# End to End Long Short Term Memory Networks for Non-Factoid Question Answering

Daniel Cohen
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA
dcohen@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA
croft@cs.umass.edu

## ABSTRACT

Retrieving correct answers for non-factoid queries poses significant challenges for current answer retrieval methods. Methods either involve the laborious task of extracting numerous features or are ineffective for longer answers. We approach the task of non-factoid question answering using deep learning methods without the need of feature extraction. Neural networks are capable of learning complex relations based on relatively simple features which make them a prime candidate for relating non-factoid questions to their answers. In this paper, we show that end to end training with a Bidirectional Long Short Term Memory (BLSTM) network with a rank sensitive loss function results in significant performance improvements over previous approaches without the need for combining additional models.

## Keywords

deep learning; question answering; non-factoid

## 1. INTRODUCTION

Traditional information retrieval (IR) methods focus on the relevance of documents to queries. In query based IR, documents are deemed relevant if they address the topic implied by the query. Collections often have more than one relevant document, and term overlap can be an effective measure of potential relevance. For the task of factoid question answering (QA), the relevant document becomes a single sentence or entity that answers the specific information request of the question. As these factoid questions are specific, a small window of text surrounding an answer can be used in a retrieval method. An example of this is seen in a sample factoid question from the TREC QA task:

**Question:** What is crips' gang color?

**Answer:** Prosecutors said the "rampage of murder and mayhem" was carried out with bullets that had been painted blue, the crips' signature color.

These two tasks contrast with the more recent task of *non-factoid QA* [11], where there is typically a range of possible answers due to the open ended nature of the question, but correctness is determined by more than topical relevance. Non-factoid answers can span multiple sentences with the majority of the text having little term overlap with the question. A typical question demonstrating these issues from the dataset used in this paper is shown below:

**Question:** How do male penguins survive without eating for four months?

**Top Answer:** Male penguins don't eat for 60 days. The female comes back after 2 months, and the male goes to feed again. During the incubation period, the male's one and only job is to keep the egg warm. So he conserves energy by not moving at all. They just huddle. During this time he can lose up to 1/3 to 1/2 of his body weight.

Here, the answer is correct, but large amounts of the text in the answer have little direct overlap with the question beyond the first sentence. While present in all QA applications, this disconnect causes significant issues for retrieving non-factoid answers. Additionally, there is often only one correct answer provided in the standard testbeds making the task even more difficult.

Deep learning, specifically recurrent neural networks (RNN), are able to learn representations of text across positions in a sequence, bridging the lexical gap between the question and its corresponding answer by receiving updates from previous information in the sequence. LSTMs expand on this by storing an internal cell state even if that cell does not activate, allowing for semantic relations that span across sentences to be learned. This is encouraging as other methods rely on a complex empirical process of determining which features to extract, and modeling semantic and syntactic dependencies is often computationally intensive [10]. LSTM networks are capable of learning representations based on their loss function [2] without the need for feature extraction [8]. This is particularly useful in the realm of non-factoid QA as conventional features often fail to significantly boost performance [11].

## 2. RELATED WORK

Surdeanu et al. [11] previously investigated IR methods in the Webscope L4 dataset. However, their implementation

*ICTIR '16, September 12-16, 2016, Newark, DE, USA*
© 2016 ACM. ISBN 978-1-4503-4497-5/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2970398.2970438

involved reranking the top N results retrieved by a standard IR system. They used a large number of features for the reranking process, and the FG2 group of features focusing on translation dependencies offered the greatest increase in performance over all other features extracted.

Deep learning in the realm of IR is not new. Severyn and Moschitti [10] have shown the efficacy of using a deep convolutional neural network (CNN) to learn pooled representations of question and answer sentences for the factoid QA task. The network they implemented involved multiple convolutional layers which evaluated the question and answer as separate inputs. A subsequent matrix computes the similarity between the two representations and is concatenated with individually pooled representations of the sentences and externally computed term frequency statistics prior to a dense layer. While this network was only applied to factoid QA pairs where the answer was no longer than a sentence, it demonstrates that deep networks succeed in learning distributional relations between the queries and relevant answers.

Iyyer et al. [3] have demonstrated the application of RNNs to the task of quiz bowl questions by modeling textual compositionality over a standard RNN with a specific ranking loss function. However, their application involves long queries with single word answers with multiple questions having the same answer.

Wang et al. [16, 15] expand on this work by investigating the performance of a BLSTM on the TREC QA task and the non-factoid task. They implemented a BLSTM, but used pretrained embeddings independent of the network and a non rank specific loss function. Their implementation fails to outperform the previous CNN network [10] on the TREC QA task, but achieves mean average precision and precision at 1 metrics within 3% and 1% of the CNN. Additionally, the final model's output is boosted using a gradient boosted regression tree with each answer's BM25 score as the BLSTM is unable to capture that information.

All of the previous work mentioned has been trained on data passed through an embedding matrix [10, 3, 16]. The most commonly used is Mikolov's skip-gram word2vec [7], though there exists continuous bag-of-words (CBOW) and GloVe [9] implementations. These methods all capture distributional representations, and Arora et al. [1] have shown that these representations are noisy factorized pairwise mutual information (PMI) matrices. Levy et al. [5] have shown that different implementations all perform at a similar level.

# 3. A NEURAL NETWORK FOR NON FACTOID RETRIEVAL

In this paper, we use a variant of a LSTM network structure implemented in [16] and [2]. The standard RNN architecture described in [2] is used such that each layer not only receives input from the layer below it, but also its own output from the previous time step. LSTM units replace the standard neuron of a RNN with additional internal structures to manage vanishing and exploding gradients. These structures consist of input, forget, and output gates that manage the information flow of the cell's internal state.

We utilize a bidirectional neural network as in [2, 16], which can be viewed as inputting the sequence in reverse order to a second layer at the same level of the graph, and then merged either through concatenation or element-wise summation. The bidirectional layers for this paper were implemented via concatenation. A simplified representation of the network is shown in Figure 1 with the output of the network represented as $\hat{y}$.

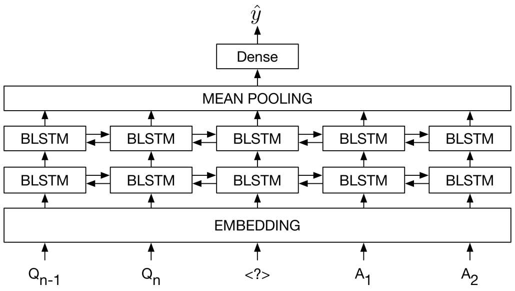


**Figure 1: A simplified representation of the BLSTM network with an $n$ length question**

Previous work [3, 10, 16] involved training word2vec embeddings externally of the neural network, which resulted in representations learned independently from the network. In the network implemented in this paper, we create an embedding layer below the BLSTM network, such that the performance during training would backpropagate to the word embedding layer which learns dense representations. In addition, we train with a rank sensitive loss function rather than treating the learning stage as a pure classification task.

# 4. EXPERIMENTS

## 4.1 Data

| **Tokens** | Webscope L4 | nfL6 |
|---|---|---|
| Min | 6 | 10 |
| Max | 350 | 79 |
| $\mu$ | 77.4 | 39.0 |
| $\sigma$ | 62.0 | 13.2 |

**Table 1: Statistical description of tokens per question-answer pair in nfL6 and Webscope L4 after preprocessing**

The datasets used for our experiments were Yahoo's Webscope L4 and a filtered, lower quality non-factoid set created from Yahoo's general Webscope L6, named nfL6. The L4 set has been used previously [11] for non-factoid QA and is sometimes referred to as the "manner" collection. It consists of 142,627 questions, of which we select 138,340 questions that satisfy the condition of being under 351 words when combined with their corresponding answer and do not contain websites. The word limit was used as LSTM networks are not capable of capturing dependencies on arbitrary length sequences and would not be able to learn representations of greater length answers. All questions are of the manner "how {to|do|did|does|can|would|could|should}..." and are high quality. Each question contains a noun and verb, and each answer is well formed. All answers that were not the highest voted answer were removed for each question as multiple answers for a question could be correct. This was done so the network would learn to better differentiate between correct and incorrect answers and not try to learn which answers would receive the highest votes.

The nfL6 dataset, after processing, consists of 87,361 questions. Unlike L4, the questions in this dataset are more generic and contains questions such as "Why is the sky blue?" and "Why do people steal?". Furthermore, answers are not as high quality. As there does not exist any equivalent dataset of generic non-factoid questions, this set was created using a linear kernel with a support vector machine. Initial training data is from UIUC's question dataset [6]. Fine grained classes of *description, manner*, and *reason* within the coarse grained class DESC were used as positive examples, with all others as negative examples. 3,500 additional training examples were attained from active training based on their distances to the hyperplane [14]. Additionally, to reduce noise, negative classifiers were trained on ENTITY, ABB, LOCATION, NUMERIC, and HUMAN classes to further reduce factoid questions in the collection.

Training, validation, and testing sets[1] for the BLSTM implementation were created in a similar fashion to [3]. A small pool of candidate answers were collected for each question based on top results in a BM25 search.

### 4.2 Network Configuration

For input to the network, each question is concatenated with its answer and a `<?>` character is inserted between the two strings as shown in Figure 1. Incorrect answers were concatenated the same way with the question string to create negative training examples. The `<?>` character was used similar to the `<EOS>` and `<S>` mark in [12] and [16, 15] respectively. This mark signifies the transition between source and target sentences and is depicted in Figure 1.

The specific network configuration consisted of a 256 dimension embedding matrix initialized uniformly, which feeds directly into two 512 length BLSTM layers with concatenated outputs. The cell activation function for the LSTM nodes is the sigmoid function, with internal gates using the *tanh* function. The output of the last BLSTM layer is mean pooled across time steps and fed into a single dense node with a sigmoid activation function. As mentioned previously, the embedding layer is part of the network during training, and thus will change word representations to best fit the loss function.

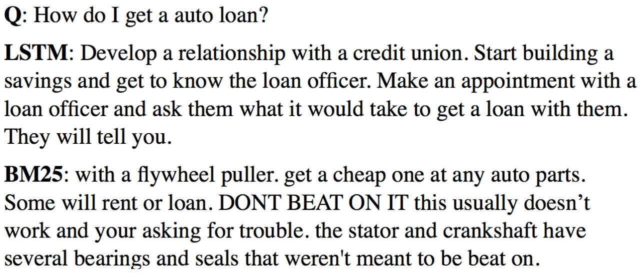
Optimization was done using the Adam algorithm [4] and trained to minimize the binary cross entropy weighted by how well the answers are separated with respect to the non-relevant training examples for each query as shown below.

$$L = \sum_{q \in Q} (1 - (q_r - \mu_{q_{nr}})) BCE_q$$

With $Q$ as all questions, $BCE_q$ as standard binary cross entropy for the question, $q_r$ as the relevant answer score and $\mu_{q_{nr}}$ as the mean of all non relevant candidate answer scores for q. As the task cares about relative ranking over binary classifications, this scales the loss relative to the distance in scores such that weights will change based on the questions with the hardest to differentiate answers.

### 4.3 Evaluation

The evaluation metrics used are mean reciprocal rank (MRR) and precision at 1 (P@1) which are both common in IR and QA evaluations. Precision at 1 is a binary metric that is 1 if the correct answer is ranked highest, and 0 otherwise. The mean is then taken to evaluate performance over a collection of questions. The reciprocal rank is the multiplicative inverse of the highest ranking correct answer retrieved for a question. Thus the mean is $\frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$ with $Q$ being the set of questions.



**Q**: How do I get a auto loan?

**LSTM**: Develop a relationship with a credit union. Start building a savings and get to know the loan officer. Make an appointment with a loan officer and ask them what it would take to get a loan with them. They will tell you.

**BM25**: with a flywheel puller. get a cheap one at any auto parts. Some will rent or loan. DONT BEAT ON IT this usually doesn't work and your asking for trouble. the stator and crankshaft have several bearings and seals that weren't meant to be beat on.

**Figure 2: Example of a question in which the BLSTM implementation successfully returns the correct answer while BM25 does not.**

The test collection was created from pooling the top 10 results from a BM25 search for each question, and including the correct answer as the 10th answer if it is not included in the list. These were then processed into sequences described in Section 4.2.

## 5. RESULTS AND DISCUSSION

The work done in [11] examined the L4 dataset using hand crafted features for reranking. However, they only selected the subset of queries which their initial IR system returned correct answers for the top N retrieved. This is not a realistic measure as it only takes into account 3.1% of all queries in the case of the BM25 benchmark for the top 10 retrieved. Additionally, performance gain is biased towards queries that behave favourably with the BM25 algorithm.

The end to end BLSTM network is instead compared against previous deep learning implementations and the BM25 baseline in Table 3. BM25 was chosen for the baseline as Yih et al. [13] have shown that *tf.idf* models are a competitive benchmark. While the CNN in [10] fails to capture any dependency between questions and answers in the L4 data, the BLSTM implementations are successful in learning a relation between them. Furthermore, end to end training significantly improves results over using an independently trained word embedding matrix without the need of an additional model to incorporate term frequency information and BLSTM layer as used in [16, 15]. This performance difference becomes more apparent when the language gap between training and testing of the embedding matrix grows. The nfL6 dataset contains slang and abbreviations not present in typical training text, which causes a hyperparameter tuned BLSTM implemented in [16, 15] to perform well below the modifications used in this paper.

The effect of the rank sensitive loss function results in significant improvement as well, referenced as BLSTM-Loss in Table 3. In training and evaluation, the network's range for $\hat{y}$ is dependent on the question rather than consistently centered around one point in [0,1]. As the task focuses on the relative rankings of candidate answers, weights are updated based on the difference of non-relevant and relevant answers instead of solely based on their respective entropy.

An interesting characteristic for these datasets is the poor

[1] Available at https://people.cs.umass.edu/dcohen

**Table 2: Results on Webscope L4 and nfL6**

| **Implementation** | **L4** | | **nfL6** | |
|---|---|---|---|---|
| | **P@1** | **MRR** | **P@1** | **MRR** |
| Okapi BM25 | 0.0783 | 0.1412 | 0.1312 | 0.2660 |
| Severyn and Moschitti | 0.0989 | 0.2434 | 0.1438 | 0.2842 |
| Wang and Nyberg | 0.4414 | 0.6152 | 0.1232 | 0.3271 |
| **BLSTM** | $0.4752^{*}$ | $0.6377^{*}$ | $0.2002^{*}$ | $0.4043^{*}$ |
| **BLSTM-Loss** | $0.5157^{*\dagger}$ | $0.6642^{*\dagger}$ | $0.2375^{*\dagger}$ | $0.4219^{*}$ |

**Table 3: Significant differences relative to Wang and Nyberg denoted by *, † denotes relative to BLSTM (using two tailed t-test with $p < 0.05$)**

performance of BM25 on both non-factoid collections. It cannot differentiate between relevant and non-relevant when the answers no longer echo terms used in the query. Figure 2 provides an example in the L4 dataset where the BLSTM correctly identifies the answer, and BM25 fails. BM25 retrieves an answer that has more query terms than the BLSTM retrieved answer; however, it does not answer the question. The BLSTM is able to learn a representation of “loan” to “credit” and “union” in addition to leveraging the query term “loan”. This reflects in the results of the Severyn and Moschitti [10] as the use of term overlap features appended to the output of a hidden layer does not improve results over the sequence based approach of the LSTM.

## 6. CONCLUSIONS AND FUTURE WORK

Implementing an end to end BLSTM with a rank sensitive loss function results in significant improvement over previous deep learning implementations without the need of term overlap information.

As the results indicate that the non-factoid RNN networks are sensitive to the training of the embedding layer, a possible character level embedding might negate the need to learn an embedding for each corpus and allow the network to update weights to represent the combination of character vectors as words.

In addition, using a convolutional layer as input to the BLSTM network can potentially result in better abstractions for the LSTM layers to process, as CNNs have been able to capture factoid level information comparable to recurrent networks for the TREC QA task [10].

## 7. ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #CNS-0934322, and in part by NSF grant #IIS-1419693. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## 8. REFERENCES

[1] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. *CoRR*, abs/1502.03520, 2015.

[2] A. Graves, N. Jaitly, and A.-R. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *ASRU, 2013*, pages 273–278, Dec 2013.

[3] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III. A neural network for factoid question answering over paragraphs. In *EMNLP*, 2014.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[5] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225, 2015.

[6] X. Li and D. Roth. Learning question classifiers. In *COLING - Volume 1*, COLING ’02, pages 1–7, Stroudsburg, PA, USA, 2002. ACL.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[8] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. K. Ward. Deep sentence embedding using the long short term memory network: Analysis and application to information retrieval. *CoRR*, abs/1502.06922, 2015.

[9] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[10] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, SIGIR ’15, pages 373–382, New York, NY, USA, 2015. ACM.

[11] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online qa collections. In *ACL:HLT*, pages 719–727, 2008.

[12] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[13] W. tau Yih, M.-W. Chang, C. Meek, and A. Pastusiak. Question answering using enhanced lexical semantic models. In *ACL*. ACL, August 2013.

[14] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, Mar. 2002.

[15] D. Wang and E. Nyberg. A recurrent neural network based answer ranking model for web question answering. In *WebQA Workshop, SIGIR ’15, Santiago, Chile.*

[16] D. Wang and E. Nyberg. A long short-term memory model for answer sentence selection in question answering. In *ACL-IJCNLP, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 707–712, 2015.