PANLP at MEDIQA 2019: Pre-trained Language Models, Transfer Learning and Knowledge Distillation

Wei Zhu, Xiaofeng Zhou, Keqiang Wang, Xun Luo, Xiepeng Li, Yuan Ni, Guotong Xie
Pingan Health Tech, Shanghai, China
{zhuwei972, zhouxiaofeng824, wangkeqiang265, luoxun492, lixiepeng538,
niyuan442, xieguotong}@pingan.com.cn

Abstract

This paper describes the models designated for the MEDIQA 2019 shared tasks by the team PANLP. We take advantages of the recent advances in pre-trained bidirectional transformer language models such as BERT (Devlin et al., 2018) and MT-DNN (Liu et al., 2019b). We find that pre-trained language models can significantly outperform traditional deep learning models. Transfer learning from the NLI task to the RQE task is also experimented, which proves to be useful in improving the results of fine-tuning MT-DNN large. A knowledge distillation process is implemented, to distill the knowledge contained in a set of models and transfer it into an single model, whose performance turns out to be comparable with that obtained by the ensemble of that set of models. Finally, for test submissions, model ensemble and a re-ranking process are implemented to boost the performances. Our models participated in all three tasks and ranked the 1st place for the RQE task, and the 2nd place for the NLI task, and also the 2nd place for the QA task.

1 Introduction

There are three tasks in the MEDIQA 2019 shared tasks (see Ben Abacha et al. (2019) for details of the tasks). The first one, NLI, consists in identifying three inference relations between two sentences: *Entailment*, *Neutral* and *Contradiction*. The second one, RQE, requires one to identify whether one question entails the other, where the definition of *entailment* is that *a question A entails a question B if every answer to B is also a complete or partial answer to A*. The third task, QA, considers not only the identification of entailment for the asked question among a set of retrieved questions, but also the ranks of retrieved answers.

In this work, we demonstrate that we can achieve significant performance gains over traditional deep learning models like ESIM (Chen et al., 2016), by adapting pre-trained language models into the medical domain. Language model pre-training has shown to be effective for learning universal language representations by leveraging large amounts of unlabeled data. Some of the most famous examples are GPT-V2 (see Radford et al., 2019) and BERT (by Devlin et al., 2018). These are neural network language models trained on text data using unsupervised objectives. For example, BERT is based on a multi-layer bidirectional Transformer, and is trained on plain text for masked word prediction and next sentence prediction tasks. To apply a pre-trained model to specific NLU tasks such as tasks for MEDIQA 2019 shared tasks, we often need to fine-tune the model with additional task-specific layers using task-specific training data. For example, Devlin et al. (2018) show that BERT can be fine-tuned this way to create state-of-the-art models for a range of NLU tasks, such as question answering and natural language inference.

We also tryout a transfer learning procedure, where an intermediate model obtained on the NLI task is used to be fine-tuned on the RQE task. Although this procedure cannot consistently improve the dev set performance for all the models, it is proven to be beneficial on the test set by adding variety to the model pool.

To further improve the performance of single models, we implement a knowledge distillation procedure on the RQE task and the NLI task. Knowledge distillation distills or transfers the knowledge from a (set of) large, cumbersome model(s) to a lighter, easier-to-deploy single model, without significant loss in performance (Liu et al., 2019a; Tan et al., 2019). Knowledge distillation recently has attracted a lot of attentions. We believe it is interesting and of great importance to explore this method on the applications of the medical domain.

Proceedings of the BioNLP 2019 workshop, pages 380–388
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

For test submissions, model ensemble is used to obtain more stable and unbiased predictions. We only adopt a simple ensemble model, that is, averaging the class probabilities of different models. After obtaining test predictions, for the NLI and RQE task, simple re-ranking operations among pairs with the same premise are used to boost the performance metrics.

The rest of the paper is organized as follows. In Section 2 , we demonstrate our experiments on the three tasks. In Section 3, transfer learning from NLI to RQE is presented. Section 4 elaborates on the knowledge distillation and the corresponding experimental results. Section 5 and Section 6 present the model ensemble technique and the reranking strategies. Section 7 explains our submission records in detail. Section 8 concludes and discusses future work.

2 Pairwise Text Modeling

This section elaborates on the fundamental methods we used for the three tasks.

2.1 RQE

The RQE task, as a pairwise text classification task, defined here involves a premise $P = (p_1, p_2, ..., p_m)$ of m words, which is a medical question posted online, and a hypothesis $H = (h_1, h_2, ..., h_n)$ of n words, which is a standard frequently asked question that is collected to build a QA system, and aims to find a logical relationship R between P and H. For the RQE task, relationship R is either *true* or *false*, indicating whether the premise entails the hypothesis or not. We mainly experiment on two groups of models, one using fixed pre-trained embedding[1], the other employing pre-trained language models.

Traditional deep learning models typically use a fixed pre-trained word embedding to map words into low-dimensional vector space, and then use some kind of encoders to encode and pool the contexts of the premise to vector r_1 and hypothesis H to r_2. And the features provided to the classification layer is $concat(r_1, r_2, ||r_1 - r_2||, r_1 * r_2)$. (see Bowman et al., 2015) Then the classification output layer is usually a dense layer with soft-max output. We experiment with the following 4 traditional deep learning models. The first model, which will be called Weighted-Transformer-NLI model, encodes the sentences via a shared Weighted Transformer module (see Ahmed et al., 2017 for details). The second model, called RCNN-NLI, encodes the premise and hypothesis via the RCNN model (see Lai et al., 2015). The third model we consider, is the decomposable attention model by Parikh et al. (2016). The fourth model is the ESIM model by Chen et al. (2016), which is one of the most popular models in the natural language inference task. We will not elaborate on the specific architecture of the last two models since readers can refer to the original papers for details.

For the RQE task, the pre-trained language models we considered are as follows: (a) the original BERT models (both base and large models); (b) the Bio-BERT model by Lee et al. (2019) which is pre-trained on scientific literature in biomedical domain; (c) the Sci-BERT model by Beltagy et al. (2019) which is trained on academic papers from the corpus of *semanticscholar.org*; (d) MT-DNN models (see Liu et al., 2019b), which are based on BERT and go through a multi-task learning procedure on the GLUE benchmark. On top of the transformer encoders from the pretrained language model, we implement two kinds of output modules: (a) linear projection, which will be referred to as LP0, which is to take the hidden state corresponding to the first token $[CLS]$ of the sentence pair; (b) a more sophisticated classification module called stochastic answer network (henceforth SAN) proposed by Liu et al. (2017). Rather than directly predicting the entailment given the input, SAN maintains a state and iteratively refines its predictions.

When implementing the traditional deep learning models, the Glove embedding (Pennington et al., 2014) is used. Before training, we use the Unified Medical System (UMLS) provided by provided by the National Library of Medicine [2] to replace all the abbreviations (e.g., *IBS*) of a medical concept or entity to its full name, or to the same name that appears in the same pair. We tune the hyper-parameters on the dev set, and report the best performance obtained by each model in Table 1.

Among the four traditional models, RCNN-NLI performs the worst. Although a powerful model as shown in Ahmed et al. (2017),

[1] We will refer to this type of models as traditional deep learning models

[2] https://www.nlm.nih.gov/research/umls/

Model	valid acc
Weighted-Transformer-NLI	0.6821
RCNN-NLI	0.5530
Decomposable attention	0.6854
ESIM	0.7218
BERT base + linear projection	0.7815
BERT base + SAN	0.7119
BERT large + linear projection	0.7782
BERT large + SAN	0.7682
Bio-BERT + linear projection	0.4338
Bio-BERT + SAN	0.4305
Sci-BERT + linear projection	0.7547
Sci-BERT + SAN	0.5993
MT-DNN base + linear projection	**0.8378**
MT-DNN base + SAN	0.7715
MT-DNN large + linear projection	0.7881
MT-DNN large + SAN	0.7815

Table 1: performances of different models on the valid set of the RQE task.

Weighted-Transformer-NLI cannot perform very well on this dataset. The ESIM model performs the best among the four. However the traditional deep learning models cannot perform well enough when compared with the results on the Round 1 leader board. We believe the reasons are as follows. First, the dataset is relatively small, thus models like Weighted-Transformer-NLI will immediately over-fit. [3] Second, the distribution of training data for RQE task is different from the distributions of the dev and test data. We see most of the pairs in train set have approximately equal length, and there are 1, 445 pairs in which the premise and hypothesis are exactly the same. Meanwhile, in dev and test sets, the premise is usually much longer than the hypothesis.

When compared with traditional deep learning models, the pre-trained language models perform significantly better on the dev set. In addition, one can see that adding a sophisticated output module like SAN on top of the pre-trained language model tends to worsen the dev performance. Among all the BERT model family, the MT-DNN model (base model) performs best, and the original BERT base model performs slightly worse. Since the MT-DNN family are BERT models fine-tuned on GLUE benchmark via a multi-task learning mechanism, and in GLUE eight out of nine tasks are pairwise text modeling tasks, MT-DNN are more equipped to model pairwise text classification tasks on different domains than the original BERT model. And we can see that MT-DNN base performs better than MT-DNN large, which is in contradiction to the results on the GLUE benchmark reported in Liu et al. (2019b). Sci-BERT and Bio-BERT model does not perform well. We believe the reasons are that the Sci-BERT and Bio-BERT models share the same feature that they are trained on scientific literature, in which the language is more formal and rigid. However, texts in RQE is drawn from online questions from medical forums, thus Sci-BERT and Bio-BERT are not suitable for this task.

layers to freeze	valid acc
0	0.7782
1	0.8013
3	0.7914
6	0.7881
9	0.8179
10	0.8344
11	**0.8378**

Table 2: performances of the MT-DNN base model with linear projection, when different number of layers are freezed during fine-tuning on the RQE dataset

We also notice that freezing the lower bi-directional transformer layers of MT-DNN significantly improves the dev set accuracy. In Table 2,

[3] Readers can refer to Guo et al. (2019) for more detailed discussions on how transformer models performs unsatisfyingly on medium or small datasets, when directly trained from scratch.

Model	valid acc
ESIM (by Romanov and Shivade, 2018)	0.7440
InferSent (by Romanov and Shivade, 2018)	0.7600
BERT base + linear projection	0.8186
BERT base + SAN	0.8143
BERT large + linear projection	0.8229
BERT large + SAN	0.8280
Bio-BERT + linear projection	0.6824
Bio-BERT + SAN	0.6882
Sci-BERT + linear projection	**0.8466**
Sci-BERT + SAN	0.8251
MT-DNN base + linear projection	0.8265
MT-DNN base + SAN	0.8287
MT-DNN large + linear projection	0.8420
MT-DNN large + SAN	0.8327

Table 3: performances of different models on the valid set of the NLI task.

we can see that freezing 11 lower layers of the MT-DNN base performs best. During training of different models, even traditional deep learning models, we notice that a model can easily over-fit on the training set of RQE, fine-tuning the whole language model will introduce much bias into the model. Meanwhile freezing the lower layers can alleviate over-fitting and maintain the generalization ability of the pre-trained models.

2.2 NLI

For the NLI task, we are tasked to identify the relationship R between the premise and the hypothesis, which is among the following three: *entailment*, *neutral* or *contradiction*. Romanov and Shivade (2018) has done a thorough investigation on how traditional deep learning models like ESIM and InferSent perform on the original NLI datasets. Thus to save time, we only implement with pre-trained language models for this task.

The BERT based models we tried are the same as we investigate on the RQE datasets, whose results are reported in Table 3. It turns out, the BERT-based model significantly outperforms the traditional models. MT-DNN models still perform quite well, but the Sci-BERT with linear projection achieves the highest accuracy on the dev set. The Bio-BERT model still cannot achieve satisfying results. We find that models behave quite differently on NLI compared with the RQE datasets. First, on the NLI dataset, BERT large and the MT-DNN large, which is derived from BERT large, perform better than their base counterparts, BERT base and MT-DNN base. Second, during tuning the hyper-parameters, we find that freezing layers leads to performance loss. Third, the SAN output module does not lead to significant performance change except for Sci-BERT, whereas on the RQE dataset adding SAN module usually leads to significant performance loss.

2.3 QA

On the basis of the results obtained on RQE and NLI task, we found that the MT-DNN models outperform other pre-trained language models. Thus, with limited time, in the QA task we chose to directly look into the MT-DNN models on the QA datasets.

The QA task requires us not only give a binary label to an answer, but also rank the answers of the same questions. There are two perspectives of treating such a task: classification and regression. The classification model just distinguishes whether the question and the answer match, and the output of Softmax layer can be used to rank the answers. However, the regression model is able to predict the matching degree between questions and answers, and rank the answers according to the matching degree. The final result achieved is a combination of two models.

From the perspective of the classification model, answers with *ReferenceScore* less than 3 are given a *not entailment* label, and the rest are labeled *entailment*. The dataset obtained with this treatment is called the QA-C dataset. Table 4 reports the performance on the dev set. To align

Model	acc	Spearman's Rank Corr
MT-DNN base on QA-R	0.8248	0.1478
MT-DNN large on QA-R	0.8333	0.2054
MT-DNN base + linear projection on QA-C	0.7479	0.0557
MT-DNN base + SAN on QA-C	0.7607	-0.0108
MT-DNN large + linear projection on QA-C	0.8333	0.0803
MT-DNN large + SAN on QA-C	0.8120	0.2146

Table 4: performances of different models on the valid set of the QA task. Here accuracy is calculated on the whole dev set.

Model	dev acc
MT-DNN base	0.8378
MT-DNN base + transfer learning on NLI	0.8220
MT-DNN large	0.7881
MT-DNN large + transfer learning on NLI	0.7957

Table 5: The performance on the RQE dev set, when we apply transfer learning, compared with the performances obtained by directly fine-tuning the MT-DNN models on the RQE dataset.

with the leader board, we calculated accuracy and Spearman's Rank Correlation Coefficient (henceforth SRCC). As is shown in Table 4, BERT base can achieve accuracy of 0.7478 after fine-tuning. However, SRCC is 0.057, which is quite poor. The results demonstrate that a binary classification model helps us to get a fair accuracy score, but it omits all the ranking information like *ReferenceRank* and *ReferenceScore* from the original data. Thus the resulting model can not tell whether an answer is better than another. Bearing that in mind, we decided to introduce a related but different model to specialize in providing ranking information, while leave the accuracy metric to the classification model.

The new model we are introducing treats the task at hand as a regression task. For a sample data, the input is a pair composed of a query and an answer. The target value is the relevance score between the query and the answer, which is defined as follows:

$$score = ReferenceScore + 1/ReferenceRank. \quad (1)$$

The reciprocal of the *ReferenceRank* is used to enlarge the gaps of relevance scores among different answers. The dataset obtained with the above modification is called the QA-R dataset. The regression model is also built on the pre-trained language models by replacing the classification output module with a regression task header (see equation (2) of Liu et al., 2019b). Table 4 shows that we can obtain a huge bump on SRCC with the regression model. The best dev SRCC we can obtain is 0.148, which is the result of fine-tuning the MT-DNN large model. With a threshold for the relevance score, we can also get the classification label from the regression label. After adjusting the threshold, we can also get accuracy of 0.8247. Thus, we can conclude that the regression model works better in capturing the ranking information without reducing the accuracy of the model.

By observing the SRCC obtained at each epoch during training, we can see the following phenomenon: SRCC can improve from 0.125 to 0.273 after a single epoch, and suddenly drop to 0.023 on the next one. SRCC seems to be quite unstable, which will be problematical when making predictions for the unknown test set. This is a problem that we fail to solve at the end of competition and requires further investigations.

3 Transfer learning

We also experimented with transfer learning for the RQE task. The procedure is to first fine-tune a MT-DNN model on the NLI dataset for a certain number of epochs, then the obtained model will further be fine-tuned on the RQE dataset. Our motivation is that first fine-tuning on the NLI task can help the pre-trained language model to adapt to the medical domain, thus making the training on RQE more stable. Table 5 reports that after the transfer learning procedure, MT-DNN base model performs worse, but it makes the MT-DNN large model perform slightly better.

4 knowledge distillation

In this section, we experiment on the idea of knowledge distillation (Hinton et al., 2015), to further boost the performance of single models. We implement knowledge distillation on each task separately.[4] The procedure is as follows:

- train a set of models on each tasks. Following Liu et al. (2019a), the set of models are: MT-DNN base and MT-DNN large, with different dropout rates ranged in $0.1, 0.3, 0.5$ for the task specific output layers, while keeping the hyper-parameters of lower BERT encoders the same with those in the previous section.

- ensemble the above models to get a label model (Ratner et al., 2018)[5]. This so-called label model is constructed by modeling a generative model over all the label functions, i.e., the single models, to maximize the log likelihood, give the label matrix (Ratner et al., 2017). The label model is a generalization of the so-called teacher model in (Liu et al., 2019a), where the teacher model is simply the average of class probabilities.

- The end model (or called the student model by Liu et al., 2019a) is trained on the soft targets given out by the label model. Here, training on the soft targets means the cross-entropy loss is averaged with the class probabilities as weights.

- Inference is the same for end model with other normal models.

In Table 6, we can see that knowledge distillation can significantly improve the performance on the NLI task, and can even achieve better results than model ensemble. However, on the RQE task, knowledge distillation cannot perform better than model ensemble, but still outperforms the best single model.

5 Ensemble

Since the test set is small, one single model is too biased to achieve great results on the test dataset. Ensemble learning is an effective approach to improve model generalization, and has been used to achieve new state-of-the-art results in a wide range of natural language understanding (NLU) tasks (Devlin et al., 2018, Liu et al., 2017).

For the MEDIQA 2019 shared task, we only adopt a simple ensemble approach, that is, averaging the softmax outputs from different models, or different runs or epochs of the same model, and makes prediction based on these averaged class probabilities. All our submissions follow this ensemble strategy. [6]

6 Re-ranking strategies for the NLI and RQE tasks

The previous sections demonstrate how deep learning models perform on the task datasets. However, in order to obtain more competitive results, one could adopt some simple heuristics.

For the NLI task, after observing the task datasets, we can see that one premise is grouped with three different hypothesis, and the latter are labeled with *entailment*, *neutral* and *contradiction* respectively. We call the three pairs with the same premise a group. Our sentence pair model does not know the idea of groups, thus the labels corresponding to the maximum class probabilities obtained by soft-max layer can conflict with one another. For example, two pairs in the same group may both be labeled as *entailment*. To eliminate the above conflicts, we adopt the following heuristic post-processing procedure:

- obtain the label predictions directly from the softmax output. If there is no conflict in a group, accept the predictions. Otherwise, in this group:

- Give the *contradiction* label to the pair with the highest score for this label

- Between the remaining two pairs, decide which one should get the *neutral* label via the scores for this label

[4] Liu et al. (2019a) extends the knowledge distillation to multi-task learning setting, which is a direction we need to explore in future work.

[5] There are alternative terminologies for knowledge distillation. We mainly follow Ratner et al. (2018).

[6] We definitely can try some more sophisticated ensemble methods, but we believe experimenting different learning strategies like MTL and knowledge distillation is more meaningful for research purpose, and is in alignment with the objective of the MEDIQA 2019 share tasks.

Model	NLI	RQE
best single model	0.8466	0.8378
model ensemble	0.8638	0.8477
knowledge distillation	0.8667	0.8411

Table 6: Comparison of performances on the dev sets, among the best single model, ensemble model and the model obtained by knowledge distillation.

- the remaining pair get the *entailment* label

For the RQE task, since the label is binary, and the number of pairs in a group in this task varies, the re-ranking heuristic is a little different, which is elaborated as follows.

- obtain the score of the *entailment* label from the model
- for each group, rank the pairs by their scores.
- denote the number of pairs in a group as n, then we directly label the last $max(1, [n/2] - 1)$ as negative pairs. and the top pair as positive pair
- For the rest of pairs, we choose a threshold t, if the score of a pair is higher than t, it is labeled *entailment*, otherwise it is labeled as *not entailment*. We choose the threshold to obtain the highest accuracy on the dev set

7 Submission results

This section discusses the submission results on the leader boards.

First, let us look at the submission history on the RQE task (presented here in Table 7). The first submission is a single MT-DNN base model trained only on the training data, with re-ranking. On the second submission, we add the available dev set in, and re-train all the models. The ensemble of a MT-DNN base and a MT-DNN large after re-ranking push the test accuracy to 0.736. Then we tryout transfer learning on the third run, two runs of MT-DNN large, which go through the transfer learning process described in Section 3, achieves 0.745 after re-ranking. Adding the end model after knowledge distillation to the combination in the third run makes the performance drops slightly to 0.740. For the final submission, we just ensemble all the models available, and achieve 0.749 on the test set, which ranks the first on the RQE task.

Table 8 presents the submission records on the NLI task. On the first submission, we experiment the model obtained by knowledge distillation, which obtains 0.865 on accuracy. The second submission, we use a single MT-DNN large fine-tuned on the train set and post-processed for re-ranking. The accuracy is 0.916 for this submission. Then the ensemble of four models, the 8-th epoch of 2 different runs of MT-DNN large, the 10-th epoch of 2 different runs of Sci-BERT, achieves an accuracy of 0.946 after re-ranking. The final submission combines MT-DNN large, Sci-BERT, MT-DNN large after knowledge distillation, obtains 0.966 after re-ranking, which ranks the third on the leader board.

For the QA task, the first two submissions are based on a single MT-DNN large model fine-tuned on QA-R data set, chosen from two different training epochs. The first submission with accuracy of 0.73 is chosen because in this epoch of training, we achieved the best Spearman's rho result on the dev dataset; Similarly,the second submission with accuracy of 0.733 is chosen at the epoch where we achieved best ACC result on the dev dataset. From the third round, we started applying ensemble strategy by considering some well performing epochs at different runs together. The two submissions with accuracy of 0.774 and 0.777 are the results of different processing strategies: max score and mean score. According to the results obtained, we find that ”max score” strategy performs slightly better on SRCC, while ”mean score” works better on ACC.

8 Conclusion and discussions

To conclude, we have shown that domain adaptation with the pre-trained language models achieves significant improvement over traditional deep learning models on the MEDIQA 2019 shared tasks. We also experimented transfer learning from the NLI task to the RQE task. Knowledge distillation obtains a single model which significantly outperforms the single models trained

Submission No.	test acc	details
1	0.675	1 * MT-DNN base (trained on train set) + re-rank
2	0.736	1 * MT-DNN base + 1 * MT-DNN large + re-rank
3	0.745	2 * MT-DNN large (TL) + re-rank
4	0.740	1 * MT-DNN large (KD) + 2 * MT-DNN large (TL) + re-rank
5	0.749	2 * MT-DNN base + 2 * MT-DNN large (TL) + 1 * MT-DNN large (KD) + 1 * MT-DNN large + re-rank

Table 7: The submission results on the RQE task. Multiplication symbol ”*” here means multiple runs or epochs of the same model (with different random seed). ”TL” means the model go through transfer learning on the NLI task. ”KD” means the model is obtained via knowledge distillation. Without declaration, all the models here are trained on the train and dev set.

Submission No.	test acc	details
1	0.865	1 * MT-DNN large (KD)
2	0.916	1 * MT-DNN large (on train set) + re-rank
3	0.946	2 * MT-DNN large + 2 * Sci-BERT + re-rank
4	0.966	4 * MT-DNN large + 4 * Sci-BERT + 2 * MT-DNN large (KD) + re-rank

Table 8: The submission records on the NLI task. Multiplication symbol ”*” here means multiple runs or epochs of the same model (with different random seed). ”KD” means the model is obtained via knowledge distillation. Without declaration, all the models here are trained on the train and dev set.

Submission No.	test acc	test Spearman’s rho	details
1	0.730	0.236	MT-DNN large (epoch with best training SRCC)
2	0.736	0.204	MT-DNN large (epoch with best training ACC)
3	0.774	0.22	MT-DNN large ensemble(rank by max socre)
4	0.777	0.18	MT-DNN large ensemble(rank by mean socre)
5	0.772	0.204	MT-DNN large ensemble(rank by mean socre)

Table 9: The submission results on the QA task.

in the usual way. Our submission results, although including model ensemble and re-ranking, are strong demonstration of the power of language model pre-training, transfer learning and knowledge distillation.

However, due to the limited time and the fact that we participate all three tasks at once, we haven’t exhaustively explore all the possible ways to boost the performance on the leader board, e.g., utilizing external sources such as medical knowledge bases to rule out false positive answers. Multi-task learning is also a direction that we need to pay more attention to.

In addition, the heuristics adopted in the re-ranking strategies resemble the relevance ranking task (Huang et al., 2013), where one compares different pairs in a group to obtain the final decisions. Due to time constraint, we didn’t implement a pairwise relevance ranking model on top of the MT-DNN model, but this research direction will be investigated by us in future work.

References

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2017. Weighted Transformer Network for Machine Translation. *arXiv e-prints*, page arXiv:1711.02132.

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. SciBERT: Pretrained Contextualized Embeddings for Scientific Text. *arXiv e-prints*, page arXiv:1903.10676.

Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 2019. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the BioNLP 2019 workshop, Florence, Italy, August 1, 2019*. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced LSTM for Natural Language Inference. *arXiv e-prints*, page arXiv:1609.06038.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-Transformer. *arXiv e-prints*, page arXiv:1902.09113.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, page arXiv:1503.02531.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *arXiv e-prints*, page arXiv:1901.08746.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding. *arXiv e-prints*, page arXiv:1904.09482.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-Task Deep Neural Networks for Natural Language Understanding. *arXiv e-prints*, page arXiv:1901.11504.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic Answer Networks for Machine Reading Comprehension. *arXiv e-prints*, page arXiv:1712.03556.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. *arXiv e-prints*, page arXiv:1606.01933.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *arXiv e-prints*, page arXiv:1711.10160.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2018. Training Complex Models with Multi-Task Weak Supervision. *arXiv e-prints*, page arXiv:1810.02840.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from Natural Language Inference in the Clinical Domain. *arXiv e-prints*, page arXiv:1808.06752.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual Neural Machine Translation with Knowledge Distillation. *arXiv e-prints*, page arXiv:1902.10461.