Predicting Question Quality Using Recurrent Neural Networks

Stefan Ruseti[1], Mihai Dascalu[1,2(✉)], Amy M. Johnson[3], Renu Balyan[3], Kristopher J. Kopp[3], Danielle S. McNamara[3], Scott A. Crossley[4], and Stefan Trausan-Matu[1,2]

[1] Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, 313 Splaiul Independenței, 60042 Bucharest, Romania
{stefan.ruseti,mihai.dascalu,stefan.trausan}@cs.pub.ro
[2] Academy of Romanian Scientists, Splaiul Independenţei 54, 050094 Bucharest, Romania
[3] Institute for the Science of Teaching and Learning, Arizona State University, PO Box 872111, Tempe, AZ 85287, USA
{amjohn43,renu.balyan,kristopher.kopp,dsmcnama}@asu.edu
[4] Department of Applied Linguistics/ESL, Georgia State University, Atlanta, GA 30303, USA
scrossley@gsu.edu

Abstract. This study assesses the extent to which machine learning techniques can be used to predict question quality. An algorithm based on textual complexity indices was previously developed to assess question quality to provide feedback on questions generated by students within iSTART (an intelligent tutoring system that teaches reading strategies). In this study, 4,575 questions were coded by human raters based on their corresponding depth, classifying questions into four categories: 1-very shallow to 4-very deep. Here we propose a novel approach to assessing question quality within this dataset based on Recurrent Neural Networks (RNNs) and word embeddings. The experiments evaluated multiple RNN architectures using GRU, BiGRU and LSTM cell types of different sizes, and different word embeddings (i.e., FastText and Glove). The most precise model achieved a classification accuracy of 81.22%, which surpasses the previous prediction results using lexical sophistication complexity indices (accuracy = 41.6%). These results are promising and have implications for the future development of automated assessment tools within computer-based learning environments.

Keywords: Question asking · Recurrent neural network · Word embeddings

1 Introduction

Many students experience difficulty understanding text content, principally due to individual differences related to prior domain knowledge, skills, and motivation [1]. Nonetheless, extended practice using active comprehension strategies, such as question asking, summarization, and elaboration, can support the development of reading comprehension [2–4]. However, a common difficulty facing educators is lack of classroom time to provide feedback to students about their success on using such reading strategies. To address this concern, intelligent tutoring systems can be used to

C. Penstein Rosé et al. (Eds.): AIED 2018, LNAI 10947, pp. 491–502, 2018.
https://doi.org/10.1007/978-3-319-93843-1_36

supplement classroom instruction, offering individual students automated feedback on the use of comprehension strategies. In these situations, ITSs commonly use Natural Language Processing (NLP) algorithms to automatically evaluate various linguistic and semantic features of students' typed input and subsequently provide summative scores and formative feedback. Students can use this feedback to improve their application of comprehension strategies, and in turn their ability to comprehend challenging text.

The goal of the current project is to develop algorithms that provide automated classifications to the instructional model, which in turn selects and delivers student feedback within a new question-asking practice module in the Interactive Strategy Training for Active Reading and Thinking (iSTART) system [5]. There is evidence demonstrating the efficacy of interventions that provide students with instruction on how to ask questions while reading [4], indicating that the development of an ITS which provides instruction and practice would benefit from methods to automatically assess the quality of the questions posed by the student with relation to the text being read. Although systems exist that automatically *generate* questions and *select* appropriate questions [6], we are unaware of any system that automatically assesses the quality of questions produced by readers. In the envisioned practice activity, students will read a text and, at pre-selected points in the text, construct questions about the content of the text. iSTART will automatically assess the quality of each question and provide the student scores and formative feedback to improve the students' questions. Thus, more accurate algorithms need to be developed. This study reports on one such attempt using deep learning techniques, specifically recurrent neural networks, to assess the quality of student-generated questions during reading.

2 Related Work

Classifying a question to assess its quality is useful in multiple scenarios such as classroom learning, reading exploration, and tutoring [7, 8]. We discuss the related work relevant to our approach using four threads. We first review question generation and its role in learning. We then describe existing question taxonomies along with existing work on question classification. Finally, we discuss recent work related to deep-learning models used for question classification and other related natural language processing (NLP) tasks.

2.1 Question Generation and Learning

Motivating students to generate questions is foundational in learning sciences [9]. Question generation supports comprehension, and promotes active learning, knowledge construction, problem solving, reasoning, and the development of sophisticated metacognitive skills [10]. Instruction on question asking generally focuses on generating deep-level reasoning questions, or questions that focus on reasoning about causal, logical, or goal-oriented systems [11]. Asking *good* questions is associated with improved comprehension, learning, and memory for targeted content [12]. In turn, comprehension and learning increase when students are trained to ask good questions

[4, 13], However, school children and adult learners tend to generate shallow level questions, and show substantial difficulties while generating questions that require deep level understanding [8].

2.2 Question Taxonomies

Questions can be categorized using a variety of taxonomies [14]. Several question taxonomies have been proposed by researchers developing models of question asking and answering in artificial intelligence, computational linguistics, discourse processing, education, and cognitive science [9, 11].

Several taxonomies originated from the Text REtrieval Conference-Question Answering (TREC-QA) track. For instance, the Ittycheriah et al. [15] taxonomy consists of 31 question types, whereas Hovy et al. [16] defined 141 question types. Harabagiu et al. [17] and Gerber [18] defined large question taxonomies spanning over 180 categories. Li and Roth [19] created a taxonomy with 6 coarse-grained and 50 fine-grained categories. Most of these taxonomies are hierarchical in nature and developed for question-answering and information retrieval systems. The focus for most of these taxonomies is on classifying entities such as person, location, organization, quantity, abbreviations and so on.

In this study, we have based our question classification categories on the Graesser–Person taxonomy [11] because their taxonomy is contextually related to learning. The Graesser–Person taxonomy [11] classifies questions into 16 categories covering different categories based on knowledge and cognitive proficiencies. These question categories are closely tied to Bloom's [20] taxonomy of cognitive activities. They further classify these categories into shallow ("who", "when", "what", "where") and deep ("why", "how", "what if") levels. Along similar lines, Mosenthal [21] developed a coding system to scale questions on five levels of abstractness, namely most concrete, highly concrete, intermediate, highly abstract and most abstract. Wisher and Graesser [8] defined categories based on knowledge such as agents and entities, class inclusion, spatial layout, compositional structures, procedures and plans, and causal chains and networks.

2.3 Question Classification

The methods used for automated question classification have varied greatly, from rule-based systems to advanced NLP techniques. Most approaches in the past relied on handcrafted rules that, though effective, are tedious to create and not scalable for matching questions. Olney et al. [22] manually built custom rules utilizing only surface features, ignoring the semantic and pragmatic context of the questions, for classifying the questions. The classifier performed quite well in distinguishing "questions" from "contributions" (F-measure = 0.99). However, classification of question type, within those identified as questions, achieved more modest accuracy (F-measure = 0.48).

More recently machine-learning techniques have been successfully applied to question classification. IBM's TREC-9 system [15] utilized maximum entropy models [23] for classifying questions labeled according to Message Understanding Conference (MUC) categories [24]. Another question classification system used the Sparse Network

of Winnows (SNoW) architecture [19]. Several question classification systems have made use of support vector machines (SVMs) [25, 26] and log-linear models [27] for classification of question types. Kopp et al. [28] used discriminant function analysis (DFA) using leave-one-out cross validation (LOOCV) for classifying question levels. A DFA on the entire set achieved an accuracy of 41.6%, while a LOOCV analysis obtained 40.1% accuracy when questions were classified for 4 levels (1 through 4). Notably, the accuracy improved when only 2 levels (shallow vs deep) were considered, up 61.1% in the LOOCV analysis.

Many more systems utilized NLP techniques including headwords and their hypernyms [14], conditional random field (CRF) [29], hierarchical directed acyclic graph (HDAG) Kernel, which is well suited to handle structured natural language data [30], parsing for classification [31], and automatic construction of grammars to match against question types [32]. Most of these systems have achieved high accuracy (nearing 90% or above) for general and coarse-grained question types.

2.4 Deep Learning Models for Question Classification

This study uses deep learning models, in particular recurrent neural networks (RNNs) [33], for classification of questions. Researchers have used deep learning for NLP tasks such as semantic parsing [34], sentence modeling [35], question answering [36], and other traditional NLP tasks [37]. Neural networks have also been used in question classification tasks. For instance, CNN has been used for semantic modeling of a sentence and later applied for six-way question classification using the TREC dataset [19, 35]. Fei et al. [38] investigated the effectiveness of neural network learning techniques for automatic classification of 233 objective questions into three difficulty levels (i.e., hard, medium, and easy), achieving an accuracy ranging between 76% and 78%. In their study, they opted to use a five-dimensional feature vector and a classic neural network architecture to characterize each question comprising of the following: (a) query-text relevance expressed as cosine similarity between the text, and question and answer pair (Q&A) vectors; (b) mean term frequency, (c) length of Q&A sequences, (d) term frequency distribution (variance) and (e) distribution of Q&A in original text.

While there has been a substantial amount of work for text classification using deep learning models [39, 40], there is little work available that has implemented deep learning models for classification of questions besides the previous studies that mostly rely on classic neural network architectures. To address this gap, this study evaluates the use of RNNs to predict the quality of questions.

3 Method

3.1 Corpus Description

We used the same data described in Kopp et al. [28]. For that study, participants read short texts and generated questions for each text read. These questions were coded by human raters using a four-level taxonomy that classified questions as shallow to deep. NLP indices were then calculated for each generated question and machine learning

techniques were used to predict the human ratings. This is the first study performed on this corpus that makes use of neural-network based methods.

To collect the questions corpus, Kopp et al. enlisted responses from 233 participants from Amazon Mechanical Turk (MTurk) [41]. With approval from the California Distance Learning Project (CDLP)[1], they collected 30 texts, which were life-relevant for adult readers. All texts were four to seven paragraphs long and had between 128 and 452 words. For each text, participants constructed questions for three to seven target sentences. Trained researchers identified these target sentences because they included information for which prior knowledge or prior information within the text could be used to elaborate on or make inferences using the sentence. Using an online survey software (www.qualtrics.com), each participant was shown three randomly-selected texts, segmented into chunks by the target sentences (i.e., a target sentence ended each chunk of text). At each target sentence, participants were instructed to type a question about the sentence content and given up to 6 min to submit. Participants' average response time was 85.2 s ($SD = 33.9$). Of the 4,629 collected responses, 4,575 were included in the final dataset. Those items that were excluded were statements, rather than questions.

From the original 16 question categories in the Graesser and Person [11] taxonomy, a modified question coding scheme was developed by Kopp et al. [28]. Table 1 contains the original 16 question categories as well as examples of the modified coding scheme. For the initial coding process, questions were classified into the original 16 categories. Within the modified scheme, two of the original question categories (instrumental/procedural and enablement) were collapsed into one category because researchers' initial training revealed they were not able to differentiate the categories reliably. The coding process also classified the questions according to the depth of the question (from 1-very shallow to 4-very deep). These differences in depth were annotated as (1) required answers are typically one word (e.g., yes or no), (2) required answers are still very short but may involve two or more elements (e.g., definitions), (3) required answers are longer, but do not relate to causal mechanisms (e.g., comparisons of two entities), and (4) required answers are lengthy and address systems' causal functions (e.g., which antecedent led to a consequence).

Table 1. The modified Graesser Person question taxonomy from Kopp et al. [28].

Level	Question categories from Graesser and Person Taxonomy [11]	Examples
Very shallow (1)	Verification, Disjunctive, Concept Completion, Quantification	Is 911 the emergency number everywhere?
Shallow (2)	Feature Specification, Definition, Example	What Constitutes and Emergency?
Deep (3)	Comparison Instrumental/procedural, enablement, Judgmental	Why are the majority from cell phones?
Very deep (4)	Interpretational, Antecedent, Consequence, Goal Orientation, Expectational	What happens if you call 911 in a non-emergency?

[1] www.cdlponline.org.

Before coding the complete dataset, two researchers went through a training process, twice coding approximately a randomly selected group of 20% of the corpus. After each training round, interrater reliability was established using Cohen's kappa, bivariate correlation (r), percent exact, and percent adjacent agreement for the depth of questions. Reliability between the researchers increased from the first training round (kappa = .74; $r = .78$; 76% exact agreement; 92% adjacent agreement) to the second (kappa = .80; $r = .86$; 79% exact agreement; 95% adjacent agreement). Once the interrater reliability for each training round was assessed, the researchers met to discuss discrepancies on that subset of the data. After the two training rounds, the entire dataset was coded. Questions were randomly selected and each researcher coded 60% of the question corpus. Interrater reliability on the final overlapping subset of the corpus was: kappa = .84, $r = .67$, exact agreement = 82%, adjacent agreement = 92%. Within this subset, discussions were conducted to resolve any discrepancies between the two researchers' assigned codes. Each question was assigned a final score, from 1 (very shallow) to 4 (very deep).

3.2 Network Architecture

Each entry from the corpus contains a text that provides previous contextual information, the target sentence and a question related to the given sentence. For predicting the question quality score, a neural network could use either the question and the sentence, the question and the text, or all three. We performed several experiments with different architectures, varying the input of the network and the model representation.

In all our experiments, the text was represented using a Recurrent Neural Network with variable input size for the word embeddings. We opted to use pre-trained Glove [42] and FastText [43] word and/or n-gram vectors. *Glove* is an unsupervised log-bilinear regression model that computes word embeddings using the global word co-occurrence matrix. In our experiments, we chose vectors of size 100 and 300 that were pre-trained on Wikipedia. *FastText* enriches the skip-gram model from Word2Vec [44] with vector representations for n-grams; for FastText, the embedding size is 300. This way, the embedding of a word can be influenced by the way it is written, which has been shown to correlate with the meaning of the word [45]. Moreover, unknown word embedding can be constructed from their corresponding n-gram embeddings.

The next step in building our architecture consisted of choosing a text representation model, usually called an encoder. We selected two of the most frequently used recurrent neural models, namely Long Short-Term Memory networks (LSTM) [46] and the Gated Recurrent Units (GRU) [47]. GRUs are considerably simpler and do not have the same representation power, but can be useful on smaller datasets because there are fewer parameters for the network to learn. Both models are usually used as bi-directional networks [48] in which two different networks are used for the forward and backward direction in the text; thus, all words are equally important in the final encoded representation.

Each encoder outputs a matrix of variable size because of the different number of words from the text. In order to be compared, the size has to be reduced to a fixed dimension. Common ways include concatenation of the last outputs from the forward

and backward networks, max-pooling over all outputs, or an attention mechanism that provides different weights to each word in the text, depending on the representation of the other text. Similar to the work of Santos et al. [49], we implemented a two-way attention pooling mechanism depicted in Fig. 1 that takes into account only the last sentence from the text and the question.

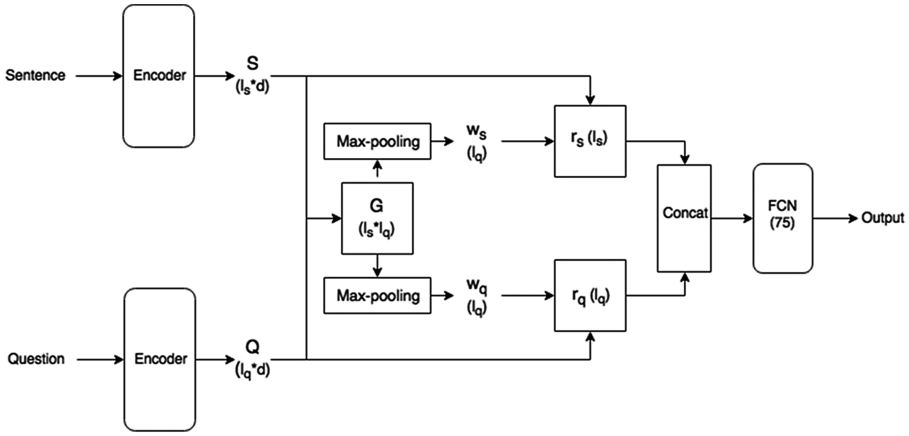

Fig. 1. Attentive pooling network architecture.

In Fig. 1, G is a matrix of size $l_s * l_q$, obtained from the outputs produced by the encoders (S and Q), as shown in Eq. 1, where U is a matrix of size $d * d$, l_s and l_q are the lengths of the sentence and question, and d is the output size of the encoder (in the case of the bidirectional encoders, the output is the double cell size). The matrix G can be perceived as a similarity function between each word in the question and each word from the sentence, which also takes into account the context in which they occur, since the similarity is computed on the encoders' outputs.

$$G = \tanh\left(S^t U Q\right) \tag{1}$$

Afterwards, the attention vectors (w_s and w_q) are constructed, which reflect the importance of each word in the sentence/question by applying max-pooling row-wise for the question, and column-wise for the sentence, followed by a softmax normalization. The question and sentence representations (r_q and r_s) are obtained by computing the dot product between the attention vectors and the encoders' outputs, as shown in Eq. 2.

$$r_q = w_q^t Q; \quad r_s = w_s^t S \tag{2}$$

The resulting representations are concatenated and passed through a dropout layer in order to minimize overfitting, followed by a hidden layer with the *tanh* activation function. Last, a softmax layer is applied to compute the probabilities for the four possible output classes. Training is performed using the Adam optimizer [50] and the cross-entropy loss function.

4 Results

The corpus was randomly split into training, validation, and test partitions each containing approximately 60%, 20% and 20% from the whole dataset. No texts were in two different partitions. Several experiments were performed with different network architectures and embedding models. Not all the results are included in this paper. Only the best configurations, with performance measured as accuracy on the validation set, were further fine-tuned with grid search, varying the cell size (50–300), hidden size (50–100), dropout probability (0–0.5) and training epochs (1–30). The reported results (see Table 2) were obtained on the test set, with the best hyper-parameters detected by the grid search. Experiments using linear regressions instead of a 4-way classification were also performed, but the results were less predictive and were not included in Table 2. LSTM results were also dropped because lower accuracy.

Table 2. Accuracy on 4-way classification on test set.

Model	Embeddings	Cell size	Hidden size	Accuracy (%)
BiGRU text, GRU question	Glove-100	75	50	75.22
GRU sentence, GRU question	Glove-100	175	50	77.11
BiGRU sentence, BiGRU question	Glove-100	100	100	78.11
BiGRU sentence, GRU question	Glove-300	100	75	77.11
BiGRU sentence, GRU question	Glove-100	200	75	80.11
BiGRU sentence, GRU question with *attention*	Glove-100	100	50	79.33
BiGRU sentence, GRU question	**FastText-300**	**100**	**75**	**81.22**

5 Discussion

This study introduced a new approach to classifying student-generated questions during reading, based on human ratings of depth. Using RNN architectures informed by GRU, BiGRU and LSTM cell types of different sizes, and different word embeddings (i.e., FastText and Glove), results revealed that the strongest model achieved a classification accuracy of 81.22%, which surpasses previous prediction results on the same dataset, that reported an accuracy of 41.6%. However, the results are lower than accuracies reported on other specific tasks of question classification using NLP techniques.

One observation that can be extracted from the results is that sentence encoding performed slightly better than encoding the text, probably since the text contains a lot of information not necessary to evaluate the question quality. In addition, the results may indicate that the network was not able to learn how to ignore potential noise from the text encoding. Arguably, adding more training examples would improve the results for the text encoder version.

Regarding the type of encoders used, the experiments showed that GRU models perform better than LSTM for this dataset likely because fewer parameters are required. The same reasoning can explain the results of the bidirectional networks. Although

BiGRUs have better expressing power (i.e., capability to encode the text based on word embeddings) than normal GRU models, they did not improve the results when used for questions. This indicates that the questions were short enough to be well represented by a simple GRU and the extra parameters were not useful.

Albeit successful in recent studies [51, 52], adding an attention layer did not improve the results, probably because of the additional matrix which needed to be learned by the neural network. In all other experiments, max-pooling was used to reduce the dimensionality of the encoders' output. Experiments using the concatenation between the last outputs of the forward and backward direction were also performed, but were not as accurate on the validation set, compared to the other solutions. As a result, they were not included in Table 2.

The best result was obtained using FastText word embeddings, which performed better than the equivalent network architecture relying on Glove word vectors of the same size (300). Smaller Glove vectors (100) achieved a higher accuracy compared to the larger ones (300) most likely because the network required fewer training parameters.

6 Conclusions

Many students, especially those with low prior knowledge, have difficulty making inferences necessary for text comprehension. iSTART is an ITS that provides reading strategy instruction to promote self-explanation and inference generation, in order to promote comprehension. Question asking can also improve readers' comprehension, especially when the questions generated are deep.

This study contributes to the development of question asking practice within iSTART. As such, practice modules for question asking require the application of NLP techniques to automatically assess the quality of students' questions. Thus, an overarching objective of this project is to create algorithms that are sensitive to the types of questions students ask during reading. Identification of particular question types would then trigger appropriate formative feedback with the hopes that the feedback would promote generation of deeper questions and improve comprehension. For example, when the algorithm detects shallow question asking, the feedback mechanism in iSTART would trigger a message prompting the reader to ask deep questions that would, for example, require learners to identify a causal mechanism.

This study describes a series of experiments that used recurrent neural networks to assess the quality of questions. The obtained results are very promising, with the best model exceeding 81% accuracy classifying questions into 4 classes of question depth. The results reported here indicate that successful classification models for question depth can be integrated into iSTART. These models could provide students with accurate formative feedback that may lead to increased reading strategy skills and overall reading comprehension.

Our next steps consist of implementing and testing these models in iSTART. Moreover, experiments should be conducted to further development of question depth classification. For instance, an experiment testing all three encoders at once (text, sentence,

question) could assess how a network making more informant decision, at the cost of increasing the needed parameters, could improve accuracy. Another experiment could test the use of only one encoder for text and extract the sentence representation from the hidden states corresponding to the words in that sentence. In addition, simpler solutions could also be tested on this corpus and compared with the deep learning methods. One example that proved successful on smaller datasets uses neural networks on top of string kernels [53] for answer selection. Although the tasks are not identical, simple similarity measures could be effective for evaluating questions, either stand-alone or as additional features in a more complex network.

Acknowledgments. This research was partially supported by the 644187 EC H2020 *Realising an Applied Gaming Eco-system* (RAGE) project, the FP7 2008-212578 LTfLL project, the Department of Education, Institute of Education Sciences - Grant R305A130124, as well as the Department of Defense, Office of Naval Research - Grants N00014140343 and N000141712300.

References

1. Snow, C.: Reading for Understanding Toward an R&D Program in Reading Comprehension. Rand Corporation, Santa Monica (2002)
2. Palincsar, A.S., Brown, A.L.: Interactive promote learning teaching independent from text to. Read. Teach. **39**, 771–777 (1986)
3. Rosenshine, B., Meister, C.: Reciprocal teaching: a review of the research. Rev. Educ. Res. **64**, 479–530 (1994)
4. Rosenshine, B., Meister, C., Chapman, S.: Teaching students to generate questions: a review of the intervention studies. Rev. Educ. Res. **66**, 181–221 (1996)
5. McNamara, D.S., O'Reilly, T., Rowe, M., Boonthum, C., Levinstein, I.: iSTART: a web-based tutor that teaches self-explanation and metacognitive reading strategies. In: Reading Comprehension Strategies: Theories, Interventions, and Technologies, pp. 397–420 (2007)
6. VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., Rosé, C.P.: When are tutorial dialogues more effective than reading? Cogn. Sci. **31**, 3–62 (2007)
7. Graesser, A.C., McMahen, C.L.: Anomalous information triggers questions when adults solve quantitative problems and comprehend stories. J. Educ. Psychol. **85**, 136–151 (1993)
8. Wisher, R.A., Graesser, A.C.: Question-asking in advanced distributed learning environments. In: Toward a Science of Distributed Learning and Training, pp. 209–234. American Psychological Association, Washington, D.C. (2007)
9. Beck, I., McKeown, M.G., Hamilton, R.L., Kucan, L.: Questioning the Author: An Approach for Enhancing Student Engagement (1997). https://eric.ed.gov/?id=ED408562
10. Kintsch, W.: Comprehension: A Paradigm for Cognition. Cambridge University Press, New York (1998)
11. Graesser, A.C., Person, N.K.: Question asking during tutoring. Am. Educ. Res. J. **31**, 104–137 (1994)
12. Davey, B., McBride, S.: Effects of question-generation training on reading comprehension. J. Educ. Psychol. **78**, 256–262 (1986)
13. Craig, S.D., Gholson, B., Ventura, M., Graesser, A.C.: Overhearing dialogues and monologues in virtual tutoring sessions: effects on questioning and vicarious learning. Int. J. Artif. Intell. Educ. **11**, 242–253 (2000)

14. Silva, J., Coheur, L., Mendes, A.C., Wichert, A.: From symbolic to sub-symbolic information in question classification. Artif. Intell. Rev. **35**, 137–154 (2011)
15. Ittycheriah, A., Franz, M., Zhu, W., Ratnaparkhi, A., Mammone, R.J.: IBM's statistical question answering system. In: Proceedings of TREC-9 Conference, pp. 229–234 (2000)
16. Hovy, E., Gerber, L., Hermjakob, U., Lin, C.-Y., Ravichandran, D.: Toward semantics-based answer pinpointing. In: Proceedings of the First International Conference on Human Language Technology Research, pp. 1–7 (2001)
17. Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., Morarescu, P.: FALCON: boosting knowledge for answer engines. In: Proceedings of Ninth Text Retrieval Conference (TREC 2000), pp. 479–488 (2000)
18. Gerber, L.: A QA Typology for Webclopedia (2001)
19. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, pp. 1–7 (2002)
20. Bloom, B.S.: Taxonomy of Educational Objectives, Cognitive Domain, pp. 20–24. McKay, New York (1956)
21. Mosenthal, P.B.: Understanding the strategies of document literacy and their conditions of use. J. Educ. Psychol. **88**, 314–332 (1996)
22. Olney, A., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., Graesser, A.: Utterance classification in AutoTutor. In: Proceedings of HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing, vol. 2, pp. 1–8. ACL, Morristown (2003)
23. Pietra, S.D., Pietra, V.D., Lafferty, J.: Inducing features of random fields. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 380–393 (1997)
24. Chinchor, N., Robinson, P.: MUC-7 named entity task definition. In: Proceedings of the 7th Conference on Message Understanding, MUC6, p. 21 (1997)
25. Hacioglu, K., Ward, W.: Question classification with support vector machines and error correcting codes. In: Proceedings of HLT-NAACL 2003, pp. 28–30. Association for Computational Linguistics, Morristown (2003)
26. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2003, p. 26. ACM Press, New York (2003)
27. Blunsom, P., Kocik, K., Curran, J.R.: Question classification with log-linear models. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006, p. 615 (2006)
28. Kopp, K.J., Johnson, A.M., Crossley, S.A., McNamara, D.S.: Assessing question quality using NLP. In: André, E., Baker, R., Hu, X., Rodrigo, Ma.M.T., du Boulay, B. (eds.) AIED 2017. LNCS, vol. 10331, pp. 523–527. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_55
29. Krishnan, V., Das, S., Chakrabarti, S.: Enhanced answer type inference from questions using sequential models. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT 2005, pp. 315–322 (2005)
30. Suzuki, J., Taira, H., Sasaki, Y., Maeda, E.: Question classification using HDAG kernel. In: Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering, vol. 12, pp. 61–68 (2003)
31. Hermjakob, U.: Parsing and question classification for question answering. In: Proceedings of Working Open-domain Question Answering, vol. 12, pp. 1–6 (2001)
32. Mishra, M., Mishra, V.K., Sharma, H.R.: Question classification using semantic, syntactic and lexical features. Int. J. Web Semant. Technol. **4**, 39–47 (2013)

33. Elman, J.L.: Finding structure in time. Cogn. Sci. **14**, 179–211 (1990)
34. Yih, W., He, X., Meek, C.: Semantic parsing for single-relation question answering. In: Association for Computational Linguistics, pp. 643–648 (2014)
35. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics, pp. 655–665 (2014)
36. Iyyer, M., Boyd-graber, J., Claudino, L., Socher, R., Daum, H.: A neural network for factoid question answering over paragraphs. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)
37. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (Almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
38. Fei, T., Heng, W.J., Toh, K.C., Qi, T.: Question classification for e-learning by artificial neural network. In: ICICS-PCM 2003, pp. 1757–1761. IEEE (2003)
39. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of 2014 Conference on Empirical Methods Natural Language Processing (EMNLP 2014), pp. 1746–1751 (2014)
40. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2267–2273 (2015)
41. Crump, M.J.C., McDonnell, J.V., Gureckis, T.M.: Evaluating Amazon's mechanical turk as a tool for experimental behavioral research. PLoS One **8**, e57410 (2013)
42. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
43. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information (2016)
44. Mikolov, T., Corrado, G., Chen, K., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of ICLR 2013, pp. 1–12 (2013)
45. Darío Gutiérrez, E., Levy, R., Bergen, B.K.: Finding non-arbitrary form-meaning systematicity using string-metric learning for Kernel regression. In: ACL, pp. 2379–2388 (2016)
46. Hochreiter, S., Urgen Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
47. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP 2014, pp. 1724–1734 (2014)
48. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**, 602–610 (2005)
49. dos Santos, C., Tan, M., Xiang, B., Zhou, B.: Attentive Pooling Networks. CoRR, abs/1602.03609. 2, 4 (2016)
50. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations (2015)
51. Xiong, C., Zhong, V., Socher, R.: Dynamic coattention networks for question answering. In: ICLR Submission, pp. 1–13 (2017)
52. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: ICLR 2017 (2017)
53. Masala, M., Ruseti, S., Rebedea, T.: Sentence selection with neural networks using string kernels. Proc. Comput. Sci. **112**, 1774–1782 (2017)