aae_roberta (kmeans)	lof	0.551 ± 0.009
f_aae_roberta (random)	lof	0.549 ± 0.011
f_dan	knn	0.548 ± 0.000

f_dan do not depend on the previous inputs. However, it was decided to keep them in the analysis as baseline since their results should not vary considerably. In this way, it can be observed how the other functions behave with respect to these 2 functions. All methods increased their AUC with the improved input relative to their performance with the original input. Further, and more critically here, the order of the methods in terms of their performance varied significantly (Friedman: $\chi^2(2) = 108.37$, $p < 2.2e^{-16}$). This confirms that the addition of the elements to the input of the AE and AAE, proposed in this work, increased the AUC concerning the input used in [7].

Table 2: AUC of the ROC curves after 5 replications with improved input and original input.

improved input			original input		
encoder	*noisy*	AUC	*encoder*	*noisy*	AUC
f_ae_transf	knn	0.581 ± 0.009	*f_tfidf_12*	lof	0.552 ± 0.000
f_ae_roberta	lof	0.577 ± 0.006	*f_dan*	knn	0.548 ± 0.000
f_aae_roberta (kmeans)	lof	0.560 ± 0.013	*f_ae_transf*	sod	0.544 ± 0.011
f_ae_dan	knn	0.559 ± 0.003	*f_ae_dan*	knn	0.542 ± 0.007
f_aae_roberta (random)	lof	0.555 ± 0.014	*f_ae_transf*	knn	0.541 ± 0.007
f_tfidf_12	lof	0.552 ± 0.000	*f_ae_dan*	sod	0.532 ± 0.015
f_aae_roberta (gaussian)	lof	0.552 ± 0.042	*f_aae_roberta (kmeans)*	lof	0.529 ± 0.013
f_ae_dan	sod	0.551 ± 0.007	*f_ae_roberta*	lof	0.524 ± 0.002
f_dan	knn	0.548 ± 0.000	*f_aae_roberta (random)*	lof	0.522 ± 0.005
f_ae_transf	sod	0.547 ± 0.016	*f_aae_roberta (gaussian)*	lof	0.515 ± 0.006

Third experiment Table 3 shows the AUC values of the AAE architecture considering the 4 methods to select the instances considered as real. It was found significant differences between all ways to select (ANOVA: $F(3, 556) = 7.52, p < 6.12e^{-05}$). In the case of pairwise comparisons with t-test, all methods have significant differences with the *random* method. The obtained results indicate

that the three methods for selecting real instances proposed in this paper improve the *random* selection method reported in [7].

Table 3: Performance of the methods to select the instances considered as real after 5 iterations.

AAE with method to select	AUC
aae_gaussian	0.510 ± 0.023
aae_dbscan	0.506 ± 0.023
aae_kmeans	0.506 ± 0.027
aae_random	0.496 ± 0.032

5 Conclusions

In this paper it was evaluated the noise reduction performance of several methods to obtain unsupervised text representations. The best representation for noise reduction are obtained with the AE and AAE architectures using the improved input proposed in this work. The input consists of a concatenation of the entity vectors, the text between the two entities including them and the full text. Moreover, using the improved input results in higher AUC values compared to using the *original* input. The obtained results confirm the importance of adding more information as input to the AE and AAE.

The obtained results demonstrate that the three methods to select the instances considered as real that are proposed in this work improve significantly the AUC of the random selection method. However, no significant differences were found between the three methods. Because of this, it is considered that the *gaussian* method may be a good choice considering its low computational cost compared to the other two.

As future work, the next step is proposing an approach that improves the representations by adding more input information. Also it is worth to investigate how to combine the input information. Finally, use of other types of layers such as long short-term memory (LSTM) will be evaluated.

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

References

1. Artetxe, M., Schwenk, H.: Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. Transactions of the Association for Computational Linguistics **7**, 597–610 (2019)

2. Bastos, A., Nadgeri, A., Singh, K., Mulang', I.O., Shekarpour, S., Hoffart, J., Kaul, M.: RECON: Relation Extraction using Knowledge Graph Context in a Graph Neural Network. In: Proceedings of the Web Conference 2021. pp. 1673–1685 (2020)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of Data. pp. 93–104 (2000)
4. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Sung, Y.H., Strope, B., Kurzweil, R.: Universal Sentence Encoder. arXiv:1803.11175v2 [cs.CL] p. 7 (2018)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of Second International Conference on Knowledge Discovery and Data Mining (KDD-96) **96**(34), 226–231 (1996)
6. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam: Open information extraction: The second generation. International Joint Conferences on Artificial Intelligence **11**, 3–10 (2011)
7. García-Mendoza, J.L., Villaseñor-Pineda, L., Orihuela-Espina, F., Bustio-Martínez, L.: An autoencoder-based representation for noise reduction in distant supervision of relation extraction. Journal of Intelligent & Fuzzy Systems **42**(5), 4523–4529 (mar 2022)
8. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the 14th Conference on Computational Linguistics (COLING-92). pp. 539–545 (1992)
9. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D.Ó., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. In: Proceedings ofthe 5th International Workshop on Semantic Evaluation, ACL 2010. pp. 94–99 (2010)
10. Holm, S.: A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics **6**(2), 65–70 (1979)
11. Jat, S., Khandelwal, S., Talukdar, P.: Improving Distantly Supervised Relation Extraction using Word and Entity Based Attention. In: 6th Workshop on Automated Knowledge Base Construction, AKBC@NIPS 2017 (2017)
12. Ji, G., Liu, K., He, S., Zhao, J.: 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). pp. 3060–3066 (2017)
13. Kim, J.T., Moldovan, D.I.: Acquisition of semantic patterns for information extraction from corpora. In: Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications. pp. 171–176 (1993)
14. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: International Conference on Learning Representations (ICLR) (2014)
15. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 831–838 (2009)
16. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting ofthe Association for Computational Linguistics. pp. 2124–2133 (2016)
17. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation Forest. ICDM'08. Eighth IEEE International Conference on Data Mining pp. 413–422 (2008)
18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv (1907.11692) (2019)

19. Lloyd, S.P.: Least Squares Quantization in PCM. IEEE Transactions on Information Theory **28**(2), 129–137 (1982)
20. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial Autoencoders. In: International Conference on Learning Representations (ICLR) Workshop (2016)
21. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the 47th Annual Meeting of the ACL. pp. 1003–1011 (2009)
22. Pang, G., Shen, C., Cao, L., Van den Hengel, A.: Deep Learning for Anomaly Detection: A Review. ACM Computing Surveys **1**(1), 3569–3570 (2020)
23. Pevný, T.: Loda: Lightweight on-line detector of anomalies. Machine Learning **102**(2), 275–304 (feb 2016)
24. Piskorski, J., Yangarber, R.: Information extraction: Past, Present and Future. In: Multi-source, Multilingual Information Extraction and Summarization 11, pp. 23–49 (2013)
25. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 148–163 (2010)
26. Ru, C., Tang, J., Li, S., Xie, S., Wang, T.: Using semantic similarity to reduce wrong labels in distant supervision for relation extraction. Information Processing & Management **54**(4), 593–608 (jul 2018)
27. Shyu, M.L., Chen, S.C., Sarinnapakorn, K., Chang, L.: A Novel Anomaly Detection Scheme Based on Principal Component Classifier. Tech. rep. (2003)
28. Smirnova, A., Cudré-Mauroux, P.: Relation Extraction Using Distant Supervision: A Survey. ACM Computing Surveys **51**(5), 1–35 (nov 2018)
29. Soderland, S.: Learning Information Extraction Rules for Semi-Structured and Free Text. Machine Learning **34**, 233–272 (1999)
30. Takamatsu, S., Sato, I., Nakagawa, H.: Reducing wrong labels in distant supervision for relation extraction. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. pp. 721–729 (2012)
31. Vashishth, S., Joshi, R., Prayaga, S.S., Bhattacharyya, C., Talukdar, P.: Reside: Improving Distantly-Supervised Neural Relation Extraction using Side Information. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1257–1266 (2018)
32. Ye, Z.X., Ling, Z.H.: 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: Human Language Technologies. pp. 2810–2819 (2019)
33. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1753–1762 (2015)
34. Zhou, P., Xu, J., Qi, Z., Bao, H., Chen, Z., Xu, B.: Distant supervision for relation extraction with hierarchical selective attention. Neural Networks **108**, 240–247 (2018)