# Explain by Evidence: An Explainable Memory-based Neural Network for Question Answering

**Quan Tran[1], Nhan Dam[2], Tuan Lai[3], Franck Dernoncourt[1]**
**Trung Le[2], Nham Le[4] and Dinh Phung[2]**
[1]Adobe Research  [2]Monash University
[3] University of Illinois at Urbana-Champaign  [4] University of Waterloo
{qtran, franck.dernoncourt}@adobe.com
{nhan.dam, trunglm, dinh.phung}@monash.edu
tuanml2@illinois.edu, nhamlevan@gmail.com

## Abstract

Interpretability and explainability of deep neural networks are challenging due to their scale, complexity, and the agreeable notions on which the explaining process rests. Previous work, in particular, has focused on representing internal components of neural networks through human-friendly visuals and concepts. On the other hand, in real life, when making a decision, human tends to rely on similar situations and/or associations in the past. Hence arguably, a promising approach to make the model transparent is to design it in a way such that the model explicitly connects the current sample with the seen ones, and bases its decision on these samples. Grounded on that principle, we propose in this paper an explainable, evidence-based memory network architecture, which learns to summarize the dataset and extract supporting evidences to make its decision. Our model achieves state-of-the-art performance on two popular question answering datasets (i.e. TrecQA and WikiQA). Via further analysis, we show that this model can reliably trace the errors it has made in the validation step to the training instances that might have caused these errors. We believe that this error-tracing capability provides significant benefit in improving dataset quality in many applications.

## 1 Introduction

Interpretability of neural networks is an active research field in machine learning. Deep neural networks might have tens if not hundreds of millions of parameters (Devlin et al., 2019; Liu et al., 2019a) organized into intricate architectures. The sheer amount of parameters and the complexity of the architectures largely prevent human to directly make sense of which concepts and how the network truly learns. The comparative lack of explainable intuition behind deep neural networks might hamper the development and adoption of those models. In certain scenarios, prediction accuracy alone is not sufficient (Caruana et al., 2015; Lapuschkin et al., 2019). For example, as discussed in (Zhang et al., 2018b; Zhang et al., 2018a), it is difficult to trust a deep model even if it has high test set performance given the inherent biases in the dataset. Thus, we argue that interpretability is perhaps one of the keys to accelerate both the development and adoption of deep neural networks.

There have been many successful attempts from the research community to make sense of deep models' prediction. These attempts can be broadly categorized into several classes. One of the major classes concerns with the network visualization techniques, for example, visual saliency representations in convolutional models (Simonyan et al., 2013; Sundararajan et al., 2017). For recurrent neural networks (RNN), Karpathy et al. (2015) focused on analyzing and visualizing the RNN to explain its ability to keep track of long-range information.

The visualization-based methods, although achieving great successes, still operate on a very high level of abstractions. It requires a great deal of machine learning knowledge to make use of those visualizations. Thus, these techniques are not always useful for a broader audience, who might not have the machine learning expertise. Looking back at classic machine learning models, one class of models stands out as being very intuitive and easy to understand: the instance-based learning algorithms. The $k$-nearest

neighbors algorithm, a prime example, operates on a very human-like assumption. To elaborate, if the current circumstances are similar to that of a known situation in the past, we may very well make this decision based on the outcome of the past decision. We argue that this assumption puts the interpretability on a much lower level of abstraction compared to the visualization methods. If somehow our model can learn how to link the evidences from the training data to the prediction phase, we will have a direct source of interpretability that can be appreciated by a broader audience.

The $k$-nearest neighbors algorithm, as an instance-based method, might not be a deep neural network technique; however, there have been many papers in the deep model literature inspired by or related to this method. A notable example is the neural nearest neighbors network (Plötz and Roth, 2018). Moreover, there is a class of problems with strong links to $k$-nearest neighbors: few-shot learning. It is from the two major papers in the few-shot learning literature, the prototypical network (Snell et al., 2017) and the matching network (Vinyals et al., 2016), we find a potential realization for our ideas.

In few-shot learning, it is possible to learn the support from each of the instances from the support set to the current prediction; however, such approach is infeasible when the training data get larger. Inspired by the techniques discussed in (Ravi and Larochelle, 2017), we apply a training-level data summarizer based on the neural Turing machine (NTM) (Graves et al., 2014) that reads the dataset and summarizes (or writes) it into a few meta-evidence nodes. These meta-evidence nodes, in turn, lend support to each of the prediction similar to a few-shot learning model. The parameters of the NTM are jointly trained with other parameters of the network. Our final model not only has great predictive power and achieves state-of-the-art results on two popular answer selection datasets, but also shows a strong "error-tracing" capability, in which the errors in the validation set can be traced to the sources in the training set.

To summarize, our contributions in this work are twofold. First, we propose a novel neural network model that achieves state-of-the-art performance on two answer selection datasets. Second, we show the utility of the error-tracing capability of our model to find the noisy instances in the training data that degrades the performance on the validation data. This capability might be very useful in real-life machine learning scenarios where the training labels are noisy or the inter-annotator agreement is low.

## 2 Proposed Framework

Question answering (or answer selection) is the task of identifying the correct answer to a question from a pool of candidate answers. It is an active research problem with applications in many areas (Tay et al., 2018a; Tayyar Madabushi et al., 2018; Lai et al., 2018; Rao et al., 2019; Lai et al., 2020). Similar to most recent papers on this topic (Tay et al., 2018b; Lai et al., 2019; Garg et al., 2020), we cast the question answering problem as a binary classification problem by concatenating the question with each of the candidate answers and assigning positive label to the concatenation containing the correct answer.

In most supervised learning scenarios, performing a full distance calculation between the current data point and every training data point would be computationally intractable. To overcome this burden, we propose a **memory controller** based on NTM to summarize the dataset into **meta-evidence nodes**. Similar to NTM, the controller is characterized by **reading** and **writing mechanisms**. Assume that we provide the controller with $K$ cells $e_1, \ldots, e_K$ in a memory bank (i.e. to store $K$ support/evidence vectors), and let us denote the $t$-th data point as $x^t$ (obtained by using a pretrained embedding model to embed the concatenation of a question and a candidate answer), the memory controller works as follows.

**Writing mechanism.** The writing mechanism characterizes how the controller updates its memory given a new data point. To update the memory, however, we first need an indexing mechanism for writing. Instead of using the original indexing of the NTM, we adopt the simpler indexing procedure from the memory network, which has been proven to be useful in this task (Lai et al., 2019). At time step $t$, for each incoming data point $x^t$, we compute the attention weight $w_{e_i^t}$ for the support vector $e_i^t$:

$$w_{e_i^t} = \frac{\exp\left(x^t \cdot e_i^t\right)}{\sum_{k=1}^{K} \exp\left(x^t \cdot e_k^t\right)}. \tag{1}$$

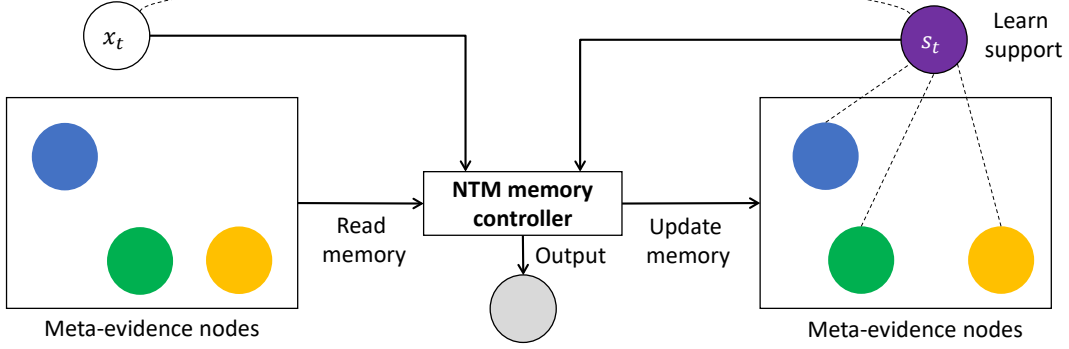From these attention weights, we find the writing index for an input $x^t$ by maximizing the cosine

Figure 1: The information flow of our model.

similarity between $x^t$ and the evidence vectors:

$$k_{\text{best}}^t = \arg\max_i w_{e_i^t}. \tag{2}$$

With the writing index found, we compute the memory update weight via a gating mechanism:

$$g^t = \epsilon \times \sigma\left(\left(W_g \cdot e_{k_{\text{best}}}^t + b_g\right) + \left(W_g \cdot x^t + b_g\right)\right), \tag{3}$$

where $\epsilon$ is a scalar, $\sigma$ is sigmoid function, and $W_g$ and $b_g$ are learnable parameters. The hyperparameter $\epsilon$ prevents the outliers to break the memory values. The memory update at time step $t$ is formalized as:

$$e_{k_{\text{best}}}^{t+1} = g^t \cdot x^t + \left(1 - g^t\right) \cdot e_{k_{\text{best}}}^t. \tag{4}$$

**Reading mechanism.** The reading mechanism characterizes how the controller uses its memory and the current input to produce an output. Instead of reading one memory cell, we aim to learn the support of all meta-evidence nodes. Thus, the weighted sum is used to create a support vector $s^t$:

$$s^t = \sum_{k=1}^{K} \left(w_{e_k^t} \cdot e_k^{t+1}\right) \tag{5}$$

We then incorporate the original input with the support vector $s^t$ to produce the negative/positive class probabilities $P(x^t)$ as follows:

$$P(x^t) = \text{softmax}\left(W_p\left(s^t + x^t\right) + b_p\right). \tag{6}$$

The overall information flow of our model is visualized in Figure 1. Our formulation draws inspiration from the NTM and the memory network. Our indexing algorithms in writing and reading mechanisms are similar to the memory network, which is simpler than the NTM. However, the memory network only stores intermediate computation steps in the memory, and these memories can be considered as internal layers of the network. Our memory, on the contrary, is external and not trained, only updated by the writing mechanism. In this regard, the memory bank of our model is more similar to the NTM.

## 3 Experimental Results

### 3.1 Question answering performance

In this subsection, we present our core results on two most popular datasets for answer selection: WikiQA (Yang et al., 2015) and TrecQA (Wang et al., 2007). Due to space constraint, details of these datasets are described in the Appendix. Similar to previous work, we use two standard measures for the task: mean average precision (MAP) and mean reciprocal rank (MRR). Our models make use of the RoBERTa contextual embedding (Liu et al., 2019b), pretrained on the ANSQ dataset (Garg et al., 2020). For our model, we vary the number of memory cells from 2 to 64. The base configuration with 2 memory cells mimics the prototypical network with one cell for each prototype class representation. Table 1 shows our

| Model | TrecQA | | WikiQA | |
|---|---|---|---|---|
| | MAP | MRR | MAP | MRR |
| RoBERTa + Evidence Memory (2) (ours) | 0.949 | 0.982 | 0.925 | 0.940 |
| RoBERTa + Evidence Memory (16) (ours) | 0.954 | 0.982 | **0.936** | **0.952** |
| RoBERTa + Evidence Memory (32) (ours) | **0.961** | **0.993** | 0.933 | 0.945 |
| RoBERTa + Evidence Memory (64) (ours) | 0.949 | 0.974 | 0.929 | 0.941 |
| RoBERTa + CETE (2020) | 0.936 | 0.978 | 0.900 | 0.915 |
| RoBERTa + ANSQ Transfer (2020) | 0.943 | 0.974 | 0.920 | 0.933 |
| BERT + GSAMN + Transfer (2019) | 0.914 | 0.957 | 0.857 | 0.872 |
| IWAN + sCARNN (2018) | 0.829 | 0.875 | 0.716 | 0.722 |
| Compare-Aggregate (2017) | 0.821 | 0.899 | 0.748 | 0.758 |

Table 1: Question answering performance.

model and the baselines' performance. All our model's configurations outperform the previous state-of-the-art models.[1] Increasing the number of memory cells beyond the basic 2 cells - one for each class - clearly helps. The performance peaks at 16 or 32 cells depending on the dataset.

## 3.2 Error-tracing performance

One of the main motivations behind our evidence-based model is the ability to interpret the output of the neural network. It is hard to quantify the interpretability of different models, however. To create a benchmark for interpretability, we look for a potential application of interpretability in real-life development of a deep neural network.

Data collection is one of the most important parts of a machine learning model's development cycle. In many cases, nevertheless, the collected data is not always clean and consistent, either due to errors made by annotators or equivocal data points. For example, the popular Switchboard Dialog Act dataset (Stolcke et al., 2000) only has 84% inter-annotator agreement. Thus, we would like to test how well different models help in identifying noisy instances in the dataset.

Our model naturally learns the most supportive group of instances given a new instance, and thus, we can easily use this information to trace from an error in validation to a group of training instances. Ideally, we will need to test all the training samples of that group, but that would quickly make the number of samples we need to check get out of control. Hence, we rely on heuristics: from the most relevant group, we only test the top $k$ most similar instances (by cosine distance in the embedding space). To create a noisy dataset given our current QA datasets, we randomly swap 10% the labels in each training set.[2] We then calculate the percentage of errors in validation that the model can correctly trace back to the training set perturbation. For quantitative benchmark, we compare our proposed model with the best baseline (i.e. the RoBERTa + ANSQ transfer model) and the top $k$ most similar representations.

| Model | TrecQA | | WikiQA | |
|---|---|---|---|---|
| | Top 1 | Top 3 | Top 1 | Top 3 |
| Random | 0.10 | 0.27 | 0.10 | 0.27 |
| RoBERTa + ANSQ Transfer (2020) | 0.19 | 0.50 | 0.20 | 0.59 |
| RoBERTa + Evidence Memory (16) (ours) | 0.53 | 0.81 | **0.51** | 0.90 |
| RoBERTa + Evidence Memory (32) (ours) | **0.80** | **0.88** | **0.51** | **0.93** |

Table 2: Error-tracing precision.

Table 2 shows the error-tracing performance of the model compared to the baseline. Our best model shows strong error-tracing capability and outperforms the baseline by a wide margin. On both datasets, our model can trace roughly 90% of the errors to the perturbed data points. This experiment clearly shows that forcing a model to provide direct evidences helps in identifying noisy training instances.

---

[1]Due to space constraint, more details about the baselines can be found in the Appendix.

[2]We tested with other percentages, but we deem 10% to provide the best compromise. Lowering the error rate, the model makes very few mistakes, thus the statistics are unreliable. Increasing the error rate, the model performs too badly, then the tracing performance would be almost perfect, but not meaningful.

# 4   Conclusion

In this paper, we propose a novel neural network architecture that not only achieves state-of-the-art performance on popular QA datasets, but also shows strong error-tracing performance, which we argue will be of great benefits to real-life machine learning applications. In the future, we would like to apply the model on different noisy user-generated datasets to test and further improve its interpretability.

# References

Weijie Bian, Si Kan Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In *The Conference on Information and Knowledge Management (CIKM)*.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *AAAI Conference on Artificial Intelligence*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Tuan Manh Lai, Trung Bui, and Sheng Li. 2018. A review on deep learning techniques applied to answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2132–2144, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Tuan Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2019. A gated self-attention memory network for answer selection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5953–5959, Hong Kong, China, November. Association for Computational Linguistics.

Tuan Manh Lai, Trung Bui, and Nedim Lipka. 2020. Isa: An intelligent shopping assistant. *arXiv preprint arXiv:2007.03805*.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8.

Md Tahmid Rahman Laskar, Xiangji Huang, and Enamul Hoque. 2020. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5505–5514.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

Tobias Plötz and Stefan Roth. 2018. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems*, pages 1087–1098.

Jinfeng Rao, Linqing Liu, Yi Tay, Wei Yang, Peng Shi, and Jimmy Lin. 2019. Bridging the gap between relevance matching and semantic matching for short text similarity modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5370–5381, Hong Kong, China, November. Association for Computational Linguistics.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018a. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 583–591.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018b. Multi-cast attention networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Harish Tayyar Madabushi, Mark Lee, and John Barnden. 2018. Integrating question classification and deep learning for improved answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3283–3294, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1274–1283, New Orleans, Louisiana, June. Association for Computational Linguistics.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

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 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32.

Yi Yang, Wen tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *The Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018a. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836.

Quanshi Zhang, Wenguan Wang, and Song-Chun Zhu. 2018b. Examining CNN representations with respect to dataset bias. In *Thirty-Second AAAI Conference on Artificial Intelligence*.