Pairwise Loss with Max Sampling: A Neural Meta-Architecture for Answer Selection

Wei Yang
David R. Cheriton School of Computer Science, University of Waterloo

Abstract

We introduce a novel neural network "meta-architecture" called Pairwise Loss with Max Sampling (PLMS) that transforms *any* Siamese network into a new design that improves classification accuracy without increasing model complexity. Pairwise loss applies hinge loss to the margin between positive and negative examples and max sampling judiciously chooses training pairs that lie along the decision boundary. PLMS is based on the reproduction of previous work on noise-contrastive estimation. Demonstrating the robustness of the generalized approach, we are able to boost the accuracy of an older but much simpler CNN to a level rivaling the best (but substantially more complex) models on a standard benchmark question answering dataset. A hierarchical composition of neural network architectures yields simpler models that are easier to understand.

1 Introduction

From the rich literature on applying neural networks to NLP tasks that has emerged over the last several years, researchers have identified architectural patterns that are effective for certain classes of tasks. For example, the Siamese architecture (Bromley et al., 1994) is often applied to tasks involving predictions over pairs of sentences (e.g., paraphrase detection and answer selection). We refer to this as a "meta-architecture", in that Siamese designs manifest in a variety of networks that all share common elements.

In this paper, we generalize the noise-contrastive sampling framework proposed by Rao et al. (2016) into a novel meta-architecture: we propose an approach where Pairwise Loss with Max Sampling (PLMS) can be applied to *any* Siamese architecture. Pairwise loss applies hinge loss to the margin between positive and negative examples and max sampling judiciously chooses training pairs that lie along the decision boundary. The result is a design that substantially improves on the base model *without increasing its complexity* (e.g., number of parameters).

We illustrate pairwise loss with max sampling on the answer selection task in question answering. Given a natural language question q and a candidate set of sentences $\{c_1, c_2, \ldots c_n\}$, the goal of answer selection is to identify sentences that contain the answer (Tellex et al., 2003).

We generalize previous work into our PLMS meta-architecture. First, we implement the model by He et al. (2015) and Severyn and Moschitti (2015) and reproduce the effectiveness of the original paper. Then we add our PLMS meta-architecture to the model of He et al. (2015) and reproduce the results in Rao et al. (2016), which either rival or beat the best published results on TrecQA, one of the standard benchmarks for answer selection.

Next, we applied our PLMS meta-architecture to a mediocre architecture (Severyn and Moschitti, 2015) and on a new answer selection dataset, InsuranceQA, which is not investigated by Rao et al. (2016) and brings new challenges. Experiments show that our techniques are able to substantially improve upon each base model. As a demonstration of the robustness of the PLMS meta-architecture, we show that by applying PLMS to a simpler neural network (Severyn and Moschitti, 2015), we can achieve competitive results with the best models in the literature in some cases. We argue that thinking about neural network architectures in this hierarchical manner yields simpler models that are easier to understand.

2 Related Work

There have been numerous applications of neural networks to answer selection in question answering (Yu et al., 2014; Iyyer et al., 2014; Severyn and Moschitti, 2015; Yang et al., 2016; dos Santos et al., 2016; He and Lin, 2016; Shen et al., 2017; Tay et al., 2017; He et al., 2016; Yin and Schütze, 2017; Bian et al., 2017; Rao et al., 2017; Zhang et al., 2017; Sequiera et al., 2017). These papers are too numerous to detail here, but a common theme is an increase over time in the complexity of the models, incorporating innovations such as multi-perspective and attention mechanisms. From the ACL wikipedia page that summarizes the performance improvements (ACL, 2018), we see that increasingly complex models are contributing to smaller and smaller gains on benchmark datasets. Furthermore, complex models exhibit larger parameter space and can be easier towards overfitting, which makes it challenging to tell whether the minor improvements come from more intelligent model designs or exhaustive tunings by fitting to a local optimum. A natural and important question to us, as the question answering community, to rethink thoroughly is: are those complicated model architectures necessary for the answer selection task?

In contrast to increasing model complexity, we propose a meta-architecture design that introduces no additional complexity to the base models to replace more complex "one-shot" models. The advantage of this approach is that we can build on simple, well-understood, and robust base models to achieve end-to-end effectiveness that is comparable to the best (complex) models available.

Our terminology borrows from the learning-to-rank (LtR) literature, which distinguishes pointwise, pairwise (Huang et al., 2013), and listwise methods (Li, 2011). Work on LtR, however, typically views pointwise and pairwise techniques are being distinct. Our contribution here is a deterministic approach to convert any Siamese neural pointwise model into a pairwise model—in other words, a generalization bridge between pointwise and pairwise methods.

As mentioned above, many of our ideas have been explored by Gutmann and Hyvärinen (2010) and Rao et al. (2016). Our work can be viewed as a generalization of Rao et al. (2016), in that their application of noise contrastive estimation was intertwined with the base model itself—whereas we

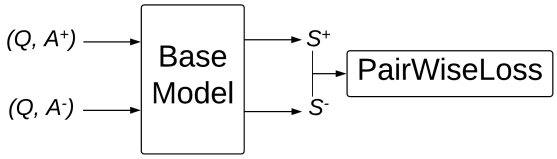

Figure 1: Architecture of our pairwise model with input of on (Q, A^+),(Q, A^-) pairs.

are able to clearly identify a meta-architecture and a deterministic set of transformations to improve on a base Siamese model. Rao et al. (2016) explored two base models, one of which we share, and therefore our experiments serve as a replication study. Finally, our application of PLMS over a mediocre neural network model substantially improves its accuracy and supports the generality and robustness claim of our approach.

3 Approach

The PLMS meta-architecture takes as input a Siamese neural network (a base model) and applies a series of deterministic transformations to create a new model that yields higher accuracy. This is shown in Figure 1, which we detail below.

3.1 Pairwise Loss

Abstractly, we consider a base model as a black box function $F(x, y; \theta) \rightarrow \hat{s} \in \mathbb{R}$. For question answering the input would be (question, candidate answer) pairs and the model predicts whether the candidate is correct. F is typically trained with respect to some loss $\mathcal{L}_f(\hat{s}, s)$. In learning-to-rank parlance, this is called a pointwise model.

In the conversion from pointwise to pairwise loss, we take the base model apply inference on two pairs of input:

$$F(Q, A^+; \theta) \rightarrow \hat{s}^+$$
$$F(Q, A^-; \theta) \rightarrow \hat{s}^-$$

where A^+ and A^- are positive and negative examples for question Q. From this, we can define a new margin hinge loss as follows:

$$\mathcal{L} = \max \sum_{min}^{max} (0, m - \hat{s}^+ + \hat{s}^-) + \lambda \|\theta\|^2$$

where our goal is to maximally discriminate positive from negative examples (m is a free parameter, and we empirically set $m = 1$ in all of our

experiments). We apply standard L2 regularization based on θ, the underlying parameters of the base model.

3.2 Negative Sampling

The next obvious question concerns how we sample positive and negative answers. To reduce the complexity of the design space, our training regime always begins with a (Q, A^+) pair from the training data. The question then becomes, how do we generate (Q, A^-) samples?

A simple but naïve baseline would be to sample (Q, A^-) randomly from the training data, and indeed that forms a baseline. We call the setup as the pairwise loss with random sampling (PLRS). In the experiments we will show PLRS can achieve similar performance to the base model with pointwise loss and no sampling strategy but is not likely to lead to significantly improvement. We argue that it is mainly due to the low quality of negative samples obtained by PLRS that cannot bring enough semantic matching signal to the domain-specific task.

However, we can do better by modifying the sampling strategy: our intuition is that if wish to maximally discriminate between positive and negative examples, then it would be most effective select negative examples that are the "closest" to the positive examples (but on the other side of the decision boundary). We call this "max sampling" and define closeness in terms of cosine similarity between the latent representations of the base model, i.e., $\text{sim}(\Lambda(A^+), \Lambda(A^-))$, where $\Lambda(\cdot)$ is the latent representation of the candidates. The max sampling process of negative samples can be formulated as follows:

$$NegS(Q;\theta) = \text{argmax}_{A^-}\ \text{sim}(\Lambda(A^+), \Lambda(A^-))$$

In addition, Rao et al. (2016) mentions a mixed sampling strategy that selects half of the samples from each strategy, which appears to be effective in some experiments (Tay et al., 2017). However, in most circumstances, including the experiments of Rao et al. (2016), we do not see significant benefit of this setup. Thus, we do not present the results of this strategy.

In practice, we run a forward inference pass (using the current model) and use the features at the fully-connected layer as the latent representation. Thus the cosine similarity is usually taken as the similarity function considering the distribution of feature space. However, if the base model does not provide an analogous latent representation, we can back off to lexical similarity (e.g., tf-idf).

Thus, the revised learning problem can be formulated as follows:

$$\text{argmin}_\theta \sum_{(Q,A^+)} \sum_{(Q,A^-)} \max(0, m - \hat{s}^+ + \hat{s}^-) + \lambda\|\theta\|^2$$

where the number of negative examples is a hyperparameter.

3.3 Base Models

In this work we consider two base models, described below:

SM-CNN (Severyn and Moschitti, 2015) is a simple, well-understood, and robust model, and can be viewed as a baseline Siamese design—our goal is to show that PLMS can substantially boost the accuracy of simple architectures. In our implementation, we show that there are actually SM-CNN four variants (see Section 4.1). Empirically, we select the setup that performs best in the experiments.

MP-CNN (He et al., 2015) achieved state-of-the-art results on sentence similarity tasks when it was first published (ACL, 2018): although other models have bested it since, it remains competitive. Our goal is to show that PLMS can "squeeze" more performance out of an already good model. MP-CNN is admittedly more complex than SM-CNN, which replaces hand engineering of features with a substantial amount of architecture engineering. From Table 2 we can see SM-CNN is much simpler than MP-CNN in term of number of parameters. This represents a recent trend in designing architectures that capture different perspectives for semantic feature extraction (Feng et al., 2015; Tymoshenko et al., 2016; Ma et al., 2017; Choi et al., 2017). With our reimplementation, we reproduce the results showed in Rao et al. (2016) on answer selection and achieve better results than originally reported.

4 Experimental Setup

4.1 Word Embeddings

Experiments in this paper are based on our own implementations, where we attempted to faithfully replicate the original are available. For SM-CNN, we initialized the embedding layer from pretrained 50-dimensional embedding on English

	TrecQA			WikiQA			InsuranceQA			
	Train	**Dev**	**Test**	**Train**	**Dev**	**Test**	**Train**	**Dev**	**Test1**	**Test2**
# of questions	1229	82	100	873	126	243	12887	1000	1800	1800
# of question-answer pairs	53417	1148	1517	8672	1130	2351	-	500,000	900,000	900,000
Avg # of pos. answers / query	5.2	2.6	3.0	1.2	1.1	1.2	1.4	1.4	1.5	1.4
Avg % of pos. answers / query	12.0	19.3	18.7	12.0	12.3	12.4	-	0.3	0.3	0.3
Avg length of questions	8			6			7			
Avg length of answers	28			25			95			

Table 1: Statistics of all datasets for experiments.

Wikipedia and the AQUAINT corpus using the skip-gram model following Severyn and Moschitti (2015); out-of-vocabulary words were randomly initialized, sampled from the uniform distribution $U[-0.25, 0.25]$. For MP-CNN, we used pretrained 300-dimensional GloVe embedding on the Common Crawl; out-of-vocabulary words were randomly initialized, sampled from the normal distribution $N(0, 0.01)$. According to Kim (2014), considering whether word embeddings are pretrained and trainable, there are four types of setups: random, static, non-static, multichannel. We tried all four types, and got similar results for the last three setups. But for simplicity, we only report results with multichannel embedding for SM-CNN and non-static embedding for MP-CNN since these configurations achieve the best results empirically. Note that the original SM-CNN implementation followed the static strategy.

4.2 Hyperparameters

For the pointwise models we optimized cross entropy loss using Adam (Kingma and Ba, 2014) and for the pairwise models we optimized the margin hinge loss described in Section 3.1 using Adadelta (Zeiler, 2012). We fixed margin m to 1 and the dropout rate to 0.5 in all experiments. We tuned the remaining hyperparameters on the validation set and selected the best parameter combination to evaluate on test set.

To examine the impact of our sampling techniques, we compared the max sampling approach described in Section 3.1 with random sampling (i.e., simply choosing random negative examples). In both cases we set the number of negative samples to $n = 8$. Different from Rao et al. (2016), we stored the latent representation of each (Q, A) pair in the memory and dynamically update them every mini-batch instead of every epoch, which allowed us to capture the most accurate representation of positive and negative answers during training. To further examine the effects of sampling, we applied max and random sampling to the pointwise models as well—by simply augmenting the training data during each mini-batch (without taking advantage of the pairwise loss).

We implemented all base models and their PLMS versions using PyTorch[1] and we will make all of our code publicly available once our paper gets accepted.

4.3 Datasets

Our evaluations were conducted on the three widely-used benchmark datasets for question selection: TrecQA (Wang et al., 2007), WikiQA (Yang et al., 2015), and InsuranceQA (Feng et al., 2015). Table 1 shows statistics of these three datasets. The numbers of parameters for each model and each dataset are showed in Table 4.1. We can see that adding our PLMS based architecture does not change the total parameters of the base model, and thus did not increase additional time complexity. Note that the same model has different number of parameters for different datasets because the vocabulary across datasets varies.

TrecQA. TrecQA was based on the Text REtrieval Conference (TREC) Question Answering track (8-13) data and was packaged by Yao et al. (2013). As pointed out by Rao et al. (2016), TrecQA has "raw" and "clean" versions, and both of them are applied as the benchmark datasets to evaluate models (He and Lin, 2016; Miao et al., 2016; Wang et al., 2016b; dos Santos et al., 2016). Our experiments are based on the raw version. From Table 1 we can the TrecQA provides the most positive answers for each question compared to the other two datasets, which provides more semantic information for the positive matching, and further-

[1] http://pytorch.org/

Model	Dimension of Embeddings	Dataset	# of Parameters
(PLMS +) MP-CNN	300	WikiQA	15,083,702
		TrecQA	25,035,302
		InsuranceQA	13,958,402
(PLMS +) SM-CNN	50	WikiQA	2,943,835
		TrecQA	6,338,935
		InsuranceQA	2,702,935

Table 2: Overview of models for experiments

more, as will be showed in the experiment section, guide the PLMS based architecture to select the negative samples better.

WikiQA. Similar to TrecQA, WikiQA is also a sentence-level dataset for open domain question answering, extracted from real users Bing query and a snippet of a Wikipedia article retrieved by Bing. For WikiQA, consistent with Yang et al. (2015), we removed questions without positive answers. From Table 1 We see that WikiQA has a smaller proportion of candidate answers in the training set but more questions and question-answer pairs in the test set than TrecQA, which brings the challenge of leveraging the matching signals in various domains with relatively less knowledge.

InsuranceQA. In InsuranceQA, all questions are extracted from the insurance domain and one question a large number of negative answers. The dataset is composed of one training set, one validation set, and two test sets. From Table 1, we see that the answer sentences in InsuranceQA are much longer than TrecQA and WikiQA. Unlike the other two datasets that each question is provided with a list of specific candidate answers, the questions in InsuranceQA share the same candidate pool (potentially larger) to find the correct answer. Empirically, we randomly sampled 50 candidate negative answers from the answer pool with 24981 sentences generated by Feng et al. (2015). More details regarding to InsuranceQA can be found in Feng et al. (2015)'s paper.

In summary, the unique characteristics of InsuranceQA bring new challenges to our approach. First, the candidate pool is much larger and the percentage of positive answers are much less that require a efficient and effective sampling technique. Meanwhile, the much longer sentences can lead to difficulties in training time increases and sentence representation learning, which provides a good testbed for examining the generalizability of our approach.

5 Results

5.1 Results on TrecQA & WikiQA

As previously described, by applying PLMS two different models and three different datasets, we demonstrate the generalizability of our meta-architecture.

Experimental results are shown in Table 3, where we report mean average precision (MAP) and mean reciprocal rank (MRR), the two standard metrics for characterizing accuracy on this task. To capture the inherent variability in training neural networks (Reimers and Gurevych, 2017), we report the mean [min, max] results from five trials with different random seeds.

Our implementations achieve accuracies comparable to the original papers, which gives us confidence they are correct. Note the implementation with the pointwise loss and no sampling is exactly the reproduction of the base model. With the full PLMS meta-architecture, we improve substantially over the base models. For TrecQA, PLMS on MP-CNN achieves scores that are at least as good as the highest scores reported in the literature, summarized in the bottom half of Table 3. Applying PLMS over SM-CNN yields gains over the base model as well, which shows the robustness of our meta-architecture. In fact, applying PLMS over a very simple model like SM-CNN outperforms many models that are far more complex. It is worth emphasizing that PLMS improves accuracy *without* increasing the complexity of the base model (i.e., number of parameters).

Findings are generally consistent with the WikiQA dataset. Although we do not beat the best-reported scores in the literature, PLMS applied to MP-CNN yields accuracies that are com-

[2]The results are obtained by pretraining on SQuAD dataset, which we argue are not fairly comparable to other models listed above.

Base Model	Loss	Sampling	TrecQA MAP	TrecQA MRR	WikiQA MAP	WikiQA MRR
Severyn and Moschitti (2015)			0.746	0.808	-	-
SM-CNN	Point	-	0.759 [0.746,0.769]	0.811 [0.799,0.819]	0.654 [0.641,0.662]	0.679 [0.675,0.684]
		Random	0.731 [0.720,0.738]	0.799 [0.790,0.809]	0.649 [0.643,0.659]	0.668 [0.661,0.678]
		Max	0.737 [0.731,0.743]	0.810 [0.803,0.817]	0.652 [0.641,0.662]	0.674 [0.669,0.680]
	Pair	Random	0.751 [0.735,0.765]	0.823 [0.815,0.830]	0.667 [0.660,0.674]	0.690 [0.684,0.695]
		Max	0.763 [0.755,0.771]	0.833 [0.830,0.836]	0.679 [0.662,0.688]	0.702 [0.699,0.704]
He et al. (2015)			0.762	0.830	0.693	0.709
MP-CNN	Point	-	0.756 [0.749,0.766]	0.818 [0.802,0.832]	0.688 [0.677,0.693]	0.705 [0.691,0.714]
		Random	0.761 [0.758,0.762]	0.819 [0.803,0.826]	0.668 [0.658,0.677]	0.687 [0.672,0.693]
		Max	0.765 [0.757,0.772]	0.832 [0.829,0.839]	0.676 [0.665,0.684]	0.692 [0.684,0.698]
	Pair	Random	0.768 [0.759,0.775]	0.837 [0.828,0.844]	0.705 [0.701,0.709]	0.712 [0.701,0.718]
		Max	**0.780 [0.771,0.787]**	**0.838 [0.830,0.846]**	0.706 [0.702,0.709]	0.710 [0.699,0.719]
Previous Work						
Yang et al. (2016)			0.750	0.811	-	-
Tan et al. (2016)			0.753	0.830	-	-
He and Lin (2016)			0.758	0.821	0.709	0.723
Rao et al. (2016)			**0.780**	0.834	0.701	0.718
Wang et al. (2016a)			0.737	0.821	0.734	0.741
Tymoshenko et al. (2016)			-	-	**0.742**	**0.759**
dos Santos et al. (2016)			-	-	0.689	0.696
Tay et al. (2017)			0.770	0.825	0.712	0.727
Yin and Schütze (2017)			-	-	0.712	0.723
Wang et al. (2017)			-	-	0.718	0.731
Min et al. (2017) (no pretraining)			-	-	0.630	0.645
Min et al. (2017) [2]			-	-	0.832	0.845

Table 3: Results on TrecQA and WikiQA datasets.

parable. Note that some of the papers that perform well on WikiQA do not report results on TrecQA, so it is unclear the extent to which those models generalize. In the last row of Table 3, we see the results from Min et al. (2017) with pretraining on the SQuAD (Rajpurkar et al., 2016) dataset outperform other approaches by about 10 absolute points. We argue that this is not a fair comparison by comparing approaches with and without pretraining, which is confirmed by the large drop of the same approach when pretraining is removed. Meanwhile, learning the effects of pretraining and domain adaption is out of our scope in this paper. Overall, results of PLMS applied to two different base models across two different datasets illustrates the generality of our meta-architecture.

We note that PLMS appears to improve MRR more than it does MAP, and MRR difference between the base models are smaller than they are for MAP. This makes sense since our loss attempts to maximize the margin between positive and negative examples—and MRR is only concerned about the appearance of the first correct answer.

In addition to the full PLMS meta-architecture, Table 3 also breaks down the impact of each component: the pairwise loss and the sampling techniques. In general, we see that max sampling is more effective than random sampling (except for WikiQA with MP-CNN, where the MRR is very close). Although it is possible to apply sampling techniques with a pointwise loss, the approach is not effective because it is unable to exploit the contrast between positive and negative examples.

5.2 Results on InsuranceQA

We also report the results of applying PLMS to SM-CNN on the InsuranceQA dataset. Due to the time constraint, we are not able to fill in the performance numbers of the MP-CNN and its PLMS variant at the submission time. The inefficiency is mainly due to the fact that MP-CNN is much more time-consuming than SM-CNN and the InsuranceQA dataset is around 12 times larger than TrecQA and 75 times larger than WikiQA. We argue that reporting the SM-CNN results alone here would not significantly alter our conclusion since we already demonstrated the generalizability of PLMS over TrecQA and WikiQA. We will add those numbers at the publication time.

The evaluation on InsuranceQA is measured on the accuracy (or precision@1) by following pre-

Our Implementation	Dev	Test1	Test2
SM-CNN	0.614	0.612	0.603
PLRS+SM-CNN	0.612	0.615	0.609
PLMS+SM-CNN	0.640	0.639	0.626
Previous Work			
Bag-of-word	0.319	0.321	0.322
Bendersky et al. (2010)	0.527	0.551	0.508
Feng et al. (2015)	0.618	0.628	0.592
Feng et al. (2015) with GESD	0.654	0.653	0.610
Tan et al. (2016)	0.684	0.681	0.622
Wang et al. (2016a)	**0.699**	0.701	0.628
dos Santos et al. (2016)	0.687	**0.717**	**0.644**

Table 4: Results on InsuranceQA

vious publications (Feng et al., 2015; dos Santos et al., 2016). From Table 4, we can see PLMS can bring consistent improvements to the base SM-CNN model. The performance of PLMS on SM-CNN is already close to the state-of-the-art numbers in Test2 set. For the other Test1 set, our approach is still about 8 absolute point behind, while we believe the PLMS on MP-CNN will achieve more competitive results.

5.3 Per-Question Analysis

To gain further insights on how PLMS improve on the base models, we conduct a per-question analysis on TrecQA dataset as shown in Figure 2. We visualize the differences between PLMS and the base models SM-CNN and MP-CNN for each question in term of the average precision score (MAP). From Figure 2, we can clearly see that PLMS provides stable and consistent improvement on both SM-CNN and MP-CNN. It improves 31 questions while hurts 13 questions on SM-CNN. The average improved scores is also larger than those of the bad-performing questions for both base models.

It's also worth to note the two figures are not in a similar shape because the base models behave differently. Overall, these two figures confirm the robust effectiveness of PLMS.

Furthermore, we present additional sample analysis from TrecQA for the best-performing question 67 ("*where are the company conde nast's headquarters?*") and the worst-performing question 14 ("*when did the khmer rouge come into power?*") in Table 5. The column "Score" denotes the matching score of the question-answer pair given by the two models: MP-CNN and PLMS + MP-CNN.

Comparing sample 1, 2 and 3 in Table 5, we can clearly see the benefits of PLMS based models over the base neural networks: base model fails to differentiate the sentences with an exact matched phrase 'conde nast' and the semantic matching signal of location such as 'commercial neighborhood' and 'time square', while the PLMS based model enlarges the scale of output scores so as to distinguishes the right answer from the tricky negative candidates. However, comparing sample 4 and 5, we can see that PLMS based method will still suffer from the semantic-oriented matching problem since both candidates contains an exact matched phrase 'khmer rouge' and a year.

6 Conclusions

This paper introduces the notion of a neural meta-architecture which takes a base Siamese neural model and automatically generates a better model by applying two ideas: pairwise loss to maximize the margin between positive and negative examples and max sampling to judiciously select training examples to exploit the loss. Experiments show that both techniques are complementary and necessary, and that our meta-architecture is robust and general. We empirically show for answer selection that hierarchically designing neural networks in this manner yields designs that are simpler, easier to understand, and yet achieve accuracies at least as good as far more complex models. From the experiments on various QA benchmark datasets, we demonstrate the robustness and generalizability of our approach.

7 Future Work

There are a few potential research directions for future exploration.

More diverse network structures: Besides Siamese type of neural networks, we are looking for more diverse types of networks that PLMS can be applied on. By designing the way to deal with the specific task according to pair-wise loss function, we can investigate PLMS is still effective for other network structures and research areas besides answer selection.

Tradeoffs between model efficiency and effectiveness: By changing the loss function for the existing models without adding model complexity, PMLS achieved better results comparing with the base model. The main insight of PLMS is that finding the most challenging negative samples is critical to improve model performance.

However, this usually requires an exhaustive

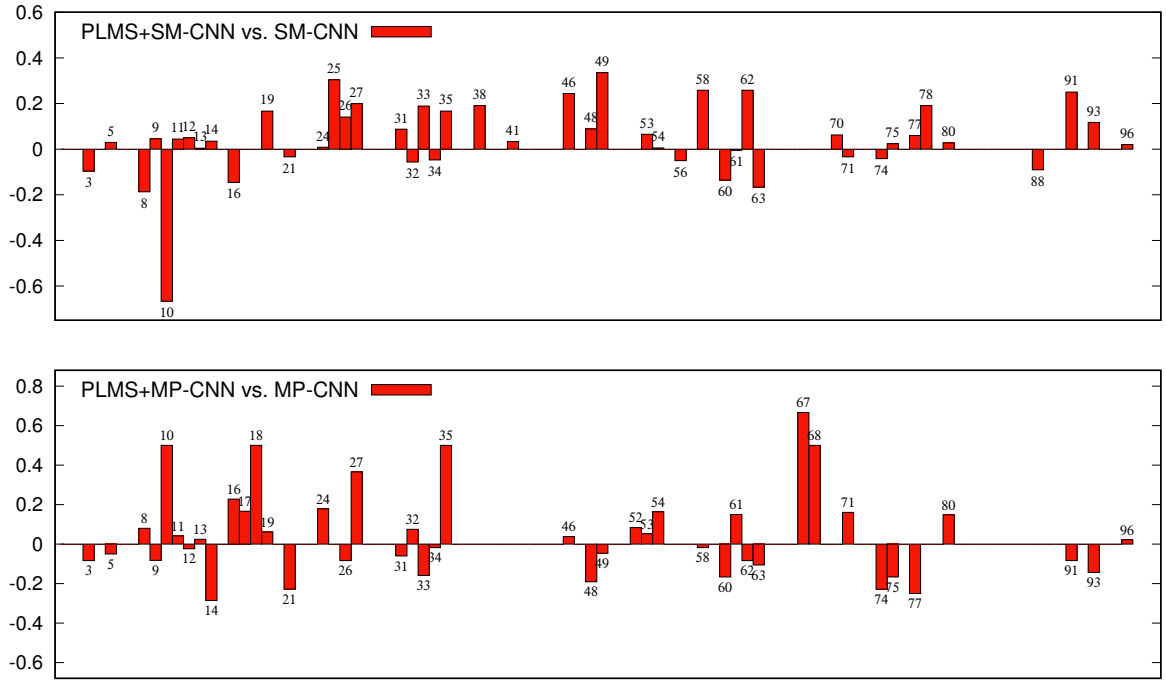

Figure 2: Visualization of per-question performances' difference on TrecQA between base model and PLMS based architecture

ID	QID	Candidate answer	Label	Score	
				Base	PLMS
1	67	other issues were bandied about in the brand-new conde nast headquarters in times square in manhattan .	1	0.029	1.449
2		since conde nast signed a deal for the building nearly three years ago , times square has become one of the most sought-after commercial neighborhoods in the city .	0	0.042	1.390
3		conde nast will be moving from its old-money environs on the east side , within easy reach of brooks brothers , paul stuart and patroon , to the hurly-burly of times square .	0	0.032	0.547
4	14	the defectors were key players in the khmer rouge 's rule after the maoist revolutionaries won a civil war in 1975 .	1	0.023	0.182
5		1996 : government announces khmer rouge breakup .	0	0.005	0.349

Table 5: Sample Analysis on TrecQA

search over the answer pool. In many real scenarios, like recommendation, the answer pool can contain thousands or even millions of candidates (i.e., users or items in recommendation setting). How to tradeoff efficiency to support fast learning in a large-scale setting while maintaining competitive effectiveness at th wato

References

ACL. 2018. Question answering (state of the art). http://www.aclweb.org/aclwiki/index.php?title=Question_Answering_(State_of_the_art). Accessed: 2018-03-15.

Michael Bendersky, Donald Metzler, and W Bruce Croft. 2010. Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 31–40. ACM.

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference*

on Information and Knowledge Management, pages 1987–1990.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 209–220.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 813–820. IEEE.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Hua He, Kevin Gimpel, and Jimmy J Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1576–1586.

Hua He and Jimmy J Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2016)*, pages 937–948.

Hua He, John Wieting, Kevin Gimpel, Jinfeng Rao, and Jimmy Lin. 2016. Umd-ttic-uw at semeval-2016 task 1: Attention-based multi-perspective convolutional neural networks for textual similarity measurement. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1103–1108.

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 Conference on Information & Knowledge Management*, pages 2333–2338. ACM.

Mohit Iyyer, Jordan L Boyd-Graber, Leonardo Max Batista Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 633–644.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *arXiv preprint arXiv:1408.5882*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hang Li. 2011. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

Rongqiang Ma, Jian Zhang, Miao Li, Lei Chen, and Jin Gao. 2017. Hybrid answer selection model for non-factoid question answering. In *Asian Language Processing (IALP), 2017 International Conference on*, pages 371–373. IEEE.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. *arXiv preprint arXiv:1702.02171*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1913–1916. ACM.

Jinfeng Rao, Hua He, and Jimmy Lin. 2017. Experiments with convolutional neural network models for answer selection. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1217–1220. ACM.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 338–348, Copenhagen, Denmark.

Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR, abs/1602.03609*.

Royal Sequiera, Gaurav Baruah, Zhucheng Tu, Salman Mohammed, Jinfeng Rao, Haotian Zhang, and Jimmy Lin. 2017. Exploring the effectiveness of convolutional neural networks for answer selection

in end-to-end question answering. *arXiv preprint arXiv:1707.07804*.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1200.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 464–473.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Enabling efficient question answer retrieval via hyperbolic neural networks. *CoRR*, abs/1707.07847.

Stefanie Tellex, Boris Katz, Jimmy Lin, Gregory Marton, and Aaron Fernandes. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 41–47, Toronto, Canada.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1268–1278.

Bingning Wang, Kang Liu, and Jun Zhao. 2016a. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1288–1297.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing Conference on Computational Natural Language Learning Joint Meeting (EMNLP-CoNLL 2007)*, volume 7, pages 22–32.

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

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016b. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 287–296. ACM.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2013–2018.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. 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*, pages 858–867.

Wenpeng Yin and Hinrich Schütze. 2017. Task-specific attentive pooling of phrase alignments contributes to sentence matching. *arXiv preprint arXiv:1701.02149*.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. In *arXiv preprint arXiv:1412.1632*.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Haotian Zhang, Jinfeng Rao, Jimmy Lin, and Mark D Smucker. 2017. Automatically extracting high-quality negative examples for answer selection in question answering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 797–800. ACM.