Answer Selection in Question Answering using Convolutional Neural Network

Md Mustafizur Rahman
School of Information
University of Texas at Austin

Outline

- Problem statement
- Related work
- Proposed architecture
- Experimental setup
- Result analysis
- Future work

Question Answering Task (QA)

- Question answering in general formulated as **answer selection** task

- Problem Statement
  - Given a question **q** and a set of candidate answers **{a_1, a_2, …, a_n},** the job is to search for the best candidate answer **a_i**
  - Or the system provides **a ranked list** of answer with **best** answer at **top** and the **worst** one at the **bottom** of the list

Types of Question Answering

- Two types of QA Tasks
  - Factoid QA task
    - E.g. Who is the president of USA?
    - TREC QA dataset
  - Non-factoid QA task
    - Covering all non-factoid QA?
    - E.g. How to update *Gedit* in Linux from terminal?
    - Stackoverflow.com dataset

Related Work

- Most of the state-of-the-art deep learning approaches are for the factoid QA
- Wang et al. [1] proposed to use Bidirectional LSTM to generate the vector representation of question and answers but this method needs BM25 feature to beat the non-deep learning baselines
- Convolutional Deep Neural Network (CDDN) and additional word overlapping feature like uni-gram are used in [2] to outperform the baselines
- Yang et al [4] proposed to combine value shared CNN with attention network for factoid QA task

Proposed Architecture

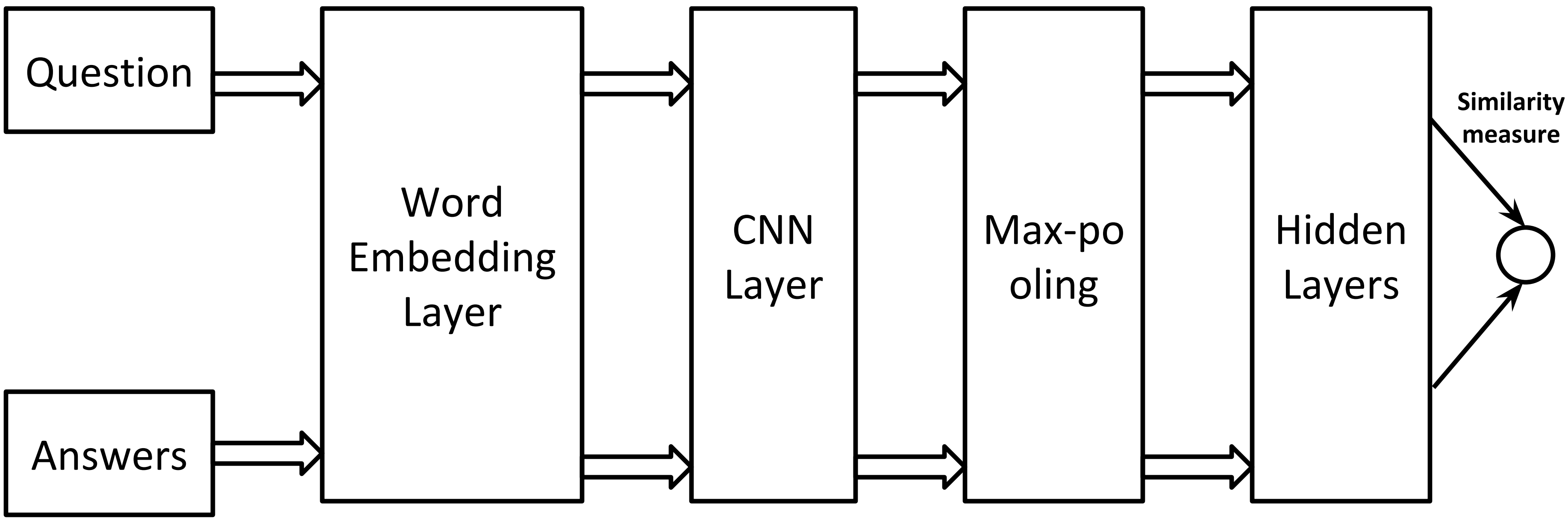


Figure 1: Proposed QA-CNN model (Architecture 1)

Variants of QA-CNN architecture

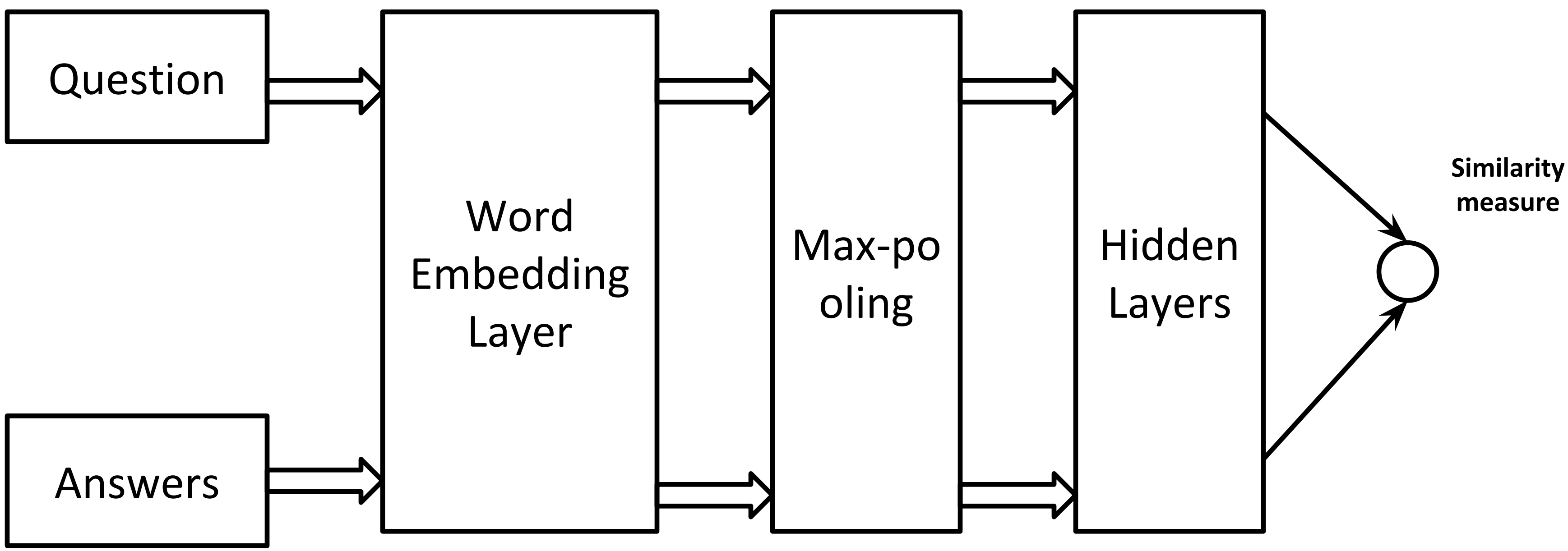
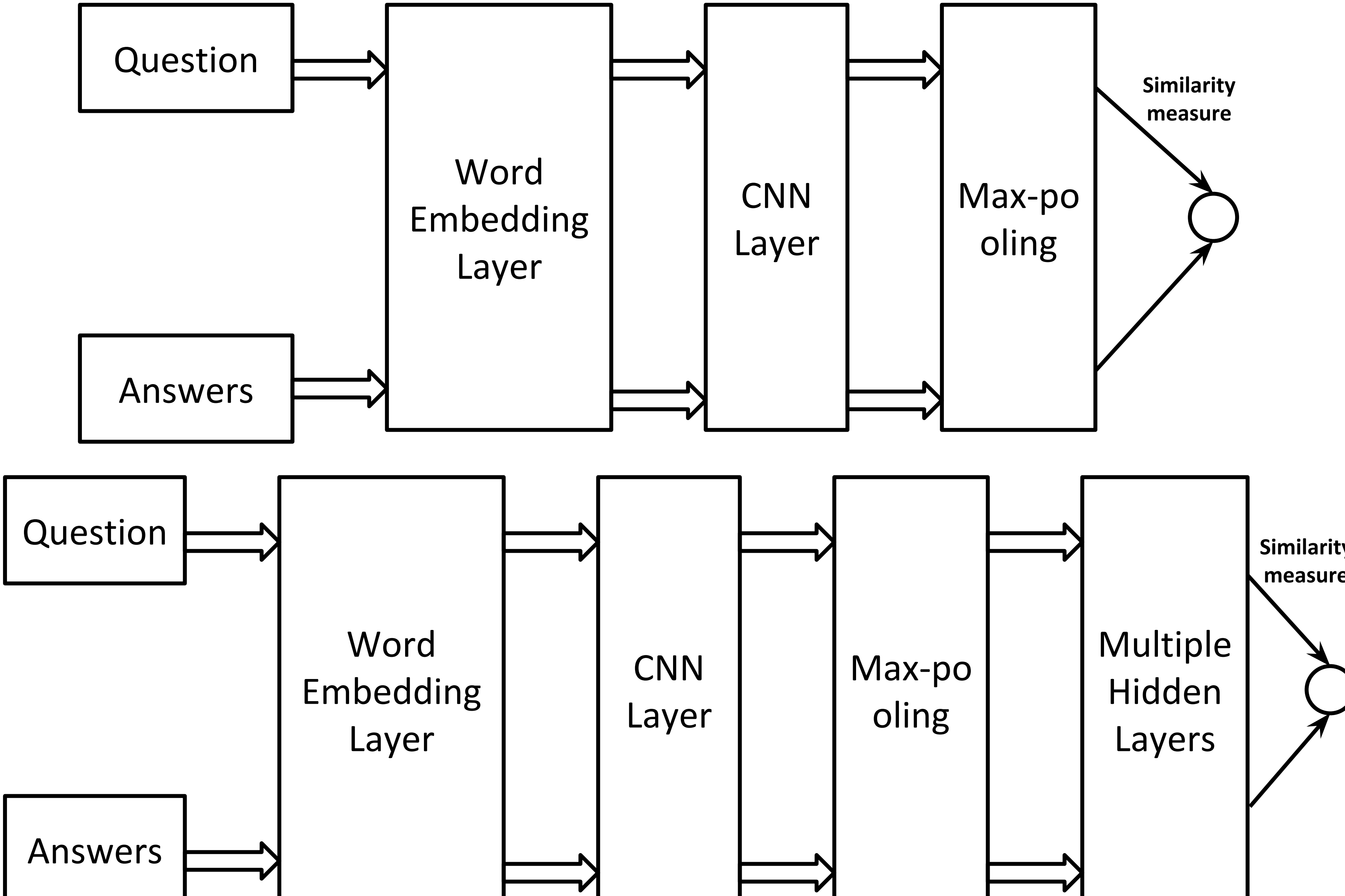


Figure 2: Embedding Based Model (Architecture 2)

Variants of QA-CNN architecture



Training Algorithm

- QA-CNN generates the vector representation of the question and the positive answer and the negative answer V_{Q}, V_{A+} and V_{A-}
- If $sim(V_Q, V_{A+}) - sim(V_Q, V_{A-}) \leq m$
  – Update the parameters of QA-CNN
- Loss function: Hinge Loss

$$L = \max\{0, m - sim(V_Q, V_{A+}) + sim(V_Q, V_{A-})\}$$

Similarity Measure

Table 1: Several similarity measures used in [3]. γ, *c* and *d* are user defined parameters.

Similarity Measure	Expression
cosine	$sim(x,y) = \frac{xy^T}{\|\|x\|\|\|\|y\|\|}$
polynomial	$sim(x,y) = (\gamma xy^T + c)^d$
Sigmoid	$sim(x,y) = tanh(\gamma xy^T + c)$
RBF	$sim(x,y) = \exp(-\gamma\|\|x-y\|\|^2)$
euclidean	$sim(x,y) = \frac{1}{1+\|\|x-y\|\|}$
exponential	$sim(x,y) = \exp(-\gamma\|\|x-y\|\|_1)$
Manhattan	$sim(x,y) = \frac{1}{1+\|\|x-y\|\|_1}$
GESD	$sim(x,y) = \frac{1}{1+\|\|x-y\|\|} * \frac{1}{1+\exp(-\gamma(xy^T+c))}$
AESD	$sim(x,y) = \frac{0.5}{1+\|\|x-y\|\|} * \frac{1}{1+\exp(-\gamma(xy^T+c))}$

Dataset

- Insurance QA dataset [3]

	Questions	Answers	Word Count
Train	12887	18540	92095
Dev	1000	1454	7158
Test 1	1800	2616	12893
Test 2	1800	2593	12905

- Training instance is a tuple of <Question, Positive answer, Negative answer>
- Sampling of negative answer are performed from a pool of 500 answers [3]
  - To reduce the computational complexity

Performance Metrics

- Precision at rank 1 (P@1)

$$P@1 = (\frac{1}{N}\sum_{i=1}^{N}\delta(r(A^{+}) = 1))$$

- Mean Reciprocal Ranking (MRR)

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r(A^{+})}$$

Parameter Settings

- Dimension of Word Embedding:
  - 100, 300 & 500
- Learning rate: 0.001
- Hyper-parameter: $\beta_1 = 0.9$, $\beta_2=0.999$ and $\varepsilon=10^{-08}$
- Margin: 0.05
- Number of filters in CNN: 500
- Types of filters: 4 {2, 3, 5 and 7}

Experimental Analysis
Effect of Word Embedding Dimension

Table 2: Architecture 1
(QA-CNN Model)

Number of dimensions	Test set	P@1	MRR
100	**Dev**	0.284	0.284
	Test 1	0.280	0.280
	Test 2	0.250	0.250
300	**Dev**	0.225	0.225
	Test 1	0.215	0.216
	Test 2	0.206	0.205
500	**Dev**	0.220	0.220
	Test 1	0.201	0.202
	Test 2	0.179	0.178

Experimental Analysis
Effect of CNN Layer

Table 2: Architecture 1
(QA-CNN Model)

Number of dimensions	Test set	P@1	MRR
100	**Dev**	0.284	0.284
	Test 1	0.280	0.280
	Test 2	0.250	0.250
300	**Dev**	0.225	0.225
	Test 1	0.215	0.216
	Test 2	0.206	0.205
500	**Dev**	0.220	0.220
	Test 1	0.201	0.202
	Test 2	0.179	0.178

Table 3: Architecture 2
(Embedding Based Model)

Number of dimensions	Test set	P@1	MRR
100	**Dev**	0.344	0.344
	Test 1	0.320	0.321
	Test 2	0.297	0.298
300	**Dev**	0.321	0.321
	Test 1	0.317	0.317
	Test 2	0.303	0.304
500	**Dev**	0.297	0.297
	Test 1	0.299	0.299
	Test 2	0.285	0.285

Experimental Analysis
Effect of Similarity Measure

Table 4: Performance for Architecture 1 (QA-CNN model)

Similarity	Dev		Test 1		Test 2	
metrics	P@1	MRR	P@1	MRR	P@1	MRR
cosine	0.004	0.005	0.002	0.002	0.001	0.002
euclidean	0.337	0.449	0.323	0.447	0.295	0.410
GESD	0.284	0.284	0.280	0.280	0.250	0.250
exponential	0.257	0.363	0.260	0.378	0.235	0.342
RBF	0.060	0.111	0.056	0.103	0.049	0.096

Table 5: Performance for Architecture 2 (Embedding Based Model)

Similarity	Dev		Test 1		Test 2	
metrics	P@1	MRR	P@1	MRR	P@1	MRR
cosine	0.004	0.008	0.002	0.008	0.002	0.006
euclidean	0.333	0.429	0.307	0.415	0.297	0.395
GESD	0.344	0.344	0.320	0.321	0.297	0.298
exponential	0.339	0.423	0.318	0.414	0.307	0.391
RBF	0.161	0.235	0.150	0.224	0.135	0.208

Experimental Analysis
Effect of Hidden Layers

Table 6: Effect of hidden layer on QA-CNN Model

Similarity	Dev		Test 1		Test 2	
metrics	P@1	MRR	P@1	MRR	P@1	MRR
QA-CNN-No-Hidden-Layer (Architecture 3)	0.266	0.266	0.280	0.280	0.247	0.247
QA-CNN-One-Hidden-Layer (Architecture 1)	0.284	0.284	0.280	0.280	0.250	0.250

Future Work

- Applying this model on
  - Stack overflow.com question answering task
    - Challenging as answer contains both text and image
- Adding more convolutional and hidden layers with different activation function
- Instead of CNN, we can also try RNN, LSTM etc.

References

- [1] Wang, Di, and Eric Nyberg. "A long short-term memory model for answer sentence selection in question answering." *ACL, July* (2015).
- [2] Yu, Lei, et al. "Deep learning for answer sentence selection." *arXiv preprint arXiv:1412.1632* (2014)
- [3] Feng, Minwei, et al. "Applying deep learning to answer selection: A study and an open task." *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015.
- [4] Yang, Liu, et al. "aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model." *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016.