Domain Adaptation for Disease Phrase Matching with Adversarial Networks

Miaofeng Liu[♦*], **Jialong Han**[♠], **Haisong Zhang**[♠], and **Yan Song**[♠]
[♦]University of Science and Technology of China
[♠]Tencent AI Lab
{water3er,jialonghan}@gmail.com,{hansonzhang,clksong}@tencent.com

Abstract

With the development of medical information management, numerous medical data are being classified, indexed, and searched in various systems. Disease phrase matching, *i.e.,* deciding whether two given disease phrases interpret each other, is a basic but crucial preprocessing step for the above tasks. Being capable of relieving the scarceness of annotations, domain adaptation is generally considered useful in medical systems. However, efforts on applying it to phrase matching remain limited. This paper presents a domain-adaptive matching network for disease phrases. Our network achieves domain adaptation by adversarial training, *i.e.,* preferring features indicating whether the two phrases match, rather than which domain they come from. Experiments suggest that our model has the best performance among the very few non-adaptive or adaptive methods that can benefit from out-of-domain annotations.

1 Introduction

In recent years, hospitals depend more on information systems to store and retrieve medical data for diagnosis and treatment. To facilitate reliable and efficient processing of medical data, *disease phrase matching* has been identified as a crucial task in those medical systems. Given two disease phrases, this task requires identifying whether they are able to interpret each other.

Owing to complicated medical terminologies, overlapping words or similar syntactic structures are not reliable cues for disease phrase matching. Table 1 shows two matching candidates for *"Latent syphilis, specified as early or late"* (Phrase 1). In the first one, the absent participial modifier and the different word order do not prevent the two phrases from matching. The second one is, however, a false match, though it shares more words and similar syntactic structures with Phrase 1.

Phrase 1	Phrase 2	Label
Latent syphilis, specified as early or late	*Syphilis latent*	Yes
Latent syphilis, specified as early or late	*Late syphilis, specified*	No

Table 1: Examples of disease phrase matching.

Given the variability of human languages, supervised phrase or sentence matching is widely applied in information identification (Madnani et al., 2012; Yin et al., 2016), textual entailment (Marelli et al., 2014), web search (Li et al., 2014), entity linking (Traylor et al., 2017), and disease inference (Nie et al., 2015). As deep learning drew attentions on various tasks (Lecun et al., 2015), dedicated neural matching models are also designed in two types of structures. 1) **Siamese-based networks** (Neculoiu et al., 2016; Mueller and Thyagarajan, 2016): the input phrases are first encoded by the same network; the encoded vectors are then used to compute similarities by metrics like Cosine. 2) **Matching-aggregating networks**: fine-grained units of the two phrases are represented and matched in word-by-word (Rocktäschel et al., 2015), one-direction (Wang and Jiang, 2016), or bilateral-multi-perspective (Wang et al., 2017) manners to produce matching features; the features are aggregated into a vector, based on which the matching label is predicted.

Despite encouraging results in other areas, neural matching models still face specific challenges on medical data. Different medical subfields like physiology and urology may adopt diverse terminologies. Due to their professional nature, it is hard to obtain human annotations at scale for a single subfield. This causes systems on a partic-

[*]Work was done during the internship at Tencent AI Lab.

Proceedings of the BioNLP 2018 workshop, pages 137–141
Melbourne, Australia, July 19, 2018.

ular target subfield or domain to have too few annotations to learn a complicated neural model. It may be tempting to involve annotations from one or more source domains for more training data. But since all above models assume in-domain annotations, the effect of source-domain annotations remains uncertain on the trained models.

This paper takes a perspective that is orthogonal to works on designing sophisticated matching networks. We employ domain adaptation in disease phrase matching to effectively exploit source annotations. Based on Bilateral Multi-Perspective Matching (BiMPM) (Wang et al., 2017), we propose a Domain-Adaptive BiMPM (DA-BiMPM) model. Inspired by domain-adversarial training (Ganin et al., 2016) on text classification (Liu et al., 2017), relation extraction (Fu et al., 2017), and paraphrase identification (Yu et al., 2018), we introduce a domain discriminator in addition to the matching predictor in BiMPM. With such a discriminator, DA-BiMPM is encouraged to learn features predictive of the matching labels, while being least discriminative of which domain the data comes from. In doing so, it is expected that the learned models distill domain-insensitive knowledge from source annotations. On two medical datasets from different subfields, we set up non-adaptive baselines fed with or without source-domain annotations, as well as an adaptive one. Experimental results show that, when trivially involving source-domain data, only the strongest baseline BiMPM can achieve a slight gain. Compared with the adaptive approach, DA-BiMPM is capable of making more improvement on BiMPM.

2 Preliminaries

Before going into details of DA-BiMPM, we start with introducing the BiMPM model (Wang et al., 2017), which is illustrated by components outside the dotted box in Figure 1. Its encoding, matching, and aggregation layers are described as follows.

Phrase Encoder. Given a disease phrase $\mathrm{P} = (\mathrm{p}_1, \ldots, \mathrm{p}_n)$ with n words, BiMPM encode it as follows. First, it transforms P in to a vector sequence $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$. Each word is represented by concatenating a pre-trained GloVe (Pennington et al., 2014) vector and a character-BiLSTM-encoded vector. A BiLSTM is then applied on $\mathbf{P}$ to represent context in both directions:

$$\overleftarrow{\mathbf{H}}^{\mathrm{P}} = (\overleftarrow{\mathbf{h}}^{\mathrm{P}}_1, \overleftarrow{\mathbf{h}}^{\mathrm{P}}_2, \ldots, \overleftarrow{\mathbf{h}}^{\mathrm{P}}_n) = \overleftarrow{\mathrm{LSTM}}(\mathbf{P}) \quad (1)$$

$$\overrightarrow{\mathbf{H}}^{\mathrm{P}} = (\overrightarrow{\mathbf{h}}^{\mathrm{P}}_1, \overrightarrow{\mathbf{h}}^{\mathrm{P}}_2, \ldots, \overrightarrow{\mathbf{h}}^{\mathrm{P}}_n) = \overrightarrow{\mathrm{LSTM}}(\mathbf{P}) \quad (2)$$

Phrase Matcher. Given context representations

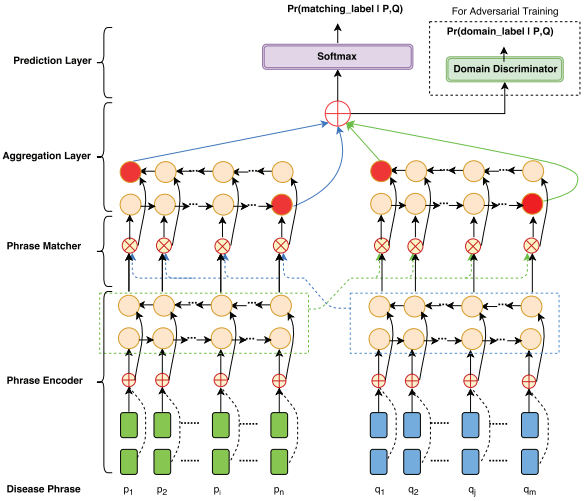

Figure 1: The architecture of (DA-)BiMPM.

of P and Q, a *phrase matcher* compares them with each time step of one against all of the other's in both directions. For example, when comparing word p_i with Q, we generated a *matching vector*

$$\mathbf{m}^{\mathrm{P}}_i = (\overrightarrow{\mathbf{h}}^{\mathrm{P}}_i \otimes \overrightarrow{\mathbf{H}}^{\mathrm{Q}}, \overleftarrow{\mathbf{h}}^{\mathrm{P}}_i \otimes \overleftarrow{\mathbf{H}}^{\mathrm{Q}}) \quad (3)$$

Here $\otimes$ denotes the *multi-perspective matching* operation defined in (Wang et al., 2017). We refer readers to this paper for details.

Aggregation Layer. Given all matching vectors $\mathbf{M}^{\mathrm{P}} = (\mathbf{m}^{\mathrm{P}}_1, \ldots, \mathbf{m}^{\mathrm{P}}_n)$ by comparing P to Q, and $\mathbf{M}^{\mathrm{Q}}$ vice versa, we apply another BiLSTM layer to aggregate both of them, respectively. Formally,

$$(\overrightarrow{\mathbf{A}}^{\mathrm{P}}, \overleftarrow{\mathbf{A}}^{\mathrm{P}}) = \mathrm{BiLSTM}(\mathbf{M}^{\mathrm{P}}) \quad (4)$$

$$(\overrightarrow{\mathbf{A}}^{\mathrm{Q}}, \overleftarrow{\mathbf{A}}^{\mathrm{Q}}) = \mathrm{BiLSTM}(\mathbf{M}^{\mathrm{Q}}) \quad (5)$$

Finally, we concatenate the four ending hidden vectors of the BiLSTM layer, *i.e.,* $\overrightarrow{\mathbf{a}}^{\mathrm{P}}_n, \overleftarrow{\mathbf{a}}^{\mathrm{P}}_1, \overrightarrow{\mathbf{a}}^{\mathrm{Q}}_m$, and $\overleftarrow{\mathbf{a}}^{\mathrm{Q}}_1$, as the *matching features* $\mathbf{F}$.

To decide whether P and Q match, we apply a fully connected softmax layer on $\mathbf{F}$ to produce the prediction $y(\mathbf{F})$. Denoting all parameters of the feature extraction layers by ϕ_f, and the prediction layer ϕ_y, for ground truth $y^{(k)}$ of the k-th phrase pair, the instance-level matching loss is

$$l^{(k)}(\phi_f, \phi_y) = l(y(\mathbf{F}^{(k)}), y^{(k)}) \quad (6)$$

3 Domain-Adversarial Training

Given the configurations of BiMPM, the network parameters $\{\phi_f, \phi_y\}$ are optimized to minimize the gap between predicted and ground-truth matching labels. When source-domain training data is involved, due to the large parameter space of ϕ_f, the model may be satisfied with fitting labels in each domain separately instead of finding a unified explanation. This limitation thus causes the model to miss potential benefits of learning domain-independent matching features.

To fully utilize source-domain annotations, we apply domain-adversarial training (Ganin et al., 2016) on BiMPM. As illustrated by the dotted box in Figure 1, we add a domain discriminator $d(.)$ on $\mathbf{F}$, *i.e.,* the matching features. The discriminator is configured with the same fully-connected and softmax layers as the matching prediction layer. Given the domain $d^{(k)}$ where the k-th phrase pair is from, the domain loss is similarly given as

$$l_d^{(k)}(\phi_f, \phi_d) = l_d(d(\mathbf{F}^{(k)}), d^{(k)}) \tag{7}$$

Different from minimizing the matching loss $l^{(k)}$, we optimize the domain loss $l_d^{(k)}$ in the contrary direction. In other words, we prefer $\{\phi_f, \phi_d\}$ that preserve little domain-specific information.

Formally, given training phrase pairs in the target domain with indices $k \in T$, and source-domain data with indices $k \in S$, our joint objective function is given as follows by interpolating both the matching and the domain losses:

$$\begin{aligned} L(\phi_f, \phi_y, \phi_d) &= \frac{1}{|S \cup T|} \sum_{k \in S \cup T} l^{(k)}(\phi_f, \phi_y) \\ &- \lambda \big[\frac{1}{|S|} \sum_{k \in S} l_d^{(k)}(\phi_f, \phi_d) + \frac{1}{|T|} \sum_{k \in T} l_d^{(k)}(\phi_f, \phi_d) \big] \end{aligned} \tag{8}$$

When optimizing the objective function, we seek for a saddle $\{\hat{\phi}_f, \hat{\phi}_y, \hat{\phi}_d\}$ such that:

$$\hat{\phi}_f, \hat{\phi}_y = \arg\min_{\phi_f, \phi_y} L(\phi_f, \phi_y, \hat{\phi}_d) \tag{9}$$

$$\hat{\phi}_d = \arg\max_{\phi_d} L(\hat{\phi}_f, \hat{\phi}_y, \phi_d) \tag{10}$$

By considering domain adaptation and matching label prediction in the joint objective, the training process pursuits a balance between both aspects. Interactions between the matching loss and the domain loss will force their shared parameters, *i.e.,* $\hat{\phi}_f$, to be generalizable across domains.

4 Experiments

4.1 Datasets and Baselines

We employ ICD10DATA[1] and MIMIC (Johnson et al., 2016) as the source and target domain datasets, respectively. ICD10DATA consists of diverse disease names from multiple medical subfields[2] and their approximate synonyms. MIMIC is a public dataset on computational physiology. The used phrase pairs are composed of terminology co-reference pairs of disease entities. Because both datasets consist of only positive pairs, we have to generate negative pairs. For each positive pair $\langle \mathrm{P}, \mathrm{Q} \rangle$, we corrupt Q with a random phrase from all other pairs containing neither P nor Q. We summarize both datasets in Table 2.

Dataset	# of Pairs	Subfield	Domain
ICD10DATA	29,783	Mixed	Source
MIMIC	22,504	Physiology	Target

Table 2: Statistics of source and target datasets.

We adopt a training/validation/testing split of 3:1:1 on the target dataset, and conduct 5-fold cross validation. Average results on the five testing sets are reported. When involving the source dataset to help train better classifiers for the target domain, we use all annotations for training. We compare DA-BiMPM with five baselines:

Cosine: Phrases are represented by summing their GloVe (Pennington et al., 2014) word vectors. Their similarities are measured by Cosine scores.

Support Vector Machine (SVM): An SVM classifier is trained and applied on the concatenation of the phrase pairs' GloVe vectors.

Random Forest: Instead of SVM, this baseline applies random forest to train matching classifiers.

Siamese-LSTM: We use an existing implementation[3] of Mueller and Thyagarajan (2016).

BiMPM: This is the matching-aggregating network (Wang et al., 2017) described in Section 2.

In DA-BiMPM, we adopt the same configuration with that of BiMPM. We empirically set λ in Equation 8 to 0.5 throughout the experiments.

4.2 Preliminary Results

Figure 2 demonstrates the changes of the three losses in Equations 6, 7, and 8, respectively. We observe that, as training proceeds to about 100 iterations, all losses tend to decrease and then converge. Readers may notice that the domain loss follows a decreasing trend, which seems inconsistent with its negative coefficient in Equation 8. Note that the matching and domain losses are both functions of the feature extraction parameters ϕ_f, thus are correlated. As the matching loss decreases, ϕ_f may inevitably capture domain-dependent information. Therefore, the trade-off between minimizing the matching loss and maximizing the domain loss cannot achieve both objectives in positive directions. It can only prevent the latter loss from decreasing too much. The same figure also shows that, after 20 iterations, the validation accuracy grows quickly and then converges to 96.04%, yielding a testing accuracy of 96.96%.

[1] http://www.icd10data.com/. We only used the ICD-10-CM (diagnosis) subset.

[2] We uniformly treat them as from one source domain.

[3] https://github.com/dhwajraj/deep-siamese-text-similarity

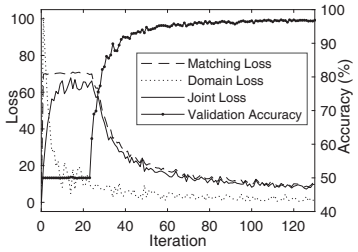

Figure 2: Training losses and validation accuracy.

Model	S. + T.	T. Only
Cosine	48.22	53.73
SVM	78.54	80.04
Random Forest	83.61	86.15
Siamese-LSTM	90.75	90.97
BiMPM	91.27	91.06
DA-BiMPM	96.96	N/A

Table 3: Testing accuracy (%) w/ or w/o source annotations.

Setting	Accuracy
BiMPM (S. Only)	90.74
BiMPM (T. Only)	91.06
BiMPM (S. + T.)	91.27
BiMPM (DDC variant)	92.39
DA-BiMPM (unsupervised)	96.12
DA-BiMPM (supervised)	96.96

Table 4: (DA-)BiMPM's testing accuracy (%) w.r.t. different settings.

4.3 Comparative Studies

In Table 3, we report the performance of all approaches. For each baseline, we train the model by aggregating both the entire source-domain dataset and the training set of the target domain. For comparison, we also trained them without the source dataset. We have the following observations.

First, when given the combined training set, the performance of the five baselines increases by the order they are presented. Specifically, the simplest Cosine approach is close to random guesses. Supervised methods like SVM and Random Forest, on the other hand, produce much better results. Neural network approaches, including Siamese-LSTM and matching-aggregating-based BiMPM, have the best performance among all baselines.

Moreover, including the source-domain dataset for training have different effects on the baselines. For the first four baselines, this dataset harms the training process and results in inferior performance. In contrast, BiMPM achieves slightly better accuracy by involving source-domain annotations. We note that such different effects may be due to a different model complexity. As a complicated model, BiMPM is able and tends to benefit from larger training data, even if they are from different domains. In summary, if exploited in a straight-forward manner, source-domain annotations cannot always guarantee better performance.

Finally, DA-BiMPM achieves more than five points of performance gain on top of BiMPM. Note that BiMPM has already taken advantage of source-domain annotations. Compared with BiMPM, DA-BiMPM only accepts domain labels as additional training information. The matching classifier trained by DA-BiMPM has the same structure, and requires the same input to make predictions, with that of BiMPM. This indicates that DA-BiMPM is making domain-adaptive exploitation of source-domain data from the feature level.

In Table 4, we evaluate DA-BiMPM in the unsupervised setting, *i.e.,* considering only source annotations in matching loss. This is done by not involving any target data when updating the prediction layer. We compete with Deep Domain Confusion (DDC) (Tzeng et al., 2014), where an adaptation layer based on Maximum Mean Discrepancy (Borgwardt et al., 2006) is applied after the phrase matcher. We also include (DA-)BiMPM's results in other relevant settings for comparison. It is observed that approaches with more information achieve better accuracy. Specifically, with access to the source data and distribution of the target training set, the unsupervised DA-BiMPM outperforms DDC-based BiMPM by nearly four points.

4.4 A Case Study

To further examine the impact of domain adaptation, we study a phrase pair *"bleed"* and *"gun shot wound to the head"* in the target set. When involving only target data, BiMPM correctly judged the pair as a mismatch. We find that, if involved in a pair, *"bleed"* on both sides tends to suggest a match. The numbers of instances for and against this feature are 1,401 and 747, respectively.

After trivially accessing source data, BiMPM achieved a slight gain. However, the above statistics are both 41 on the source set, implying a different data distribution. BiMPM was mislead on the above pair, and gave a false positive label. Meanwhile, DA-BiMPM overcomes the domain difference, and corrected the label to negative.

5 Conclusion

We present DA-BiMPM, a domain-adversarial network for disease phrase matching. It outperforms the base model BiMPM as well as four other baselines, with or without source annotations. Experiments also demonstrate that, when trivially combined with target-domain training data, source-domain data does not always make positive impacts. However, DA-BiMPM can better exploit the source-domain data, even if BiMPM or its DDC variant have taken advantage of it.

References

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* 22(14):e49–e57.

Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. volume 2, pages 425–429.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59):1–35.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data* 3.

Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521(7553):436–444.

Hang Li, Jun Xu, et al. 2014. Semantic matching in search. *Foundations and Trends® in Information Retrieval* 7(5):343–469.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1–10.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 182–190.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval@ COLING*. pages 1–8.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*. pages 2786–2792.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Repl4nlp Workshop at ACL*.

Liqiang Nie, Meng Wang, Luming Zhang, Shuicheng Yan, Bo Zhang, and Tat-Seng Chua. 2015. Disease inference from health-related questions via sparse deep learning. *IEEE Transactions on Knowledge and Data Engineering* 27(8):2107–2119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* .

Aaron Traylor, Nicholas Monath, Rajarshi Das, and Andrew McCallum. 2017. Learning string alignments for entity aliases. In *6th Workshop on Automated Knowledge Base Construction (AKBC)*.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* .

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with lstm. In *Proceedings of NAACL-HLT*. pages 1442–1451.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814* .

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *TACL* 4:259–272.

Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, pages 682–690.