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

Diyah Puspitaningrum

Indonesia

diyahpuspitaningrum@gmail.com

http://www.diyahpuspitaningrum.net

The 7th International Conference on Cyber and IT Service Management
(*CITSM 2019*)

Improving Performance of Relation Extraction Algorithm via Leveled Adversarial PCNN and Database Expansion (1)

- Expanding database for better relation extraction by expanding data.
- PCNN + Database Expansion ----> to diminish wrong labels due to the incomplete/NA relation instances.
- We introduced the leveled adversarial attention neural networks (LATTADV-ATT).
- We introduce MDL-based database expansion to enlarge new instances using the most similar itemsets of the most common patterns of entity pairs.
- The attention of selective sentences in PCNN is to reduce noisy sentences.
- The use of adversarial perturbation training as a tool to improve the robustness of system performance.

Improving Performance of Relation Extraction Algorithm via Leveled Adversarial PCNN and Database Expansion (2)

- There are two issues:
- 1) rule generation in database expansion method; This is to find entity pair for generating instances,
- 2) advanced improvement of the classifier by using leveled strategy relation extraction.
- Experimental result has shown that the use of the database expansion is beneficial: P@100=0.842 (at no expansion), P@100=0.891 (at expansion factor k=7).

Contributions

- 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.

Methodology

- Given a dataset of text sentences with entity pair, $X = \{x1, x2, \ldots, xn\}$, and given R is relation labels, our model measures the probability of each relation r, where $r \in R$.

Hypotheses

- Our hypotheses are:

- 1) The MDL database expansion is beneficial to improve the classifier system performance.
- Preliminary Facts: the MDL can capture very well most frequent itemsets.
- The merged results must meet the grammatical/semantic rules.

- 2) The use of leveled sytrategy is beneficial to improve the performance of a classifier

Preprocessing:
The MDL Database Expansion Method

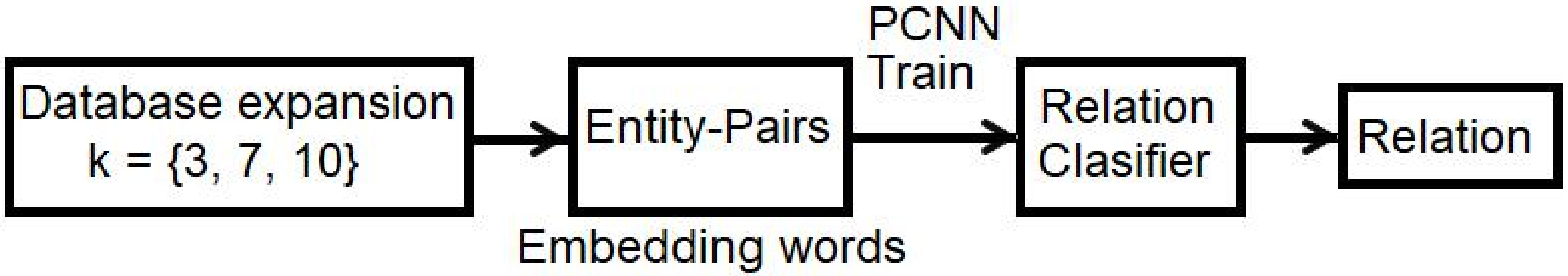


Fig. 1. Overview schema of the leveled adversarial PCNN relation extraction system with database expansion

- Pre-trained words learned from the New York Times (NYT) corpus 2005-2006.
- 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 embedding words were obtained using word2vec (https://code.google.com/archive/p/word2vec/).

Preprocessing: The MDL Database Expansion Method (2)

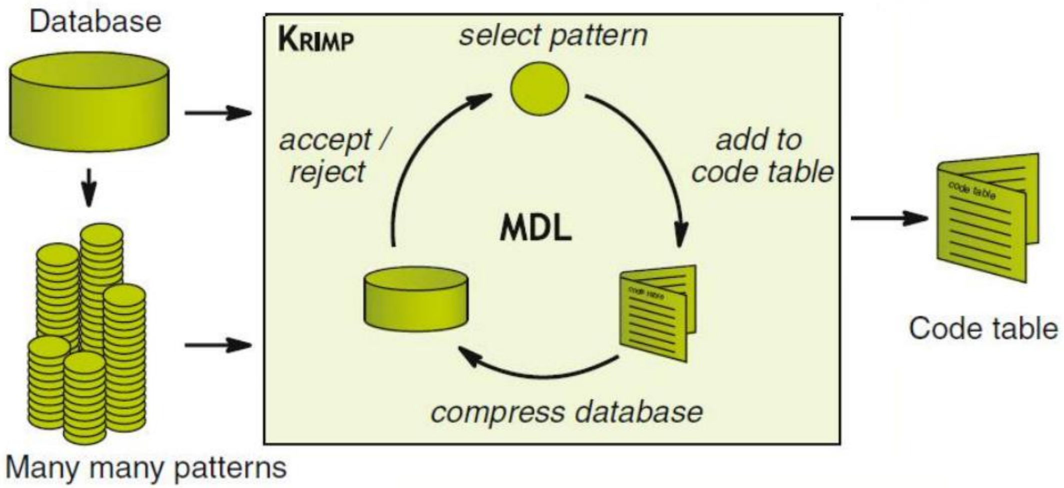


Fig. 2. How KRIMP works

The goal of MDL is to find CT that minimizes the compressed MDL size of the database:

$$L(D, CT) = L(D|CT) + L(CT|D) \qquad (1)$$

Preprocessing:
The MDL Database Expansion Method (3)

- Consider a set of patterns in CT, the expanded pattern generated from k-topmost similar itemsets defined by:
- $$F_i^* = \bigcup_{i=1}^{k} \min_{1 \le j \le l, i \ne j} \qquad (2)$$
- Where and Δ denotes text similarity technique, e.g., cosine similarity or Jaccard similarity. The l denotes the number of tuples in CT.

Preprocessing: The MDL Database Expansion Method (4)

- There are three kinds of situations about frequent itemsets that KRIMP generates from a sentence:
- **Ideal:** The ideal itemset of Fi* 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 Fi*, thus creating an incomplete pair. This situation needs an expansion process of k-topmost similar pattern generation until Fi* has at least one entity pair inside. If there is no entity pair found in Fi* after the combination of k-topmost similar itemset, then Fi* 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.

Preprocessing: The MDL Database Expansion Method (5)

(a). Ideal

0 1 2 3 : 0 is Entity (PERSON), 2 is Entity (LOCATION)
0 4 5 : 0 is Entity (PERSON), 4 is Entity (ORGANIZATION)

(b). Half ideal

0 5 6 : 0 is Entity (PERSON)
4 7 : 4 is Entity (ORGANIZATION)
1 2 : 2 is Entity (LOCATION)

(c). Not ideal

1 3 5 : No Entity is found

CT:

0	1	3	6	
0	3	5	6	(sim=0.75)
1	2	5	6	(sim=0.5)
2	5	6	7	(sim=0.25)
0	3	5		(sim=0.57735)
0	3	7		(sim=0.57735)
2	5	7		(sim=0)
4	5	7		(sim=0)

Base itemset = 0 1 3 6
k = 3 Extension itemsets:
0 1 3 6 U 0 3 5 6 U 0 3 5 U 1 2 5 6 = 0 1 2 3 5 6
0 1 3 6 U 0 3 5 6 U 0 3 7 U 1 2 5 6 = 0 1 2 3 5 6 7

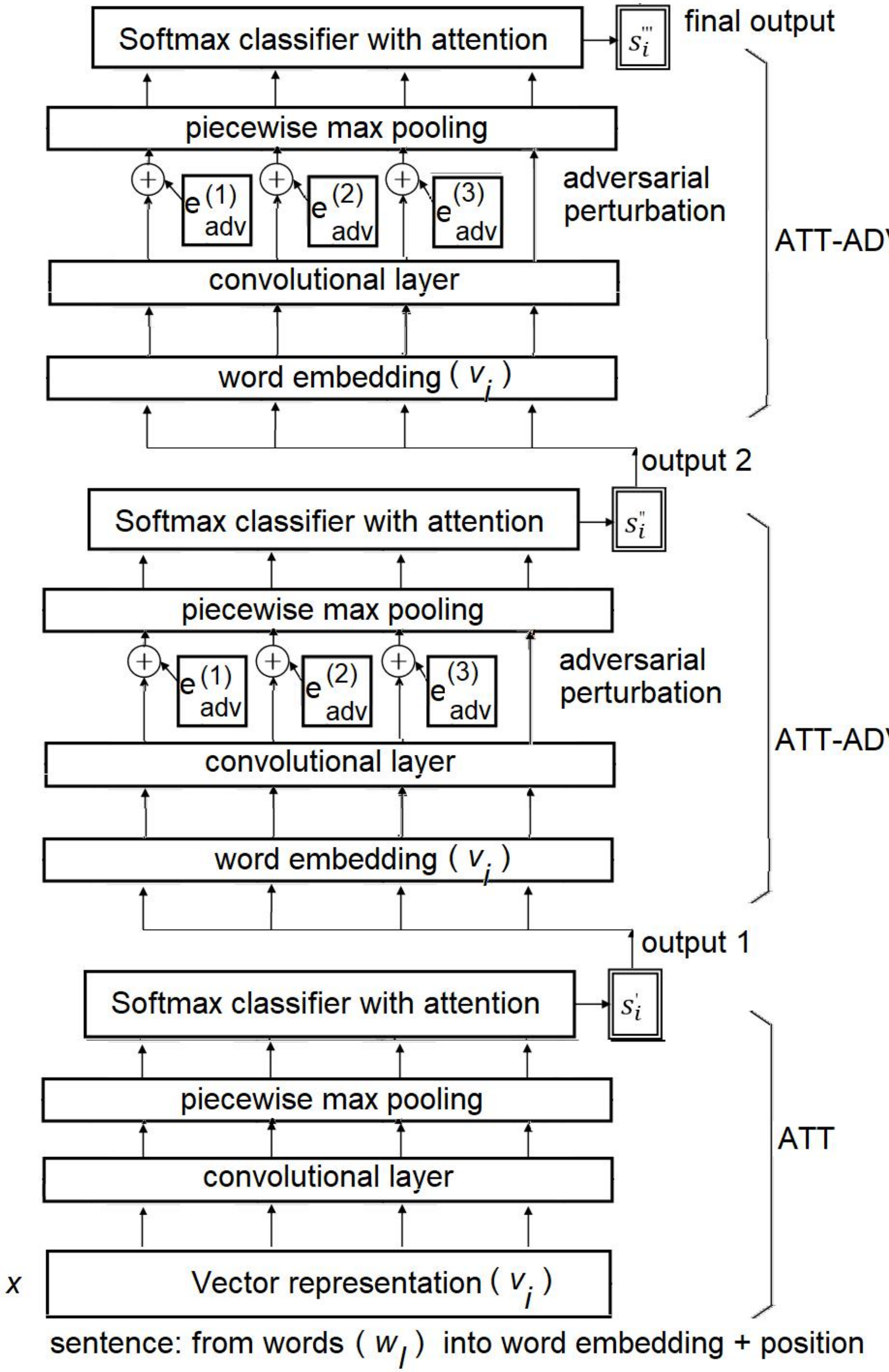
Preprocessing: The MDL Database Expansion Method (6)

- 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.

An Example of Database Expansion

Relation	/business/person/company
entity1.id	m.0dan05
entity1.name	danay
entity2.id	m.0ele14
entity2.name	eleftherotypia
*F**	steijn danay eleftherotypia
TP	danay has organization eleftherotypia

The Proposed Classifier (PCNN LATTADV-ATT) Model



The Proposed Classifier (PCNN LATTADV-ATT) Model (2)

- Selective attention (ATT): When distributed vector representations of X learned, ATT only gives attention to sentence sr that sincerely express relation r through computing the weighted average of sentences s1, s2, ..., sn, with sr defined as:

$$s_r = \sum_i \alpha_i^r s_i \qquad (3)$$

- The softmax function is used to compute the embedding vector of the query vector qr. The αr denotes the attention weights with relation r, αr=softmax(tanh(si)⊤ qr)), see [4]. ATT de-emphasizes noisy sentence. ATT helps to avoid massive false labeling during training ⊤ q(r) and testing. Final loss function for ATT[15]:

$$L_{ATT}(X,\theta) = -\sum_{i=1}^{K} \log P(r \mid x, \theta) \qquad (4)$$

- The θ denotes parameters, and the K denotes the total number of predefined relation labels r. The conditional probability of relation:

P(r|x,θ)=softmax(Asr + b) (5)

- Where A denotes a weighted diagonal representation matrix of relations, sr denotes attended sentence, and b denotes a bias vector.

The Proposed Classifier (PCNN LATTADV-ATT) Model (3)

- Adversarial training (ADV): Adversarial training first introduced in [2]. In ADV, a small amount of perturbation, eadv, 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 = {v1, v2, ..., vn}, ADV adds eadv to V. The final loss function for ATT-ADV is defined as follows [15]:

$$LATT\text{-}ADV(X,\theta)=LATT(X+eadv,\theta) \qquad (6)$$

where:

$$eadv=\epsilon g/||g|| \qquad (7)$$

and

$$g = \nabla_v L_{ATT}(X, \hat{\theta}) \qquad (8)$$

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

Experiments

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.

Experiments (2)

Training set	k	#Rel	#Ent-Pair	#Mention	#Sentence
base	0	53	172399	293162	570088
cos	3	53	172499	293170	570388
jac	3	53	172499	293170	570388
cos	7	53	173099	293187	572990
jac	7	53	173099	293185	572988
cos	10	53	173299	293185	573649
jac	10	53	173299	293185	573649

Testing set	k	#Rel	#Mention	#Sentence
base	0	53	1950	172448
cos	3	53	1950	172525
jac	3	53	1950	172525
cos	7	53	1950	173175
jac	7	53	1950	173175
cos	10	53	1950	173341
jac	10	53	1950	173341

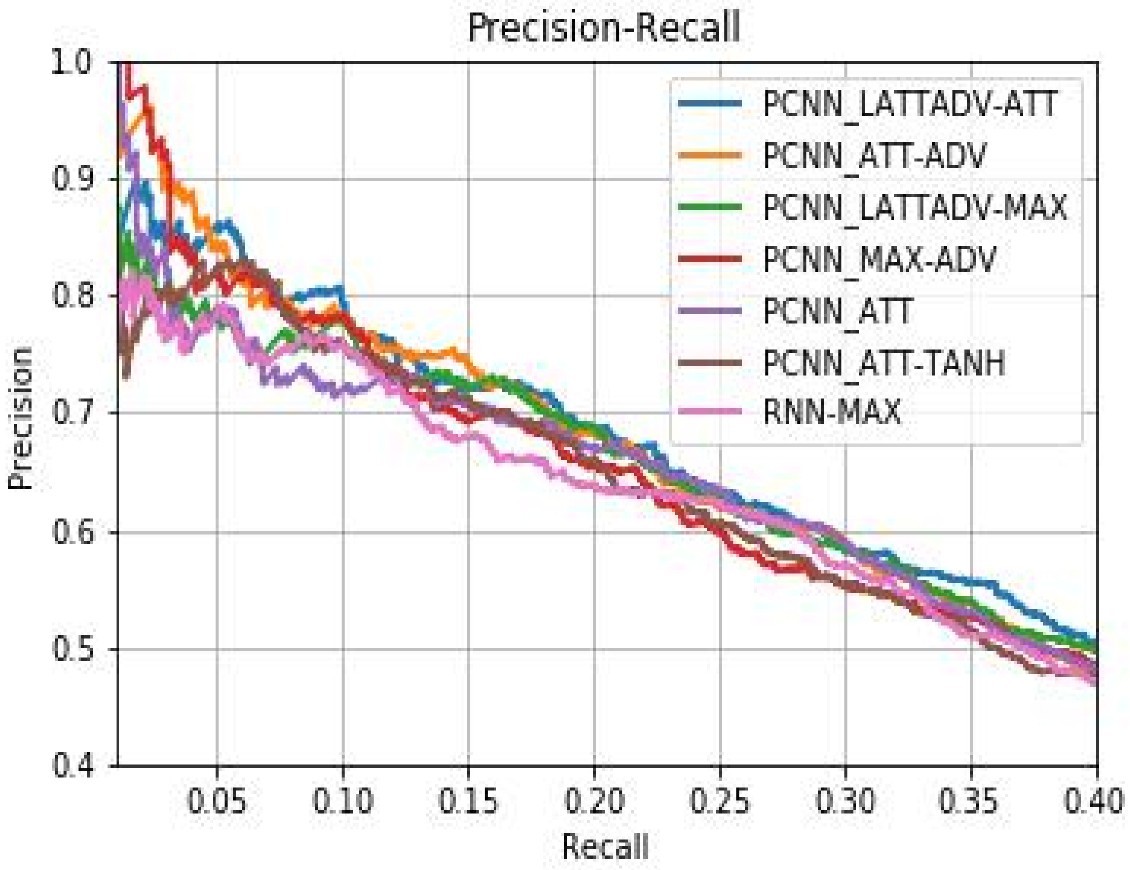
Word size=50, position size=5, hidden size=230.
#Rel: number of relation.
#Ent-Pair: number of entity pair.
cos: NYT dataset+expansion with k topmost similar itemsets by Cosine similarity.

Experiments (3)

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	0.861	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

Experiments (4)



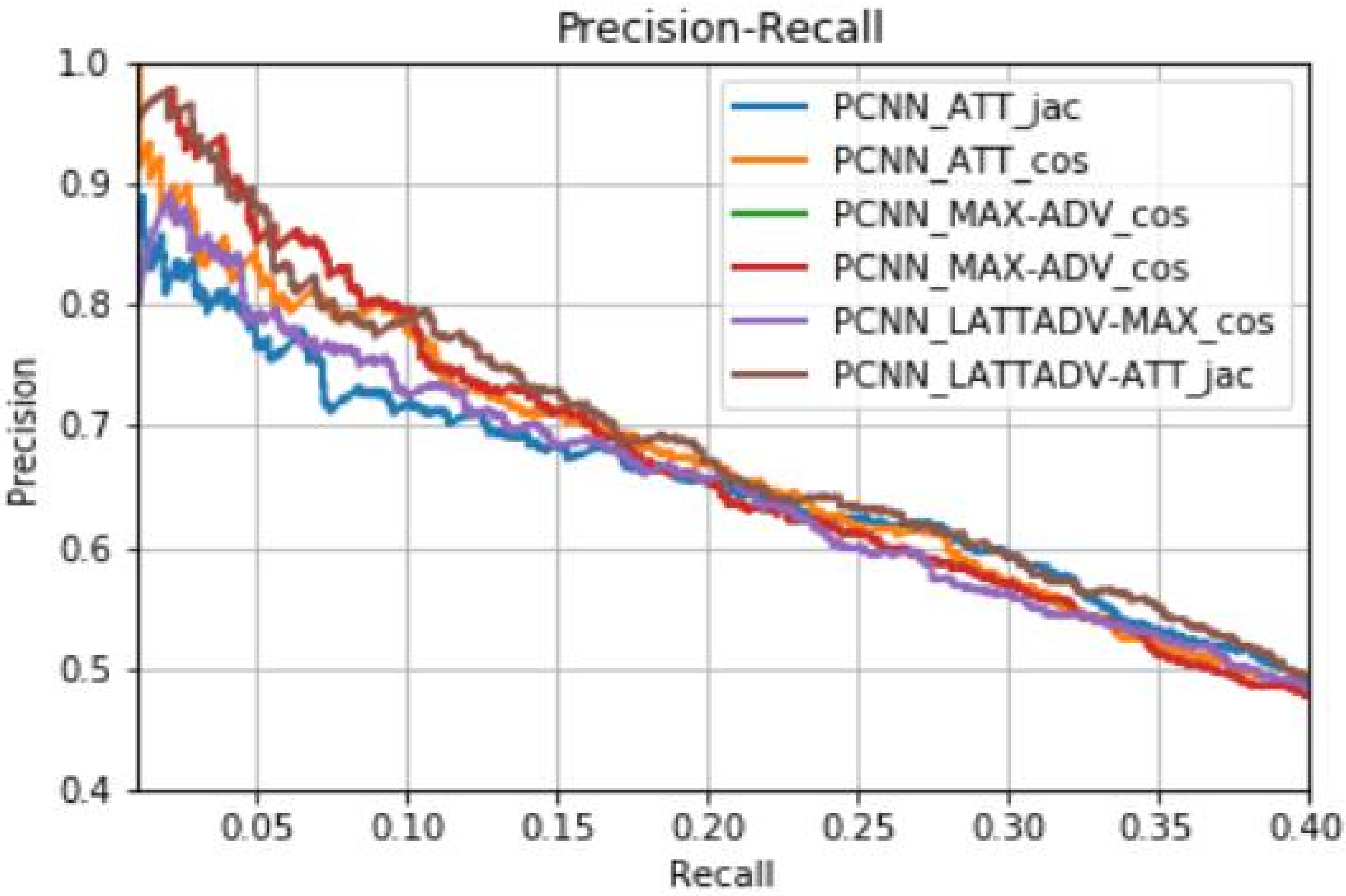
LATTADV-ATT: PCNN + leveled attention and adversarial attention
ATT-ADV: PCNN + attention + adversarial
LATTADV-MAX: PCNN + leveled attention and adversarial + maximum function
MAX-ADV: PCNN + maximum function + adversarial
ATT: PCNN with attention
ATT_tanh: PCNN with attention + tanh activation function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

Experiments (5)

M1: PCNN ATT; M2: PCNN ATT-ADV; M3: PCNN MAX-ADV;
M4: PCNN LATTADV-MAX; M5: PCNN LATTADV-ATT
Jac: Jaccard similarity; Cos: Cosine similarity

Alg	k	AUC	Max F1	P@100	P@200	P@300	Mean	Borda
M1	k=0	0.399	0.448	0.753	0.726	0.724	0.734	28
M1	Jac, k=3	0.398	0.44	0.782	0.761	0.728	0.757	33
M1	Jac, k=7	0.397	0.45	0.802	0.716	0.711	0.743	35
M1	Jac, k=10	0.404	0.438	0.822	0.796	0.734	0.784	95
M1	k=0	0.399	0.448	0.753	0.726	0.724	0.734	28
M1	Cos, k=3	0.408	0.45	0.782	0.781	0.764	0.776	109
M1	Cos, k=7	0.406	0.442	0.822	0.791	0.738	0.784	100
M1	Cos, k=10	0.4	0.44	0.822	0.751	0.73	0.767	45
M2	k=0	0.412	0.442	0.861	0.786	0.754	0.801	149
M3	k=0	0.402	0.447	0.802	0.776	0.741	0.773	74
M3	Cos, k=3	0.406	0.446	0.881	0.796	0.761	0.813	109
M3	Cos, k=7	0.407	0.445	0.891	0.811	0.744	0.815	100
M3	Cos, k=10	0.399	0.438	0.861	0.781	0.728	0.79	45
M4	k=0	0.41	0.452	0.782	0.761	0.741	0.761	79
M4	Cos, k=3	0.405	0.44	0.852	0.766	0.734	0.784	80
M4	Cos, k=7	0.406	0.454	0.842	0.761	0.731	0.778	88
M4	Cos, k=10	0.405	0.44	0.852	0.766	0.734	0.784	73
M5	k=0	0.418	0.456	0.842	0.801	0.761	0.801	187
M5	Jac, k=3	0.415	0.449	0.852	0.791	0.764	0.802	169
M5	Jac, k=7	0.422	0.455	0.891	0.791	0.771	0.818	207
M5	Jac, k=10	0.416	0.451	0.861	0.796	0.751	0.803	179

Experiments (6)



- Fig. 7.Precision-Recall of LATTADV-ATT with expansion factor (k=7)

Effect of Number of Relations in Database Expansion

- 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).
- Test carefully. 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.
- The best improvement is achieved using LATTADV-ATT with Jaccard similarity expansion at *k*=7 (Table III and Fig. 7).

Effect of The Deep Learning Strategy Setup and The Similarity Metric Used

- 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=0.64%, P@300=2.25%, 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%, mean of precisions=11.44%.
- **The use of the database expansion can improve the relation extraction system's performance.**
- **Cosine or Jaccard similarity can be used substitutive.**

Conclusions

- 1) The MDL database 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]I.J. 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.

References

- [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 XIII, 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. Bamman, 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 Convolutional 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.