Improving Performance of Relation Extraction Algorithm via Leveled Adversarial PCNN and Database Expansion

Diyah Puspitiningrum
Department of Computer Science
University of Bengkulu
Bengkulu, Indonesia
diyahpuspitiningrum@gmail.com

Abstract—This study introduces database expansion using the Minimum Description Length (MDL) algorithm to expand the database for better relation extraction. Different from other previous relation extraction researches, our method improves system performance by expanding data. The goal of database expansion, together with a robust deep learning classifier, is to diminish wrong labels due to the incomplete or not found nature of relation instances in the relation database (e.g., Freebase). The study uses a deep learning method (Piecewise Convolutional Neural Network or PCNN) as the base classifier of our proposed approach: the leveled adversarial attention neural networks (LATTADV-ATT). In the database expansion process, the semantic entity identification is used to enlarge new instances using the most similar itemsets of the most common patterns of the data to get its pairs of entities. About the deep learning method, the use of attention of selective sentences in PCNN can reduce noisy sentences. Also, the use of adversarial perturbation training is useful to improve the robustness of system performance. The performance even further is improved using a combination of leveled strategy and database expansion. There are two issues: 1) database expansion method: rule generation by allowing step sizes on selected strong semantic of most similar itemsets with aims to find entity pair for generating instances, 2) a better classifier model for relation extraction. Experimental result has shown that the use of the database expansion is beneficial. The MDL database expansion helps improvements in all methods compared to the unexpanded method. The LATTADV-ATT performs as a good classifier with high precision P@100=0.842 (at no expansion). It is even better while implemented on the expansion data with P@100=0.891 (at expansion factor k=7).

Keywords—relation extraction, database expansion, MDL, PCNN, classification

I. INTRODUCTION

The relation extraction task can be in an unsupervised or supervised domain. In terms of supervised, relation extraction must classify an entity pair to a set of trained relations using documents that contain the entity pair. A binary classification relation extraction system predicts whether or not a given document contains any relation for the entity pair. In a multi-class classification relation extraction system, the system classifies the ontology of a relation of the given document.

The supervised techniques for machine learning require a large amount of training data for learning. Since hand-annotated datasets consume time, distant supervision [6][9][3][12] produces a large amount of training data by aligning knowledge base facts with text. More sophisticated models in recent years have primarily focused on extracting structured information from plain unstructured text using deep learning methods. The use of PCNN in relation extraction has been topics in recent researches [4][5][15][16].

This study proposes a sentence-level of leveled adversarial attention on PCNN for relation extraction tasks on multi-class classification with the help of database expansion using association rules mining algorithm based on the Minimum Description Length principle (Krimp algorithm). The algorithm itself produces an excellent representative model of a database since the Krimp mining only the most frequent/interesting patterns that compress the database best. Many researchers have investigated Krimp implementation in many applications, such as in clustering, classification, and recommender system (see [14][13][7][8]). In the Information Retrieval area, for the relation extraction problem, the Krimp algorithm can be used to grab only the interesting itemsets taken from the database summary called code table. Then PCNN LATTADV-ATT (or shortly LATTADV-ATT) uses those itemsets to capture the possible semantic relationship between pair of entities in every itemset. Itemsets are enlarged using k topmost similar patterns/itemsets in code table using Jaccard or Cosine similarity until all pairs of entities found.

Our contributions are two folds:

• To demonstrate the use of database expansion through MDL based semantic identification as a novel preprocessing technique in relation extraction task;
• To propose LATTADV, a novel independent deep learning framework for relation extraction classifier. By independent means that the model also can be implemented on any classification tasks.

II. METHODOLOGY

In this section, we first started the problem definition and hypothesis. Then we introduce the preprocessing method (the database expansion method) used in our experiment and the architecture of the LATTADV classifier model.

A. Problem Definition

Given a dataset of text sentences with entity pair, \( X = \{x_1, x_2, \ldots, x_n\} \), and given \( R \) is relation labels, our model measures the probability of each relation \( r \), where \( r \in R \).

B. Hypothesis

Our hypotheses are: 1) The MDL database expansion is beneficial to improve the classifier system performance. It is because the MDL can capture very well the association between the most common go together pattern itemsets of all the essential and the most frequent words in \( X \). The merged of the most similar association itemsets together with entity identification in the expansion process helps in creating new
instances that meet the grammatical/semantic rules. 2) The use of leveled architecture is beneficial to improve the performance of a classifier.

C. Preprocessing: The MDL Database Expansion Method

As the dataset, we use pre-trained words learned from the New York Times (NYT) corpus 2005-2006 (https://catalog.ldc.upenn.edu/LDC2008T19). The dataset previously used by many researchers in the field of relation extraction [9][3][12][16]. The entities of the dataset were annotated with Stanford NER (https://nlp.stanford.edu/software/CRF-NER.html) and linked to Freebase (www.freebase.com). As the initial/first embedding words were obtained using word2vec (https://code.google.com/archive/p/word2vec/).

As there are many NA values in the dataset, the relation extraction surely is not a trivial task. It is not only about compares the extracted relation instances against Freebase relations automatically but also how to capture and generate the k topmost semantic relations between pairs of entities to produce the new expanding sentences. The role of database expansion is to diminish wrong class labels of relation through a semantic understanding of the entity arrangement. Fig. 1 shows the schema of the LATTADV-ATT relation extraction system with database expansion.

![Fig. 1. Overview schema of the leveled adversarial PCNN relation extraction system with database expansion](image1)

Following is the procedure to expand the database. Given X (see Section 2.1), KRIMP, a state-of-the-art algorithm in MDL, will produce a summary of X called code table (or CT). The CT is a collection of items. An itemset here denotes the most common words in a dataset of X that usually go together. The KRIMP algorithm [14] works as an association rule algorithm, but it only accepts patterns that help compress the dataset X optimally. The goal of MDL is to find CT that minimizes the compressed MDL size of the database:

\[
L(D, CT) = L(DCT) + L(CTD) \tag{1}
\]

Assume a transaction denotes a set of important words that represent a text sentence \(x\). The \(L(DCT)\) denotes the size of the encoded database when the CT encodes it, viz. the sum of encoded lengths of all transactions in \(X\). The \(L(CTD)\) denotes the size of encoded CT, viz. a sum of the addition of all sizes of code in CT and their length of encoded itemsets. The size of a code denotes the length of the code in standard CT (individual item). The length of the encoded itemset denotes the length of the encoded itemset throughout the CT. Candidate itemset contains a set of words that frequently go together. It is only inserted into CT if it produces good compression; otherwise, the KRIMP will reject the candidate itemset. All the relative frequencies and the compression results also evaluated. MDL only accept CT that minimize \(L(D, CT)\) best. See Fig. 2.

We propose a database expansion using itemsets in a code table. Different from Widened KRIMP for diverse parallelism [10] that maximizes the sum of minimum distances between pairs of members of the selected subset in CT, in our proposed method, the focus is in the semantic of entity pair. Start from a base itemset; we expand data by combining \(k\)-topmost itemsets rank by similarity score. Expansion is accepted only if until \(k\)-topmost itemsets, at least there is one entity pair is presence. A base itemset starts from the first tuple to the last tuple in CT. Every tuple in CT denotes a set of most frequent words that usually go together. Lower minimum support set up in CT leads to more itemsets available in CT.

Consider a set of patterns in CT, \(F = \{F_1, F_2, ..., F_i\}\), the expanded pattern generated from \(k\)-topmost similar itemsets defined by:

\[
F_i^* = \bigcup_{j=1}^{k} \min_{l \leq j \neq l} F_i \tag{2}
\]

Where \(F_i, F_j \in F\) and \(\Delta\) denotes text similarity technique, e.g., cosine similarity or Jaccard similarity. The \(l\) denotes the number of tuples in CT.

There are three kinds of situations about frequent itemsets that KRIMP generates from a sentence:

- **Ideal**: The ideal itemset of \(F_i^*\) is a situation where an itemset (a collection of essential keywords of a sentence) has an entity pair inside. If the situation found, the itemset is appended to the database to expand the existing data.

- **Half ideal**: Another situation is the half ideal itemset. This kind of situation is when a pair of entities is found only a half in \(F_i^*\), thus creating an incomplete pair. This situation needs an expansion process of \(k\)-topmost similar pattern generation until \(F_i^*\) has at least one entity pair inside. If there is no entity pair found in \(F_i^*\) after the combination of \(k\)-topmost similar itemset, then \(F_i^*\) is skipped away, and the base itemset is set up with the next item set.

- **Not ideal**: This is the situation when no entity found in the itemset. We skip this type of itemset since the expansion method never uses a not ideal code as a base itemset.

In our research, the expansion process focus on three entities: PERSON, LOCATION, and ORGANIZATION. After detecting the availability of the three entities in an itemset in CT (or in a sentence), the possible semantic relations are as follow:

- **"/business/company/location":** when an entity of ORGANIZATION and entity of LOCATION found in the same sentence. The newly generated instance has a pattern of "X has location Y." Where X is an ORGANIZATION entity, Y is a LOCATION entity.

- **"/people/person/place_lived":** when an entity of PERSON and entity of LOCATION found in the same sentence. The
newly generated instance has a pattern of "X has location Y." Where X is a PERSON entity, Y is a LOCATION entity.

*/business/person/company": when an entity of PERSON and entity of ORGANIZATION found in the same sentence. The newly generated instance has a pattern of "X has organization Y." Where X is a PERSON entity, Y is an ORGANIZATION entity.

2) Main Components: Three main components of LATTADV:

a) Sentence encoder: Convolutional Neural Network (CNN) is used to build distributed representation of sentence $x_i$, $i = \{1, 2, ..., n\}$.

![Architecture of LATTADV-ATT](attachment:image.png)

b) Selective attention (ATT): When distributed vector representations of $X$ learned, ATT only gives attention to sentence $s_i$ that sincerely express relation $r$ through computing the weighted average of sentences $s_1, s_2, ..., s_n$ with $s_i$ defined as:

$$s_i = \sum \alpha_i^r s_i$$

The softmax function is used to compute the embedding vector of the query vector $q_r$. The $\alpha$ denotes the attention weights with relation $r$, $\alpha^r = \text{softmax}(\text{tanh}(s_i)^T q_r))$, see [4]. ATT de-emphasizes noisy sentence. ATT helps to avoid massive false labeling during training $\sum L_{ATT}(X, \theta)$ and testing. Final loss function for ATT[15]:

$$L_{ATT}(X, \theta) = -\sum_{i=1}^{K} \log P(r \mid x, \theta)$$

The $\theta$ denotes parameters, and the $K$ denotes the total number of predefined relation labels $r$. The conditional probability of relation:

$$P(r \mid x, \theta) = \text{softmax}(A s_i + b)$$

Where $A$ denotes a weighted diagonal representation matrix of relations, $s_i$ denotes attended sentence, and $b$ denotes a bias vector.
c) Adversarial training (ADV): Adversarial training first introduced in [2]. In ADV, a small amount of perturbation, $e_{adv}$, is added on word embedding after that adversarial training computes the gradient direction of a loss function with aims to maintain closeness and linearity to input data (viz. by linearizing loss function near the input $X$). Given word embedding of all the words in $X$ called as $V$, $V = \{v_1, v_2, \ldots, v_n\}$, ADV adds $e_{adv}$ to $V$. The final loss function for ATT-ADV is defined as follows [15]:

$$L_{ATT,ADV}(X, \theta) = L_{ATT}(X + e_{adv}, \theta)$$

where:

$$e_{adv} = g/\|g\|$$

and

$$g = \nabla_{\theta} L_{ATT}(X, \hat{\theta})$$

The $\theta$ denotes all the parameters. The $\|g\|$ denotes the norm of gradients over all the words from all the sentences in $X$.

### III. EXPERIMENTS

In this section, we first introduce the dataset we used in our experiment. Then we introduce evaluation metrics and the baselines. After that, we demonstrate the capability of LATTADV as a classifier. Then we discuss the effect of database expansion utilization on the performance of the relation extraction system.

In our LATTADV model, both in ATT and MAX version, for similarity measure, we use cosine and Jaccard similarity, respectively. We tune our experiments with the following parameter settings: sentence embedding size 230, position embedding dimension 5, word embedding size 50, sliding window size 3, maximum of training epochs 60, learning rate 0.5, batch size 160, dropout rate 0.5, weight decay 0.00001, and maximum of relations 53.

### TABLE I. STATISTICS OF THE USED NYT DATASET

| Training set | $k$ | #Rel | #Ent-Pair | # Mention | #Sentence |
|--------------|-----|------|-----------|-----------|-----------|
| base         | 0   | 53   | 172399    | 293162    | 570988    |
| cos          | 3   | 53   | 172499    | 293170    | 570988    |
| jae          | 3   | 53   | 172499    | 293170    | 570988    |
| cos          | 7   | 53   | 173099    | 293187    | 572990    |
| jae          | 7   | 53   | 173099    | 293185    | 572988    |
| cos          | 10  | 53   | 173299    | 293185    | 573649    |
| jae          | 10  | 53   | 173299    | 293185    | 573649    |

| Testing set  | $k$ | #Rel | #Mention | #Sentence |
|--------------|-----|------|----------|-----------|
| base         | 0   | 53   | 1950     | 17248     |
| cos          | 3   | 53   | 1950     | 17252     |
| jae          | 3   | 53   | 1950     | 17252     |
| cos          | 7   | 53   | 1950     | 173175    |
| jae          | 7   | 53   | 1950     | 173175    |
| cos          | 10  | 53   | 1950     | 173341    |
| jae          | 10  | 53   | 1950     | 173341    |

Word size=50, position size=3, hidden size=230.
#Rel: number of relation.
#Ent-Pair: number of entity pair.
cos: NYT dataset + expansion with k topmost similar itemsets by Cosine similarity.

### TABLE II. STATISTICS OF THE USED NYT DATASET

| Method                | k  | AUC | Max F1 | P@100 | P@200 | P@300 | Mean |
|-----------------------|----|-----|--------|-------|-------|-------|------|
| LATTADV-ATT           | 0  | 0.418 | 0.456  | 0.842 | 0.801 | 0.761 | 0.801|
| ATT-ADV               | 0  | 0.412 | 0.441  | 0.860 | 0.786 | 0.754 | 0.801|
| LATTADV-MAX           | 0  | 0.409 | 0.452  | 0.782 | 0.761 | 0.741 | 0.761|
| MAX-ADV               | 0  | 0.402 | 0.447  | 0.802 | 0.776 | 0.741 | 0.773|
| ATT                   | 0  | 0.399 | 0.448  | 0.752 | 0.726 | 0.724 | 0.734|
| ATT_tanh              | 0  | 0.395 | 0.443  | 0.822 | 0.781 | 0.741 | 0.781|
| RNN-MAX               | 0  | 0.395 | 0.441  | 0.762 | 0.756 | 0.734 | 0.751|

LATTADV-ATT: PCNN + leveled attention and adversarial attention
ATT-ADV: PCNN + attention + adversarial
LATTADV-MAX: PCNN + leveled attention and adversarial + maximum function
MAX-ADV: PCNN + maximum function + adversarial
ATT: PCNN with attention
ATT_tanh: PCNN with attention + tanh activation function
RNN-MAX: recurrent neural network + maximum function

### A. Dataset

The number of the corpus in the NYT dataset (Section 2.3) is 1.8 million news articles and accommodates 53 relation labels. For vocabulary, we use words with frequency $\geq 100$.

In our experiment, the NYT dataset was expanded and split into six datasets as expansion results using Cosine and Jaccard similarity, respectively, with $k=[3,7,10]$. The training and testing datasets were generated randomly using cross-validation with a ratio of 80:20. Table I shows the statistics of the NYT extend datasets. Different similarity metrics result in different pattern extensions, even if they have the same $k$ factor.

### B. Evaluation Metrics and Baselines

We measure AUC, max F1, precision-recall, and precision scores at P@100, P@200, P@300 to investigate the performance of all PCNN models with various architectures. Also, the Borda voting score is used to rank the relation extraction method. The higher the score, the better the method. We compare our LATTADV-ATT to several state-of-the-art deep learning methods such as Recurrent based Neural Network (e.g., RNN-MAX) and many PCNN based neural networks, either single-stage or leveled. PCNN ATT [4], PCNN ATT-ADV, and PCNN MAX-ADV [15] is to mention several (see Table II and Table III for other benchmark methods).

![Fig. 6. Architecture of LATTADV-ATT](image-url)
**C. Model as a Classifier**

Table II and Fig. 6 show the performance of LATTADV-ATT on the original NYT dataset. Figure 6 refers to Table II. It shows that LATTADV-ATT outperforms all other baselines, including the recurrent neural networks. Also, the use of attention selector yields a better result than the maximum selector. Its best performance means that the LATTADV-ATT method can perform as a good classifier.

**D. Effect of Number of Relations in Database Expansion**

![Table III. Results, ordered by Borda voting score](image)

Table III and Fig. 7 depicts a list of methods in decrease order by Borda voting methods of all used metrics (AUC, max F1, P@100, P@200, P@300, and mean of precision). Figure 7 refers to Table III. Table III shows that a careful step of expansion factor k can have a significant effect on the improvement of the relation extraction system. Larger k means a more quantity of relations generated in the expanded database (Table I). To obtain better performance, we must test first the appropriate k value carefully because the generated merging itemsets of CT (the F*) may too far away in meaning from the original sentences meaning if the k is too big. As an example, the expansion factor k=10 can lead to a long pattern bias of F*. This situation is because the LATTADV-ATT keeps combining the base itemset with all the most similar itemsets found in the CT until at most k itemsets. The best improvement is achieved using LATTADV-ATT with Jaccard similarity expansion at k=7 (Table III and Fig. 7).

**E. Effect of The Deep Learning Strategy Setup and The Similarity Metric Used**

From Table III, we have proven that the use of the leveled strategy, as well as the attention and the adversarial training in LATTADV-ATT, demonstrates significant improvement in relation extraction system performance, especially against the best state-of-the-art (PCNN ATT-ADV). The LATTADV-ATT outperforms all other methods in all metrics of AUC, maximum F1, and precisions (Table III). From the AUC, the NYT dataset used is imbalanced. Similar to the AUC, the F1 score of the LATTADV-ATT also shows improvement. (The F1 score measurement is to keep a balance between Precision and Recall.) The LATTADV-ATT also improves the precision scores very well. Compared to PCNN ATT-ADV as the best baseline of state-of-the-art methods used, the percentages of the LATTADV-ATT Jaccard k=7 improvement are as follows: AUC=2.43%, maximum of F1=2.94%, P@100=3.48%, P@200=6.44%, P@300=2.25%, and mean of precisions=2.12%. Also, when compared to the attention network without expansion (PCNN ATT k=0), the percentages of the LATTADV-ATT Jaccard k=7 improvement are as follows: AUC=5.76%, maximum of F1=1.56%, P@100=18.33%, P@200=8.95%, P@300=6.49%, and mean of precisions=11.44%. Therefore the use of the database expansion can improve the relation extraction system’s performance. Cosine or Jaccard similarity can be used substitutive.

**IV. CONCLUSIONS**

In this paper, we develop a new method for relation extraction task using database expansion and leveled piecewise convolutional neural networks with attention and adversarial training network (LATTADV-ATT). The proposed method can make full use of all informative sentences and serve as an excellent relation classifier on the embedding sentences. Its database expansion feature enlarges the database by generating new sentences that meet the semantic meaning between identified pairs of entities. Experimental results show that: 1) The MDL expansion is beneficial to improve the performance of the classifier model. This ability is because the minimum description length-based algorithm can sharply capture associations, and the semantic understanding of entity pair helps in generate new expansion sentences. 2) The leveled network with attention and adversarial training strategy in the proposed strategy is beneficial to improve the performance of the PCNN-based relation extraction classifier.

**REFERENCES**

[1] C. Nogueira dos Santos, B. Xiang, and B. Zhou, “Classifying Relations by Ranking with Convolutional Neural Networks,” in ACL (1), 2015. [The 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, pp.626–634, 2015].

[2] J.I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” CoRR abs/1412.6572, 2014.

[3] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D.S. Weld, “Knowledge-Based Weak Supervision for Information Extraction of
Overlapping Relations,” in HLT’11(1). Association for Computational Linguistics, Stroudsburg, PA, USA, pp.541–550, 2011.

[4] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, “Neural Relation Extraction with Selective Attention over Instances,” in Proceedings of the 54th Annual Meeting of ACL(Vol. 1: Long Papers). Association for Computational Linguistics, pp.2124–2133, 2016.

[5] T. Liu, K. Wang, B. Chang, and Z. Sui, “A Soft-label Method for Noise-tolerant Distantly Supervised Relation Extraction,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp.1790–1795, 2017.

[6] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant Supervision for Relation Extraction Without Labeled Data,” in Proceedings of ACL’09: Vol.2-Vol.2 (ACL’09). Association for Computational Linguistics, Stroudsburg, PA, USA, pp.1003–1011, 2009.

[7] D. Puspitaningrum, “Patterns, Models, and Queries,” Ph.D. Dissertation. Department of Information and Computing Sciences, Utrecht University, 2012.

[8] D. Puspitaningrum, Fauzi, B. Susilo, J.A. Pagua, A. Erlansari, D. Andreswari, R. Efendi, and I. S. W. B. Prasetya, “An MDL-Based Frequent Itemset Hierarchical Clustering Technique to Improve Query Search Results of an Individual Search Engine,” in Information Retrieval Technology, G. Zuccon, S. Geva, H. Joho, F. Scholer, A. Sun, and P. Zhang (Eds.). Springer International Publishing, Cham, pp.279–291, 2015.

[9] S. Riedel, L. Yao, and A. McCallum, “Modeling Relations and Their Mentions Without Labeled Text,” in Proceedings of the 2010 ECML PKDD: Part III (ECML PKDD’10), Berlin, Heidelberg: Springer-Verlag, pp.148–163, 2010.

[10] O. Sampson and M.R. Berthold, “Widened KRIMP: Better Performance through Diverse Parallelism,” in Advances in Intelligent Data Analysis XII, Hendrik Blockeel, Matthijs van Leeuwen, and Veronica Vinciotti (Eds.), Switzerland: Springer, pp.276–285, 2014.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” in Journal of Machine Learning Research, 15, 1, pp.1929–1958, Jan 2014.

[12] M. Surdeanu, J. Tibshirani, R. Nallapati, and C.D. Manning, “Multi-instance Multi-label Learning for Relation Extraction,” in Proceedings of the EMNLP-CoNLL’12. Stroudsburg, PA, USA: Association for Computational Linguistics, pp.455–465, 2012.

[13] M. van Leeuwen and D. Puspitaningrum, “Improving Tag Recommendation Using Few Associations,” in Advances in Intelligent Data Analysis XI, Jaakko Hollmén, Frank Klawonn, and Allan Tucker (Eds.). Berlin, Heidelberg: Springer, pp.184–194, 2012.

[14] M. van Leeuwen, J. Vreeken, and A. Siebes, “Compression Picks Item Sets That Matter,” in Knowledge Discovery in Databases: PKDD 2006, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.). Berlin, Heidelberg: Springer, pp.585–592, 2006.

[15] Y. Wu, D. Batmunkh, and S. Russell, “Adversarial Training for Relation Extraction,” in Proceedings of the 2017 Conference on Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp.1778–1783, 2017.

[16] D. Zeng, K. Liu, Y. Chen, and J. Zhao, “Distant Supervision for Relation Extraction via Piecewise Convolutinal Neural Networks,” in EMNLP, L. Marquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds. Association for Computational Linguistics, pp.1753–1762, 2015.