

# A Study of Question Effectiveness Using Reddit “Ask Me Anything” Threads

**Kristjan Arumae, Guo-Jun Qi, Fei Liu**

University of Central Florida, 4000 Central Florida Blvd., Orlando, Florida 32816  
k.arumae@knights.ucf.edu, guojun.qi@ucf.edu, feiliu@cs.ucf.edu

## Abstract

Asking effective questions is a powerful social skill. In this paper we seek to build computational models that learn to discriminate effective questions from ineffective ones. Armed with such a capability, future advanced systems can evaluate the quality of questions and provide suggestions for effective question wording. We create a large-scale, real-world dataset that contains over 400,000 questions collected from Reddit “Ask Me Anything” threads. Each thread resembles an online press conference where questions compete with each other for attention from the host. This dataset enables the development of a class of computational models for predicting whether a question will be answered. We develop a new convolutional neural network architecture with variable-length context and demonstrate the efficacy of the model by comparing it with state-of-the-art baselines and human judges.

## Introduction

Learning to ask effective questions is important in many scenarios. For example, doctors are trained to ask effective questions to gather necessary information from patients in a short time (Molla and Vicedo 2007); journalists frame their questions carefully to elicit answers (Vlachos and Riedel 2014); students post their questions on class discussion forums to seek help on assignments (Moon, Potdar, and Martin 2014). Naturally the more effective a question, the better the information-seeking or problem-solving purpose is served. Despite its importance, the problem of “what constitutes a good question” is largely unexplored. This paper seeks to close the gap by designing and evaluating algorithms that learn to discriminate effective questions—questions that elicit answers—from those that do not. We conduct a large-scale data-driven study using Reddit “Ask Me Anything” (henceforth “AMA”) discussion threads.

Reddit is a vibrant internet community with more than 36 million registered users (Reddit 2015). It was founded on June 23rd, 2005 and is currently ranked as the 24th most frequently visited website. The vast stream of data provides an unprecedented opportunity to study question effectiveness. Our focus of this work is on the “Ask Me Anything” subreddit. It is a popular subforum with more than 15 million subscribers. Each AMA thread emulates an online press

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

---

**AMA Holocaust survivor who immigrated to the US in 1951 with my husband and twin daughters. AMA.**

I am sitting with my 89-year-old grandmother who is always looking for a new audience. she has a spectacularly clear memory and important stories to tell. Here is her brief self-introduction: I was born in Tluste, Poland (which is now the Ukraine) in 1923...

- Q1 Do you think it takes a horrible experience to not take life for granted? (No Answer)
  - Q2 How has your experience changed your view of the world compared to the younger generations? (No Answer)
  - Q3 How does she feel about the situation in Greece now? (No Answer)
  - Q4 Do you sometimes have dreams, nightmares or flashbacks about things you saw/experienced then?
  - A4 Oh yes. In the beginning I dreamed of hiding my babies and then it was hiding my grandchildren...
- 

Figure 1: A snippet of the AMA thread. It includes the title (shown in bold), a brief intro from the AMA host, four questions (Q1–Q4) posted by Reddit users, among which only the last question was answered by the host (A4).

conference. An AMA host initiates a discussion thread with “Ask Me Anything” or “Ask Me Almost/Absolutely Anything.” The host will provide a brief background description and invite others to ask any questions about any topic. An example is illustrated in Figure 1. The questions in an AMA thread compete with each other for attention from the host. The host will chooses to answer some questions while leaving others behind.

In this paper we will investigate whether it is possible to automatically predict which questions will be answered by the host. We hypothesize that the question wording is important. For example, “*How does she feel about...*” (Figure 1) appears to be a difficult question that requires a delicate answer. Further, some topics are considered more favorable/unpleasant than others by AMA hosts. We draw on recent development of deep neural networks to predict question answerability. The advantage of neural models is the ability to automatically derive question representations that encode both syntactic structure and semantic knowledge. We develop a novel convolutional neural network (CNN) architecture that considers *variable-length context*. We draw an analogy between linear classifiers with n-gram features and

the new CNN model, and demonstrate improved results. The contribution of this work includes:

- A large-scale, real-world question-answering dataset collected from Reddit.<sup>1</sup> The dataset contains over 10 million posts and 400,000 questions generated by Reddit users. It is a very valuable resource for future research on question answering and question suggestion.
- We introduce a new variable-length context convolutional neural network model and demonstrate its efficacy against state-of-the-art baselines. We additionally present a human evaluation study to gain further insight into question answerability.

## Related Work

We discuss related work on community question answering and Reddit-inspired natural language processing studies.

A domain of research that is related to this study is community question answering (CQA), where users post questions on discussion forums and solicit answers from the community. Popular CQA websites include Yahoo! Answers, Ask Ubuntu, Stack Exchange, and Quora. The question-answering patterns in these websites are different from those of the Reddit AMA. For example, AMA questions are largely opinion-eliciting and they can be “any questions about any topic,” whereas CQA questions are problem-solving oriented and focus on technical topics. Multiple factors can contribute to unanswered questions on CQA websites, including the posting time, question quality, user reputation, and reward mechanisms (Anderson et al. 2012; Li et al. 2012; Liu et al. 2013; Ravi et al. 2014; Nakov et al. 2016). CQA corresponds to a single-inquirer multiple-responders setting, whereas AMA corresponds to a multiple-inquirers single-responder setting, because the AMA host single-handedly responds to all questions posted to the discussion thread. Whether a question will be answered depends on the question content—if it generates sufficient interest that warrants an answer from the host. Danish, Dahiya, and Talukdar (2016) analyze a range of factors that influence the AMA question answerability. In contrast to their work, we focus on developing neural networks that automatically learn question representations that incorporate syntactic and semantic knowledge.

Reddit has been explored for studying various language-related problems. Bendersky and Smith (2012) investigate characteristics of “quotable” phrases. The machine-detected phrases are quotable if they obtain endorsement from Reddit users. Wallace et al. (2014) leverage contextual information for irony detection on Reddit. Schrading et al. (2015) develop classifiers to predict texts that are indicative of domestic violence relationships. Jaech et al. (2015) introduce a comment ranking task that predicts the popularity of comments based on language features such as informativeness and relevance. Ouyang and McKeown (2015) create a corpus of personal narratives using Reddit data. Most recently, Tan et al. (2016) and Wei et al. (2016) respectively present studies on understanding the mechanisms behind “persuasion.”

<sup>1</sup>The Reddit question-answering dataset is available at: <http://www.nlp.cs.ucf.edu/downloads/>

They acquire discussion threads from the “ChangeMyView” subreddit. Users of this subforum state their views on certain topics, invite others to challenge their views, and finally indicate if the discussion has successfully altered their opinions. The studies find that both interaction patterns and language cues are predictive of persuasiveness. While Reddit has become a rising platform for language-related studies, the Reddit AMA corpus described next is expected to drive forward research in question-answering and question answerability.

## The Reddit AMA Corpus

We describe a methodology for creating a large-scale real-world dataset from Reddit “Ask Me Anything” subforum. Our goal is to collect *all* AMA threads and their associated posts during a span of eight years. The process is non-trivial, considering the sheer volume of data. In particular, we are faced with two challenges. First, the Reddit API returns at most 1000 threads per search query. To collect all threads, we implement a 30-minute query window to retrieve threads during each time period. Second, a thread is represented as a JSON object, but certain posts may be missing. The JSON object is restricted to contain up to 1,500 posts in order to avoid excessive page loading time, whereas the largest AMA threads can reach over 20,000 posts. To build a complete thread structure, we make additional API calls to retrieve missing posts and insert them back to the thread structure. An AMA thread contains a collection of posts organized into a tree structure. Each tree node is a post that includes a variety of useful information, such as author, body text, creation time, and replying posts.

The dataset contains a total of 67,723 discussion threads and over 10 million posts. Each thread contains 248 posts on average. The dataset spans eight years, from the beginning of Reddit AMA (May 2009) to end of the data collection period (April 2016). A number of celebrities have held AMA sessions during this time, including Barack Obama, Arnold Schwarzenegger, Astronaut Chris Hadfield, and Bill Nye. Other community members with unique experiences are also asked to host AMA sessions. The dataset is near-complete, missing only 250 threads (0.4% of all threads)—mostly due to server timeouts. More data statistics are presented in Table 1. The AMA dataset provides a highly valuable resource for future research on Reddit and question-answering.

**Preprocessing.** The goal of preprocessing is to create a collection of question posts and their associated labels. If the question post is replied by the AMA host, it is assigned a label of 1, otherwise 0. We perform a series of thread-level and post-level filtering operations on the dataset to achieve this goal. First, threads that contain fewer than 100 first-tier posts are removed. These threads may contain spam or be of low-quality. Next, the “AMA Request” threads are ignored. These threads request certain AMA sessions to be held but are not real AMA discussions. On the post-level, we restrict the question posts to be of first-tier and within the AMA host’s active period. The active period is defined as the beginning of the thread to the last post of the host. This setting ensures that the questions are directed towards the host, and the host is aware of the question posts. We further require question posts to contain a single sentence ending with a

	2009	2010	2011	2012	2013	2014	2015	2016	Total/Avg
# of threads	5,055	11,987	21,381	11,555	7,517	5,514	4,054	660	67,723
Avg # of posts per thread	87.55	97.04	94.12	214.08	308.35	395.46	338.98	447.02	247.83
Avg # of sentences per post	3.19	3.03	2.87	2.79	2.62	2.57	2.65	2.65	2.80
Avg # of words per post	49.82	46.44	43.49	41.98	38.07	37.02	39.02	40.37	42.03
# of question posts	697	2689	27,835	90,522	109,969	109,569	62,874	16,064	420,219
% of question posts w/ answer	44.76	30.83	30.03	20.48	19.66	19.77	22.02	18.56	25.76

Table 1: Statistics of the Reddit AMA corpus. The data ranges from the beginning of Reddit (May 2009) to the end of the data collection period (April 2016). Macro-average scores over all years are reported in the final column.

question mark. This allows us to focus on the question content and remove factors such as question length (a few words vs. several paragraphs), multiple questions in a single post, follow-up questions, and non-question content. The above filtering process generates 420,219 questions. Among them, about 26% are answered by AMA hosts (see Table 1).

## Our Approach

So far we have described a large-scale real-world dataset for predicting question effectiveness, we proceed by introducing our proposed solutions. We define *effective* questions as those that successfully elicit responses from AMA hosts in a time competitive environment<sup>2</sup>. The task naturally lends itself to a classification formulation where questions that receive answers from AMA hosts are labeled as 1, otherwise 0. We do not make use of the answer text in this study. The model learns to discriminate effective questions from ineffective ones based on question body. We introduce a new convolutional neural network architecture for this study. Deep neural models have demonstrated success on a range of natural language processing tasks, most notably machine translation (Sutskever, Vinyals, and Le 2014; Luong and Manning 2016), language generation (Rush, Chopra, and Weston 2015), and image captioning (Xu et al. 2015). This study presents a novel, context-aware convolutional neural network (CNN) architecture that considers both *context length* and *context variety*.

Concretely, let  $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$  be a single-sentence question post consisting of  $N$  word tokens, where  $N$  is the sequence length. If a question contains more than  $N$  words, it is truncated; otherwise, zeros are padded to the end of the word sequence. Each word is replaced by a word embedding ( $x_i \in \mathbb{R}^d$ ) before it is fed to the CNN model. With a slight abuse of notation, we use  $x_i$  to represent the word index in the vocabulary and  $\mathbf{x}_i$  (bold-face) to represent its embedding. We use the 300-dimension ( $d=300$ ) word2vec embeddings pre-trained on the Google News dataset with about 100-billion words.<sup>3</sup>

An important building block of the CNN model is the convolutional layer. It takes a word span as input, concatenates

<sup>2</sup>Throughout the paper we use question *effectiveness* and *answerability* interchangeably.

<sup>3</sup><https://code.google.com/archive/p/word2vec/>  
Words that do not have embeddings are set to all zeros. The out-of-vocabulary rate for this dataset is about 2%. We additionally trained embeddings using Reddit dataset but they did not yield improved performance over the Google word2vec embeddings.

the word embeddings (denoted as  $\mathbf{x}_{j-n+1:j}$  for position  $j$  and a span of  $n$  words, see Eq.(1)), applies a **filter** function  $h_j = f(\mathbf{w}^\top \mathbf{x}_{j-n+1:j} + b)$ , and produces a scalar-valued feature representation  $h_j$  for the word span (Eq.(2)). The filter is expected to activate when the word span encodes certain feature; it thus “memorizes” the feature using the parameters  $\mathbf{w}$  and  $b$ . The convolution process applies each filter to a sliding window of  $n$  words that runs over the input sequence. This process produces a **feature map** for each filter. A max-over-time pooling is applied to each feature map to select the most prominent feature  $\hat{h}$  in the input sequence (Eq.(3)).

$$\mathbf{x}_{j-n+1:j} = [\mathbf{x}_{j-n+1}; \mathbf{x}_{j-n+2}; \dots; \mathbf{x}_j] \quad (1)$$

$$h_j = f(\mathbf{w}^\top \mathbf{x}_{j-n+1:j} + b) \quad (2)$$

$$\hat{h} = \max_{j=1:N-n+1} h_j \quad (3)$$

In the aforementioned process, the word span can be of varying sizes ( $n=1,2,3,4,5,\dots$ ). Multiple filters can be applied to word span of each size. The combination of the two factors empowers the model to capture a flexible, enriched representation of the question body. But, how can we search for the optimal configuration of word span sizes (i.e., **context length**) and the number of filters applied to word span of each size (i.e., **context variety**), in order to best depict the prominent features encoded in the word span?

Existing studies use brute force to search for optimal configurations. Kim (2014) explores window sizes of  $\{3,4,5\}$  words, each with 100 filters. They are combined in a CNN architecture with one convolution/max-pooling layer, followed by a softmax output layer. Albeit simple, the CNN architecture outperforms several state-of-the-art neural models (Socher et al. 2013; Le and Mikolov 2014). Lei, Barzilay, and Jaakkola (2015) exploit non-consecutive word spans, using tensor decomposition to recognize patterns with intervening words. They explore spans of  $\{2,3\}$  words and  $\{50,100,200\}$  filters per word span. Zhang and Wallace (2015) experiment with various combinations of window sizes and numbers of filters per window size. Their results suggest that searching over the range of 100 to 600 for optimal number of filters is reasonable.

Different from previous studies, we draw an analogy between linear classifiers with n-gram features and convolutional neural networks, and use it to derive the optimal architecture configuration. We hypothesize that greater context leads to a higher degree of variety. Take n-grams for an example, trigrams in theory lead to  $O(|\mathcal{V}|^3)$  number of varieties, where  $|\mathcal{V}|$  is the vocabulary size. The intuition be-

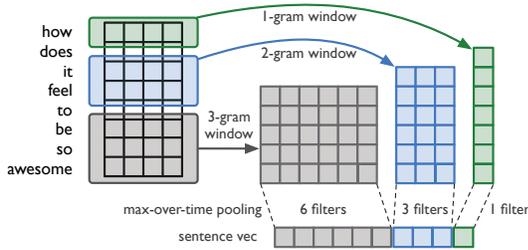


Figure 2: A toy example illustrating a convolutional neural network architecture that combines context window of  $\{1,2,3\}$ -grams and applies a varying number of filters to each window size.

hind it is that *the number of filters assigned to a word span correlates with the number of effective features* that can be encoded by a span of  $n$  words.

To this end, we explore n-gram statistics of large-scale, real-world datasets (Table 2), including the Europarl corpus (Koehn 2005) and Google N-grams dataset (Brants and Franz 2006). The Europarl corpus contains the European Parliament proceedings collected since 1996. In this dataset, the number of bigrams is about 15 times the number of unigrams (denoted by  $\uparrow_u 15x$ ), whereas the number of 3-grams, 4-grams, and 5-grams are of similar sizes, corresponding to averagely 2.4x of the bigram number (denoted by  $\uparrow_b 2.4x$ ). This is not Zipf’s law since we count the number of unique n-grams—regardless of their frequencies—in order to induce a relationship between the window sizes and numbers of effective features. Note that n-grams that appear only once are filtered out from the Europarl corpus since they are unlikely to be effective features. In a much larger Google N-grams dataset (Brants and Franz 2006) where n-grams appear less than 40 times are filtered out, similar ratios hold.

To summarize, because a grid search on the optimal combination of window sizes and number of filters per window size is prohibitively expensive, we rely on heuristics derived from n-gram statistics to combine five window sizes of  $\{1,2,3,4,5\}$  words and set the numbers of filters to be  $\{x, 20x, 60x, 60x, 60x\}$  respectively, where  $x$  is a parameter tunable according to the available computing power. The ratios are obtained by averaging across the two corpora. We advocate for a convolutional neural network architecture that combines multiple window sizes and applies a varying numbers of filters per window size. The approach brings together two considerations on both the context length and context variety, where greater context leads to higher variety. Finally, the convolution/max-pooling layer produces a vector representation for each question (illustrated in Figure 2). We apply a softmax layer on the top to predict whether the question will be answered by the AMA host.

## Experiments

Having described a context-aware CNN architecture in the previous section, we next compare it to a range of baselines and report performance on the Reddit AMA dataset.

Order	Europarl Corpus		Google N-grams	
	#Ngrams	Increase	#Ngrams	Increase
1-gram	53,253	—	13,588,391	—
2-gram	816,091	$\uparrow_u 15x$	314,843,401	$\uparrow_u 23x$
3-gram	2,070,512	$\uparrow_b 2.5x$	977,069,902	$\uparrow_b 3.1x$
4-gram	2,222,226	$\uparrow_b 2.7x$	1,313,818,354	$\uparrow_b 4.2x$
5-gram	1,557,598	$\uparrow_b 1.9x$	1,176,470,663	$\uparrow_b 3.7x$

Table 2: N-gram statistics of two large-scale, real-world datasets. Unique n-grams are counted; low-frequency ones are discarded.

## Baselines and Evaluation Metric

Our goal is to design and evaluate algorithms that learn to discriminate effective questions from ineffective ones. The task naturally lends itself to a text classification setting. We compare the context-aware CNN approach with two baselines. The first baseline is a logistic regression classifier with bag-of-words features. We use the LIBLINEAR implementation with  $l_2$  regularization (Fan et al. 2008). The model contains over 100k unigram features with stopwords removed. A bigram logistic regression model would be computationally prohibitive for this task due to the enormous feature space. We use the logistic regression baseline because of its proven track record on text classification. The same approach has been adopted in a prior work on classifying Reddit posts (Danish, Dahiya, and Talukdar 2016).

The second baseline is a state-of-the-art convolutional neural network architecture inspired by (Kim 2014). It has a flat structure that contains one convolution layer followed by a max-over-time pooling layer to build the sentence representation. We use a trigram context window and set the number of filters to be 100. This setting has been shown to perform robustly in previous work (Kim 2014; Lei, Barzilay, and Jaakkola 2015). Because the dataset is imbalanced—only about 25% of the questions are answered—we create a balanced dataset during training by setting the weights of answered/unanswered questions to be 4/1. This applies to all CNN trainings and has shown success on a number of classification tasks (Huang et al. 2016).

We create the train/valid/test splits by randomly sampling 100k/10k/100k questions from the 420,219 question collection. The train/valid/test data are temporally uniform, meaning there is an even representation of data instances from the entire timeline in each split. The metric used for model evaluation is the area under the ROC curve (AUC). Higher AUC scores are better. When the dataset is highly imbalanced, AUC scores have been shown to be a more appropriate metric than F-scores (Murphy 2012).

## System Evaluation Results

We report the performance of three systems in Table 3. Because the neural network approaches demonstrate performance variation due to the randomness in parameter initialization, 5 independent runs are performed for each test condition. Their mean, min, and max are reported.

Overall, we found that the context-aware CNN model to produce improved performance than logistic regression and

System	Mean	Min	Max
Logistic Regression	<b>0.512</b>	0.512	0.512
Baseline CNN	<b>0.516</b>	0.509	0.525
Context-aware CNN ( $x=5$ )	<b>0.523</b>	0.503	0.537

Table 3: Results on 100k test set as evaluated by AUC scores.

the basic CNN. Predicting whether or not a question will be answered by the AMA host can be a challenging task. This is illustrated by the AUC scores of all systems. There could be multiple reasons. First, the three systems employed in this study make extensive use of text features. While it allows us to push to the extreme of what can be achieved using text-only features, other useful non-text information could be incorporated in the future, such as question posting time. Second, each AMA host may have personalize preference on what types of questions they would prefer to answer. Some AMA hosts stay longer and answer all questions, while others, particularly celebrity hosts, may only answer questions sparsely for a few hours. Since this is a first and preliminary work on predicting question effectiveness, we plan to incorporate these factors in future studies.

Because the baseline CNN considers an unoptimized combination of context length (window size) and context variety (number of filters), we are curious to know how the two factors individually affect its efficacy in predicting question effectiveness. A set of experiments are conducted with varying window sizes of  $\{1, 2, 3, 4, 5\}$ -gram and number of filters of  $\{5, 100, 300\}$ . Results are reported on the 10k validation set. Each condition is again evaluated using 5 runs and the mean scores are reported in Table 4. When there are only 5 filters (first row), the best performance was achieved using a context window of 4 words. It appears that the neural model may attempt to memorize a few cue phrases that are particularly good for making predictions. When there are 300 filters, the model yields the best performance using a window of 1-word. It seems the network relies on a combination of unigrams to make the prediction. The context-aware CNN model in this study combines five sub-optimal baseline CNN settings, including (1-gram, 5), (2-gram, 100), (3-gram, 300), (4-gram, 300), and (5-gram, 300) and yields a better performance of 0.537 on the same dataset.

Filters	1-gram	2-gram	3-gram	4-gram	5-gram
5	0.512	0.510	0.510	<b>0.515</b>	0.511
100	0.518	0.516	0.510	0.519	<b>0.520</b>
300	<b>0.525</b>	0.517	0.515	0.517	0.507

Table 4: Results on 10k valid set. Baseline CNN performance with different window sizes and varying numbers of filters per window.

We further explore the effect of different training data sizes. Results are reported in Table 5. When the training data is of small size (1~2k), the context-aware CNN model shows signs of instability. With more data, the performance steadily improves.

Train Size	Mean	Min	Max
1k	<b>0.521</b>	0.507	0.530
2k	<b>0.511</b>	0.501	0.518
5k	<b>0.519</b>	0.507	0.529
10k	<b>0.523</b>	0.508	0.542
20k	<b>0.521</b>	0.512	0.528
50k	<b>0.525</b>	0.515	0.535
100k	<b>0.537</b>	0.531	0.544

Table 5: Results on 10k valid set using varied training data size.

## Human Evaluation Results

Finally, we examine how well humans can distinguish if a question is more effective than another. The hypotheses we investigate include (1) Do human annotators reach consensus on if certain questions are more likely to be answered than others? (2) Do questions preferred by human annotators agree with those answered by AMA hosts? (3) Will system prediction results be more close to human annotators or AMA hosts?

To answer these questions, we select 300 pairs of questions from our AMA dataset. In each pair, the question post A and B come from the same thread. They are paired with one having received an answer, while the other remains unanswered. They are temporally adjacent to each other so that the AMA host has a chance to see both questions in a small timeframe—the host receives a notification when new content is posted to the thread—but pick just one to respond to. We use the Amazon Mechanical Turk (mturk.com) for this study. In each HIT (Human Intelligence Task), a turker is presented with 3 pairs of questions collected from an AMA thread. One question of each pair is answered by the host. The question order is randomized so that the turkers will not find any specific pattern. The turkers are also provided with the thread title and background information from the AMA host. 5 turkers are recruited and they are rewarded \$0.05 for completing each HIT. We request the turkers to be in the U.S. and have a HIT approval rate of 90% or more.

We observe that the turkers demonstrate a reasonable degree of consensus among themselves: 63% of the selections are agreed by 4 turkers or more; 30% of the selections are agreed by all 5 turkers. On the other hand, there appears to be some discrepancy between the questions chosen to answer by turkers vs. AMA hosts. We adopt the majority-vote selection from the turkers. Overall, turkers have agreed with AMA hosts on 55% of the pairs regarding whether question A or B will be answered. The context-aware CNN has slightly higher agreement with turkers (54.33%) than AMA hosts (49.33%). Additionally, we hired an undergraduate student to perform the task. The student yields a higher level of agreement with turkers (68.6%) than AMA hosts (55%). These findings suggest that the answered questions are the preference of AMA hosts. Considering the subreddit has served as an informational as well as promotional platform for the hosts, the AMA hosts may have chosen questions in order to promote personal opinions and these questions may not be the ones that audience would like to have answered.

The human evaluation allows us to derive insights on why some questions are preferred over others, these findings will be incorporated in future work.

## Conclusion

This paper seeks to build computational models that learn to discriminate “answerable” questions from those that are not. We construct a large-scale question answering dataset that contains over 10 million posts and 400,000 single-sentence questions collected from the Reddit AMA subforum. As a preliminary study we present a novel context-aware convolutional neural network architecture that considers both context length and context variety for predicting question answerability. While other model architectures such as recurrent neural networks can be explored in the future, this study presents a first step toward using data-driven approaches for studying question effectiveness.

## References

- Anderson, A.; Huttenlocher, D.; Kleinberg, J.; and Leskovec, J. 2012. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of KDD*.
- Bendersky, M., and Smith, D. A. 2012. A dictionary of wisdom and wit: Learning to extract quotable phrases. In *Proceedings of Workshop on Comp Ling for Literature*.
- Brants, T., and Franz, A. 2006. Web 1T 5-gram version 1 ldc2006t13. *Philadelphia: Linguistic Data Consortium*.
- Danish; Dahiya, Y.; and Talukdar, P. 2016. Discovering response-eliciting factors in social question answering: A reddit inspired study. In *Proceedings of ICWSM 2016*.
- Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A library for large linear classification. *JMLR* 9:1871–1874.
- Huang, C.; Li, Y.; Loy, C. C.; and Tang, X. 2016. Learning deep representation for imbalanced classification. In *Proceedings of CVPR*.
- Jaech, A.; Zayats, V.; Fang, H.; Ostendorf, M.; and Hajishirzi, H. 2015. Talking to the crowd: What do people react to in online discussions? In *Proceedings of EMNLP*.
- Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.
- Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.
- Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML*.
- Lei, T.; Barzilay, R.; and Jaakkola, T. 2015. Molding CNNs for text: Non-linear, non-consecutive convolutions. In *Proceedings of EMNLP*.
- Li, B.; Jin, T.; Lyu, M. R.; King, I.; and Mak, B. 2012. Analyzing and predicting question quality in community question answering services. In *Proceedings of WWW*.
- Liu, J.; Wang, Q.; Lin, C.-Y.; and Hon, H.-W. 2013. Question difficulty estimation in community question answering services. In *Proceedings of EMNLP*.
- Luong, M.-T., and Manning, C. D. 2016. A hybrid word-character approach to open vocabulary neural machine translation. In *Proceedings of ACL*.
- Molla, D., and Vicedo, J. L. 2007. Special section on restricted-domain question answering. *Computational Linguistics* 33(1):41–62.
- Moon, S.; Potdar, S.; and Martin, L. 2014. Identifying student leaders from MOOC discussion forums through language influence. In *Proceedings of EMNLP*.
- Murphy, K. P. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- Nakov, P.; Marquez, L.; Moschitti, A.; Magdy, W.; Mubarak, H.; Freihat, A. A.; Glass, J.; and Randeree, B. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of SemEval*.
- Ouyang, J., and McKeown, K. 2015. Modeling reportable events as turning points in narrative. In *Proc. of EMNLP*.
- Ravi, S.; Pang, B.; Rastogi, V.; and Kumar, R. 2014. Great question! question quality in community q&a. In *Proceedings of ICWSM*.
- Reddit. 2015. Happy 10th birthday to us! Celebrating the best of 10 years of reddit. <https://redditblog.com/2015/06/23/happy-10th-birthday-to-us-celebrating-the-best-of-10-years-of-reddit/>.
- Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP*.
- Schrading, N.; Alm, C. O.; Ptucha, R.; and Homan, C. M. 2015. An analysis of domestic abuse discourse on Reddit. In *Proceedings of EMNLP*.
- Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*.
- Tan, C.; Niculae, V.; Danescu-Niculescu-Mizil, C.; and Lee, L. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of WWW*.
- Vlachos, A., and Riedel, S. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL Workshop on Language Technologies and Computational Social Science*.
- Wallace, B. C.; Choe, D. K.; Kertz, L.; and Charniak, E. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of ACL*.
- Wei, Z.; Liu, Y.; and Li, Y. 2016. Is this post persuasive? Ranking argumentative comments in the online forum. In *Proceedings of ACL*.
- Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*.
- Zhang, Y., and Wallace, B. C. 2015. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. <http://arxiv.org/abs/1510.03820v4>.