ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414944

ENHANCING DEEP PARAPHRASE IDENTIFICATION VIA LEVERAGING WORD ALIGNMENT INFORMATION

Boxin Li[⋆†] *Tingwen Liu*[⋆†] *Bin Wang*[§] *Lihong Wang*[¶]

[⋆] Institute of Information Engineering, Chinese Academy of Sciences
[†] School of Cyber Security, University of Chinese Academy of Sciences
[§] Xiaomi AI Lab
[¶] National Computer Network Emergency Response Technical Team Coordination Center of China
{liboxin, liutingwen}@iie.ac.cn, wangbin11@xiaomi.com, wlh@isc.org.cn

ABSTRACT

Recent deep learning based methods have achieved impressive performance on paraphrase identification (PI), a fundamental NLP task, judging whether two sentences are semantically equivalent or not. However, their success heavily relies on massive labeled samples, which are time-consuming and expensive to obtain. To alleviate this problem, this study explores the effect of word alignment information (WAI), extracted by existing monolingual alignment tools, on deep PI baseline models. Apart from directly encoding WAI into fixed-size embeddings, we propose a novel auxiliary task so that the baselines can be pre-trained using a large amount of unlabeled in-domain data. Moreover, our proposed auxiliary task can also jointly train with the baselines, aiming to eliminate the overheads of preprocessing WAI at the test period. Experimental results verify that our methods can significantly outperform the deep PI baseline model.

Index Terms— Paraphrase identification, deep neural networks, word alignment information, multi-task learning, pre-training

1. INTRODUCTION

Paraphrase identification (PI), detecting whether two sentences convey the same meaning or not, is a fundamental problem in NLP. Many deep learning based PI methods have been proposed and achieved promising results in recent years. In general, they mainly focus on how to improve the representation and generalization capabilities of deep PI models, such as encoding as much interaction information between two sentences as possible [1, 2, 3, 4], and designing deeper models [5, 6, 7, 8, 9]. However, excellent performance achieved by these methods heavily depends on a large number of labeled samples, which are time-consuming and expensive to obtain.

Some previous studies have already explored word matching information (WMI), i.e., exact *string* matching, between two sentences to enhance the performance of deep PI models. For instance, Tay et al. [5] encoded WMI as fixed-size embeddings and concatenated them with the word embeddings. Chen et al. [10] developed a collaborative and adversarial game to explicitly extract the common features between two sentences. However, WMI contains no synonym words or appositive words, which are also helpful to the PI task.

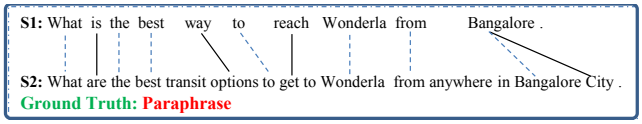

Fig. 1. One example of monolingual alignment from QQP. Here, word alignment information (WAI) refers to word pairs connected by lines. While word matching information (WMI), shown by the dotted lines, contains no synonym words or appositive words, such as (way, options) and (Bangalore, City).

To address this problem, we observe that there exists many well-designed and off-the-shelf monolingual alignment tools [11, 12, 13], which are able to automatically pair semantically similar units from two pieces of text. The outputs of a monolingual alignment tool, i.e., word alignment information (WAI), containing synonym words and appositive words as well (as illustrated in Fig. 1.), are more closely related to the PI task, and thus have the potential to further advance the performance of deep PI models.

Furthermore, inspired by [14], which utilizes unlabeled in-domain data with a framework of variational autoencoders to further improve the deep PI baseline model's performance, we propose a novel auxiliary task, classifying whether a word in one sentence has its similar counterpart in the other sentence based on its hidden representation in deep PI baseline models. Apparently, our proposed auxiliary task is capable of pre-training deep PI baselines when a large number of unlabeled in-domain data is available. Moreover, we can also jointly train the end task (PI) with this auxiliary task using

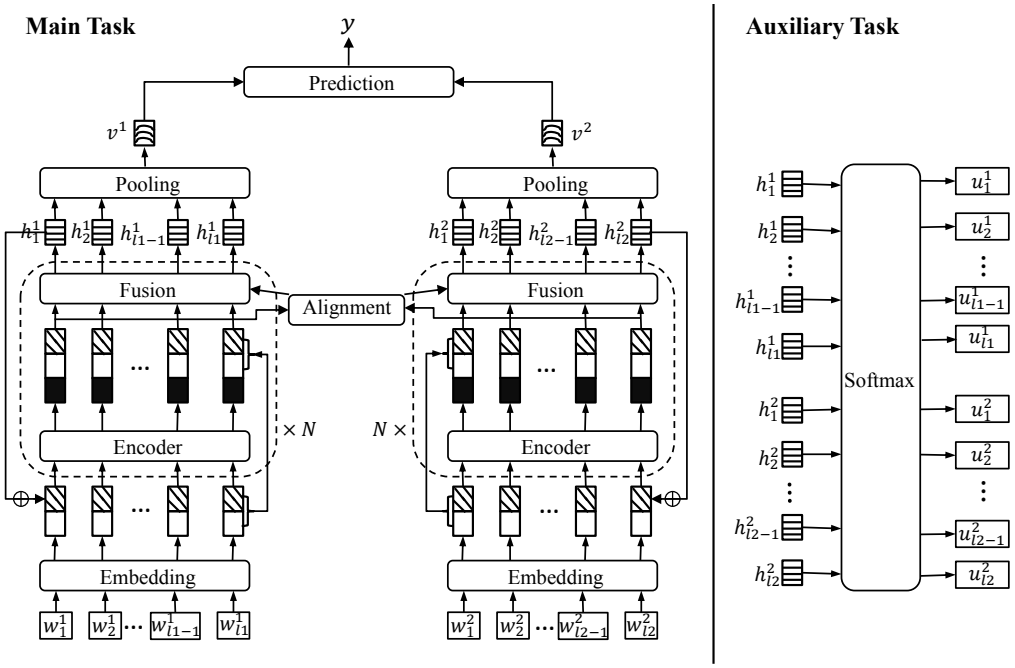

Fig. 2. The overview of the main task [6] and our proposed auxiliary task.

some multitask-learning scheme, lifting the restriction of preprocessing WAI at the test phase.

To sum up, the contributions of this paper are as follows:

(1) We explore in detail the effects of WAI from a pair of sentences on deep PI models.

(2) We propose a novel auxiliary task to pre-train deep PI models in the setting with a large amount of unlabeled in-domain data to improve the baselines' generalization ability.

(3) The proposed auxiliary task can also be used to jointly train with the end task, free from preprocessing WAI at the test period.

(4) Experimental results demonstrate that our methods can significantly advance the deep PI model in both supervised and semi-supervised settings.

2. METHOD

In this section, we will introduce our approaches to leverage WAI to advance the deep PI model.

2.1. Notations and Task Definition

PI is traditionally formulated as a binary classification task. Let $\mathbf{x}^{(i)} = (s^{(i),1}, s^{(i),2})$ denote the *i-th* sample of two sentences. Let the label of $\mathbf{x}$ be $y \in \{0, 1\}$, where *"0"* denotes the non-paraphrase relationship and *"1"* stands for the paraphrase relationship. Given the training dataset $D = \{(\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{m}$ with m samples, a baseline solution is to directly train a classification model using D. In this work, we take the baseline method as a deep neural network-based model, such as RE2 [6]. These methods usually first transform a sample $\mathbf{x}$ (here a sentence pair) into a fixed-size vector $\mathbf{z}$, i.e., $\mathbf{z} = f_\Theta(\mathbf{x})$, where f_Θ denotes the transformation function parameterized by Θ. Next, a softmax classifier, parameterized by β, is learned to map $\mathbf{z}$ to a paraphrase relationship label y. Then, using cross-entropy loss, the goal of PI is to minimize:

$$J(\Theta; \beta) = -\frac{1}{m}\sum_{i=1}^{m} log\, p(y^{(i)}|f_\Theta(\mathbf{x}^{(i)}); \beta), \qquad (1)$$

where $y^{(i)}$ denotes the ground truth label of the *i-th* sample.

2.2. Base Model

In this paper, we use RE2 [6] as our base model, as illustrated in the left part of Fig. 2., due to its promising performace with remarkable efficiency compared to other state-of-the-art models for the PI task. We introduce readers to see more details in [6].

2.3. Word Alignment Information as Additional Input

Here we take the mainstream method to leverage WAI following [5]. To be specific, we convert WAI into a binary value feature $u \in \{0, 1\}$ to represent whether a word in one sentence has its similar counterpart in the other sentence; and then we encode it into a fixed-size vector embedding which can be tuned during networks' training. We call this strategy

as the *embedding* scheme. Note that the overall architecture of this scheme is almost the same as the base model, with only a slight difference that we feed the combination of word embeddings and WAI embeddings as input into the baseline model.

2.4. Auxiliary Task

In order to make full use of unlabeled in-domain data, and meanwhile reduce the prepocessing overheads during the test time, we propose a novel auxiliary task that classifies WAI categories of each word in input sentence pairs based on their hidden representations in the deep PI baseline model, as illustrated in the right part of Fig. 2. We can readily regard the proposed auxiliary task as a binary tagging task, of which the categories can be chosen from 1/0 to represent whether each word in a sentence has its similar counterpart in the other sentence or not. The golden WAI labels can be extracted from existing monolingual alignment tools[1,2]. In this way, after the deep baseline model outputs the hidden states, we feed the hidden state of each word into a softmax function for our auxiliary task to get its predicted probabilities:

$$p(\hat{u}_j^q|h_j^q) = softmax(W_2 h_j^q + b_2), \tag{2}$$

where $W_2 \in \mathbb{R}^{T\times d}$ and $b_2 \in \mathbb{R}^T$ are the parameters of softmax, T is the total number of WAI categories, d is the dimension of the hidden state in the deep baseline model, h_j^q is the hidden state of the *j-th* token of the *q-th* sentence in the deep learning process, and u_j^q is its corresponding WAI label. Thus, let $\beta^{'}$ denote the parameters of the softmax classification layers, i.e., $\beta^{'} = (W_2, b_2)$, using cross-entropy loss, the goal of the auxiliary task is to minimize:

$$J(\Theta;\beta^{'}) = -\frac{1}{2m}\sum_{i=1}^{m}\sum_{q=1}^{2}(\frac{1}{l_q^{(i)}}\sum_{j=1}^{l_q^{(i)}} log\, p(u_j^{(i),q}|f_\Theta(\mathbf{x}^{(i)});\beta^{'})), \tag{3}$$

where $l_q^{(i)}$ denotes the total number of words contained in the *q-th* sentence of the *i-th* sample and m is the total number of training samples.

Multi-task Learning. Here we adopt a simple and effective multi-task learning scheme, i.e., hard parameter sharing, meaning that the end task (PI) and the auxiliary task share the common feature representation layers, and meanwhile possess respective task-specific classification layers. Combining Eq.1 and Eq.3, the total loss function of our proposed framework is as follows:

$$\begin{aligned} J(\Theta;\beta,\beta^{'}) &= -\frac{1}{m}\sum_{i=1}^{m} log\, p(y^{(i)}|f_\Theta(\mathbf{x}^{(i)});\beta) \\ -\frac{\lambda}{2m}\sum_{i=1}^{m}\sum_{q=1}^{2}(\frac{1}{l_q^{(i)}}&\sum_{j=1}^{l_q^{(i)}} log\, p(u_j^{(i),q}|f_\Theta(\mathbf{x}^{(i)});\beta^{'})), \end{aligned} \tag{4}$$

where λ is a hyperparameter to tradeoff the importance of the auxiliary task.

The idea behind this scheme is that the monolingual alignment tools exploit word similarity features, syntactic features and POS features and so on, to train monolingual aligners on labeled alignment datasets, which is equivalent to incorporate some prior knowledge. For example, their outputs include some synonyms and aligned named entities from shorthand to full name. When jointly learning the end task and the auxiliary task, two improvements are introduced into the networks by the auxiliary task: 1) injecting some external knowledge; and 2) providing a regularization effect.

Pre-training. Hashimoto et. al. [15] introduced a joint many-task model to successively grow its depth by increasingly learning natural language processing tasks from lower-leval tasks, such as Part-Of-Speech tagging and syntactic parsing, to higher-level tasks, such as sentence similarity and natural language inference. Inspired by their work, we pre-train the baseline model using our auxiliary task when a large number of in-domain unlabeled data is available, expecting to integrate more WAI into the original deep PI model.

3. EXPERIMENT

In this section, we conduct experiments to verify the effectiveness of WAI to advance the deep PI baseline model.

3.1. Experimental Setting

Datasets. We evaluate our methods on the public PI dataset QQP (Quora Question Pairs) [2], which contains 384,348 training pairs, 10,000 dev pairs and 10,000 test pairs. We remove 18 training pairs, which contains only punctuations in at least one sentence, resulting in 384,330 training pairs. In order to conduct and compare both supervised and semi-supervised experiments, we select 1k, 5k, 10k, 25k labeled pairs in the training set to make up four new labeled training sets, and remove the labels of data in the training set to construct the unlabeled set, along the same line as previous studies [14, 16].

Table 1. Settings of hyperparameters.

epochs	bsize	hsize	blocks	#enc.	#conv.	dropout
30	128	200	2	3	3	0.2

Implementation Details. Our methods in this paper are implemented with PyTorch[3] to follow the original paper[4] and trained on Tesla K80 GPUs. The hyperparameters are listed in Table 1. We train all of our models using the Adam [17] optimizer with the default parameters β_1 and β_2 to be 0.9 and 0.999 respectively, and the initial learning rate is set to

[1]https://github.com/ma-sultan/monolingual-word-aligner/

[2]http://code.google.com/p/jacana/

[3]https://pytorch.org/

[4]https://github.com/alibaba-edu/simple-effective-text-matching

0.001. The word embeddings are initialized with Glove [18] of 300 dimensions and not updated during training. We set the max sequence length to 100. For the tradeoff hyperparameter λ in Eq.4, we choose from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. The size of the WAI and WMI vector embedding is chosen from $\{1, 5, 10, 15, 20\}$. We experiment on three random sampling training sets for each amount of labeled data, and report the average results. For all experiments, we choose the model which scores the best over the classification accuracy metric on the dev set, and then evaluate it on the test set.

Table 2. Comparison on the test set of QQP. Here, "***emb***", the abbreviation of "embedding", means encoding WMI or WAI into fixed-size vector embeddings and tuned during training. "***mul***", the abbreviation of "multi-task", represents jointly training the end task (PI) with the proposed auxiliary task using a multi-task learning scheme. "***pre***", the abbreviation of "pre-train", represents pre-training neural networks using the proposed auxiliary task with unlabeled data. Best results are in bold and 2nd best are underlined. Classification accuracy (%) is used as the evaluation metric.

Model	1k	5k	10k	25k
Supervised				
LexDec [1]	68.9	72.2	73.9	75.7
ESIM [4]	68.3	74.6	76.7	79.5
BiMPM [2]	67.0	72.6	74.6	78.6
CSRAN [19]	63.9	72.9	76.3	79.7
Semi-supervised				
DeConv-AE [16]	60.2	65.1	67.7	71.6
LSTM-LVM [16]	62.9	67.6	69.0	72.4
DeConv-LVM [16]	65.1	69.4	70.5	73.7
DV-VAE[5] [14]	68.1	73.1	74.7	77.0
Ours				
RE2 [6]	67.7	73.9	76.3	79.4
+emb(WMI)	73.0	76.5	78.3	80.7
+mul(WMI)	70.0	75.6	77.6	80.5
+pre(WMI)	73.5	76.6	78.3	81.2
+pre+mul(WMI)	73.6	76.6	79.2	81.8
+emb(WAI)	73.4	77.1	78.9	80.7
+mul(WAI)	69.8	75.6	77.8	80.9
+pre(WAI)	<u>74.5</u>	<u>78.4</u>	**80.7**	<u>82.1</u>
+pre+mul(WAI)	**74.8**	**78.5**	<u>80.6</u>	**82.9**

3.2. Experimental Result and Analysis

We compare our methods with both supervised and semi-supervised baselines, and can draw some interesting conclusions from Table 2.

[5]We omit their std. errors for conciseness.

First, it's clear that our methods that incorporate WAI significantly outperform the RE2 baseline for all of the four subsets with distinct labeled data size. Meanwhile, the performance of our methods are also far better than that of all the supervised and semi-supervised baselines. These results demonstrate the importance of incorporating WAI for the deep PI model.

Second, WAI almost always achieves better results than WMI under the same scheme. This indicates that the monolingual alignment tools indeed introduce more useful prior knowledge, which is helpful to further advance the deep PI baseline.

Third, the performance of the multi-task learning scheme is inferior to that of the *embedding* scheme. But these gaps narrow rapidly when the size of labeled data increases, which shows the promising prospects of the multi-task learning scheme, considering that it's able to avoid preprocessing overhead at the inference phase.

Fourth, we observe that the hybrid scheme (*"pre+mul"*) exploiting WAI obtains best results with an average increase of 4.9% over the RE2 baseline on all of the four subsets with distinct labeled data size, compared to the average increment of 3.5% of WMI, which further demonstates the superiority of WAI. Moreover, significant improvements are always observed when leveraging the pre-training scheme, verifying its effectiveness of utilizing unlabeled in-domain data to improve supervised learning.

4. CONCLUSION AND FUTURE WORK

In this paper, we explored in detail the effect of WAI from a pair of sentences on the deep PI model. In addition to the direct way to concatenate word embeddings and WAI embeddings, we proposed a novel auxiliary task to judge whether a word in a sentence has its similar counterpart in the other sentence or not. Our proposed auxiliary task enables opening up avenues for both supervised and semi-supervised settings. Experimental results show that WAI has superiority over WMI and are able to significantly advance the deep PI baseline model especially when a large number of unlabeled in-domain data is abailable.

For future work, we will investigate the impact of the quality of monolingual word alignment tools over the PI task, develop some more specific and accurate monolingual word alignment tools to further advance deep PI models, and extend our methods to other languages such as Chinese.

5. ACKNOWLEDGEMENTS

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDC0204 0400). We would like to thank anonymous reviewers for their insightful comments. Tingwen Liu is the corresponding author.

6. REFERENCES

[1] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah, "Sentence similarity learning by lexical decomposition and composition," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1340–1349.

[2] Zhiguo Wang, Wael Hamza, and Radu Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 4144–4150.

[3] Yichen Gong, Heng Luo, and Jian Zhang, "Natural language inference over interaction space," in *International Conference on Learning Representations*, 2018.

[4] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen, "Enhanced lstm for natural language inference," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1657–1668.

[5] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui, "Co-stack residual affinity networks with multi-level attention refinement for matching text sequences," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4492–4502.

[6] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen, "Simple and effective text matching with richer alignment features," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4699–4709.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[9] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2019.

[10] Qin Chen, Qinmin Hu, Jimmy Xiangji Huang, and Liang He, "Can: Enhancing sentence similarity modeling with collaborative and adversarial network," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 815–824.

[11] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark, "Answer extraction as sequence tagging with tree edit distance," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 858–867.

[12] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark, "A lightweight and high performance monolingual word aligner," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013, pp. 702–707.

[13] Md Arafat Sultan, Steven Bethard, and Tamara Sumner, "Feature-rich two-stage logistic regression for monolingual alignment," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 949–959.

[14] Zhongbin Xie and Shuai Ma, "Dual-view variational autoencoders for semi-supervised text matching.," in *IJCAI*, 2019, pp. 5306–5312.

[15] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher, "A joint many-task model: Growing a neural network for multiple nlp tasks," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1923–1933.

[16] Dinghan Shen, Yizhe Zhang, Ricardo Henao, Qinliang Su, and Lawrence Carin, "Deconvolutional latent-variable model for text sequence matching," in *AAAI*, 2018.

[17] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[18] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[19] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui, "Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1565–1575.