Distilling Essence of a Question: A Hierarchical Architecture for Question Quality in Community Question Answering Sites

Mun Kit Ho
School of EEE
Nanyang Technological University
Singapore
munkit001@e.ntu.edu.sg

Sivanagaraja Tatinati
School of EEE
Nanyang Technological University
Singapore
tatinati@ntu.edu.sg

Andy W. H. Khong
School of EEE
Nanyang Technological University
Singapore
andykhong@ntu.edu.sg

***Abstract*—Community question answering (CQA) sites have grown to be useful platforms where users search for highly specific information to resolve a problem. However, the significant increase in the number of user-generated content with high variance in quality on these sites not only presents challenges for user navigation but also outgrow the community's peer reviewing capacity. This necessitates ways to automatically assess the quality of new questions so as to maintain quality of content served to its users. While existing methods commonly employ social network indicators as features, our model aims to avoid social influence biases arising from these indicators by predicting the quality from semantic evaluation of the question text. Formulation of the proposed model is non-trivial as it requires the extraction of meaningful features from the noisy question text at different granularities while filtering redundant information. In this work, a neural architecture is proposed to address this problem by aggregating the textual features extracted at word- and sentence-level in a hierarchical manner. In addition, a unique attention mechanism that focuses on sentence segments for interpreting a question is developed. This new mechanism employs the global topical information from common problem contexts. The proposed approach is verified on the Stack Overflow question dataset and is shown to outperform existing neural models.**

***Index Terms*—community question-answering, question quality.**

I. Introduction

Community question answering (CQA) sites such as Quora, Yahoo Answers and Stack Overflow have become popular platforms for knowledge sharing and building in the age of Web 2.0. As the volume of these user-generated content grows, maintaining content quality becomes paramount. This is so that users can easily navigate and search for good quality information that matches their needs. Specifically, exceptional questions that pose useful problem statements and are well-written should be ranked higher and recommended to searchers, whereas badly-authored questions should be routed back for amendments or, to a large extent, even deleted. As a solution for this, automated evaluation methods for question quality (QQ) have been proposed [1], [2]. These automated tools are useful for the websites to evaluate the quality of newly-posted content at scale so that early intervention on poor-quality questions can be performed. This can also serve as a convenient tool for the asker to assess the written question before posting, thus preserving the website's overall content quality.

This paper focuses on the problem of QQ prediction when a question is first posted. Existing QQ models are often constructed using features derived from text combined with platform social indicators such as reputation, past question counts, or votes. While these features can be highly predictive, it has been reported that such aggregate indicators can cause user voting behavior to conform towards community norm [3], thus carrying forward patterns set by early votes and influential community members. To avoid such social influence bias in the QQ model, we propose a neural architecture that only relies on textual features of the questions. Existing text-based prediction models commonly employ hand-crafted text-based features [4] or highly-specialized topic-modelling based features [5]. These methods, however, require extensive labor and domain expertise to extract features and tune the classifiers. On the other hand, conventional sequential neural models such as convolutional neural network (CNN) and long short-term memory (LSTM) [6] have recently been successful in many text classification tasks. Although these models generally achieve high performance on benchmarking datasets, their performance is limited when applied to question modeling. This is due to the equal treatment of all segments within the question during feature extraction and neglecting the naturally disjoint semantics between sentences. Emphasis placed on specific parts of a question can be helpful in filtering out noisy texts inherent in user-generated texts (e.g., abbreviations and spelling errors) and prioritize the use of some highly-discriminative sentences. For example, expressions identical to the highlighted sentence in Fig. 1 (right) are usually indicative of a bad question where the asker might have failed to perform background research.

Inspired by the recent success of attention-layer based neural architectures for text classification applications, we address the above-mentioned challenges by applying data-driven attention modules for sentences in a given question and the

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

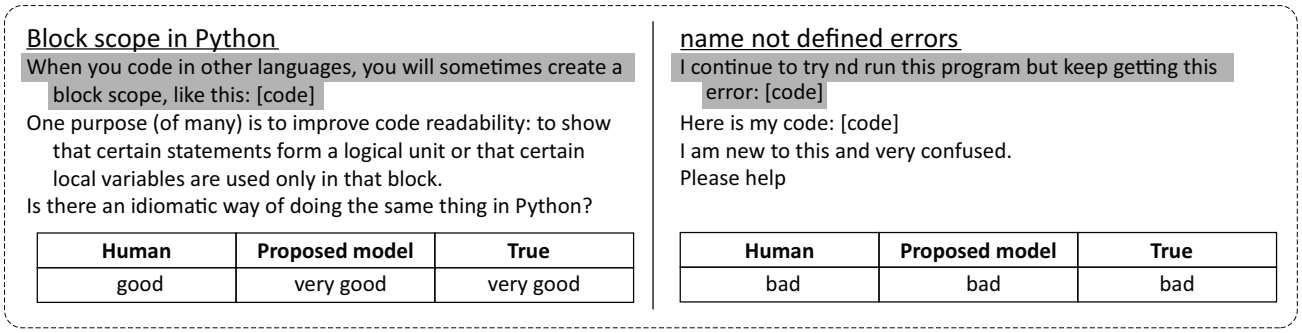

Fig. 1. Sentences identified as highly-discriminative by the proposed model and how its predicted label compares against human and true labels. (left) Example of a *very good* question, and (right) Example of a *bad* question.

constituent words. A hierarchical arrangement of this structure, similar to that described in [7], is proposed to better interpret the disjointed sentences progressively, and select them based on relative importance for QQ prediction. As opposed to [7], we regulate the attention weights for identifying the essential words in a given question via another hierarchical arrangement of attentions. This is achieved by using words in the most relevant topic identified via topic modeling. The performance of both hierarchical approaches (with and without topics) are compared against other sequential baselines on the Stack Overflow dataset with labels created according to community feedback indicators.

II. Method

Consider each question Q being composed of a set of l_s sentences $Q = \{s_1, s_2, ..., s_{l_s}\}$. Each sentence s_i consists of a set of l_i words $\{w_{i,1}, w_{i,2}, \ldots, w_{i,l_i}\}$, where i denotes the sentence index. Typical human interpretation of QQ involves identifying essential words from sentences and subsequently ordering the sentences in terms of contextual importance. The proposed two-stage hierarchical attention architecture, as shown in Fig. 2, takes this into account by learning the weighting schemes at these two levels using parameters u_w, u_s. Here, the subscripts w and s denote for the words and sentences, respectively.

A. The proposed two-stage hierarchical attention network (HAN) with topic-weighted attention (TwAtt)

The question words are first mapped into vector representations using a pre-trained embedding layer. A sentence encoder in the form of a bi-directional gated recurrent unit (GRU) is then employed to incorporate contextual information from surrounding words by learning hidden representations of the sequences. The final hidden representation of each word $h_{i,t}^{(w)}$ is obtained by concatenating the forward and backward hidden states as

$$h_{i,t}^{(w)} = [\overrightarrow{GRU}(w_{i,t}, \vec{h}_{i,t-1}^{(w)});\ \overleftarrow{GRU}(w_{i,t}, \overleftarrow{h}_{i,t+1}^{(w)})].$$

To eliminate noisy text elements that do not contribute significantly towards the sentence semantics, attention mechanisms are applied to the sentence encoder.

Attention mechanisms have gained popularity for enabling neural networks to focus only on the important features. While a few common variants have been introduced [8], [9], the neural architecture typically share identical structural components of key-value pairs and queries. An attention mask is first computed by matching a query against all keys to find compatibility scores. The mask scores then determine the corresponding values at the output. This reduces subsequent computations to only the most relevant feature, therefore improving model learning. We propose two variants of sentence-level attention for QQ. The conventional attention mechanism (vanilla) that is identical to [7] is first described. The second proposed topic-weighted attention (TwAtt) technique is subsequently introduced to regularize the attention mechanism and hence achieve better representation of each sentence.

For the vanilla attention mechanism [7], a vectorized parameter u_w serves as the query that interacts with the transformed hidden vectors to generate an attention coefficient. This parameter is analogous to a learned 'locus of attention' that guides the attention on certain words during interpretation of a sentence. Defining matrix W_w as the transformation weights, the attention coefficient is therefore computed using

$$a_{i,t}^{(w)} = u_w^{\top} tanh(W_w h_{i,t}^{(w)}). \tag{1}$$

The softmax-normalized coefficients

$$\lambda_{i,t} = \frac{exp(a_{i,t}^{(w)})}{\sum_t exp(a_{i,t}^{(w)})} \tag{2}$$

are then used as weights that determine the importance of a word in forming the overall sentence representation. Finally, the vector representation for sentence i is obtained via a weighted-average of the word hidden representations, i.e.,

$$s_i = \sum_t \lambda_{i,t} h_{i,t}^{(w)}. \tag{3}$$

Humans prioritize certain textual clues according its context while comprehending a problem. Likewise, a single attention scheme learned by a single u_w vector may underfit the highly diverse range of question topics. Therefore, we propose an augmentation of the vanilla version with a context-dependent attention that computes the attention coefficient

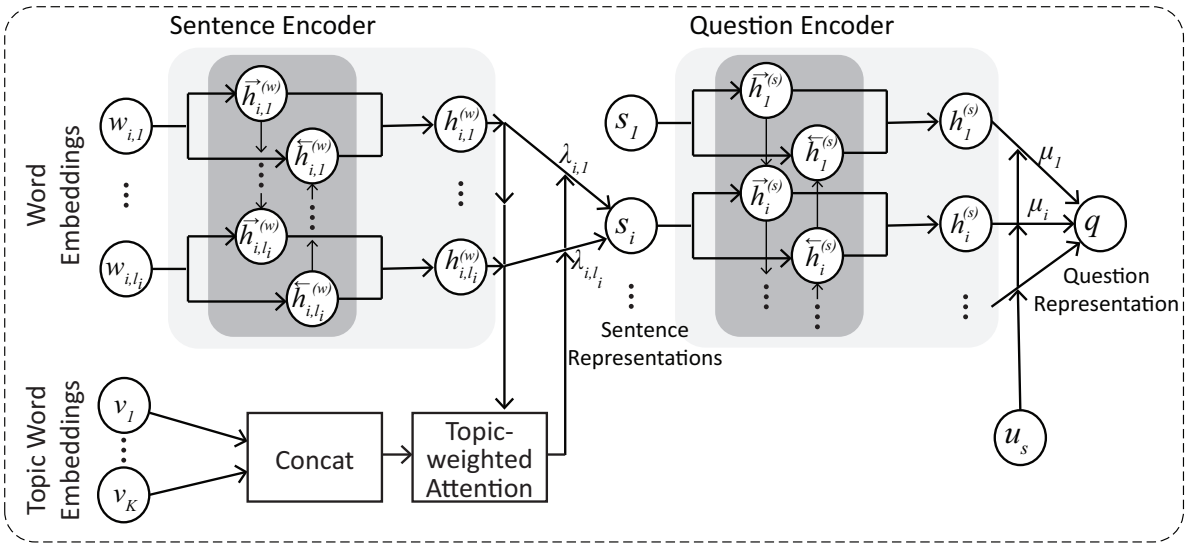

Fig. 2. Architecture of proposed tHAN network.

based on topical words, which, as a consequence, allows the algorithm to narrow features learned to within a local topic space. To achieve this topic-weighted attention (TwAtt), topic words are obtained from two sources, either tags assigned by the questioner, or words generated from topic models. The topic model is trained using latent Dirichlet allocation (LDA) [10] on all questions at document level. As shown in Fig. 2, word embeddings of the K most representative words $\{v_1, v_2, ..., v_K\}$ for the most relevant topic are then used to enhance information passed to the attention layer.

The key operation of this attention mechanism is illustrated in Fig 3. Inspired from other attention module designs [11], the single parameter vector in (1) is replaced by a variable query vector guided by topic words v_k. These topic words are first transformed into query vectors via a matrix W_v and non-linear function *tanh*. Similarly, the word hidden representations undergo an identical process to form the key vectors. A dot-product attention then computes an interaction score between the transformed topic word representations and the hidden representation of the question words, i.e.,

$$score_i(t,k) = tanh(W_v v_k)^\top tanh(W_w h_{i,t}^{(w)}).$$

Salient latent features from the transformed topic word representations are obtained via max-pooling, which is then used to derive the attention coefficient

$$a_{i,t}^{(w)} = \max_k score_i(t,k).$$

This is subsequently followed by a weighted-average to generate the sentence representation s_i described in (2) and (3).

B. Sentence importance selection

Different discourse function of each sentence indicates that not every sentence is equally important in determining the quality of the question. Hence we propose to identify highly-discriminative sentences at this layer. Similar to word

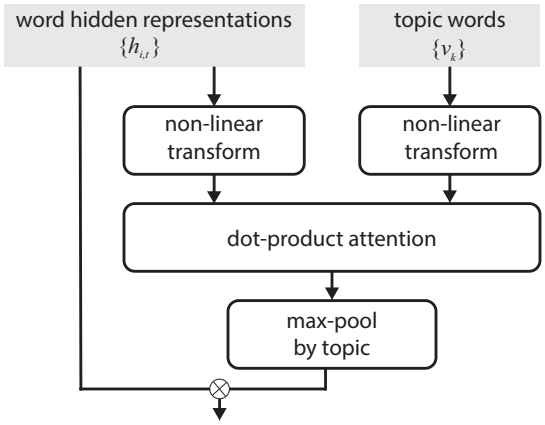

Fig. 3. Topic-weighted attention (TwAtt) mechanism.

representation described in Section II-A, each sentence representation s_i first undergoes an encoding process to obtain its hidden representation

$$h_i^{(s)} = [\overrightarrow{GRU}(s_i, \vec{h}_{i-1}^{(s)}); \overleftarrow{GRU}(s_i, \overleftarrow{h}_{i+1}^{(s)})].$$

This module consists of a vectorized parameter u_s that interacts with hidden vectors to generate a sentence attention coefficient $a_i^{(s)}$, which is subsequently normalized by the softmax function, i.e.,

$$a_i^{(s)} = u_s^\top tanh(W_s h_i^{(s)}); \quad \mu_i = \frac{exp(a_i^{(s)})}{\sum_i exp(a_i^{(s)})}.$$

Finally, the vector representation for the question q is obtained via a weighted average of the sentence hidden representations

$$q = \sum_i \mu_i h_i^{(s)}.$$

To determine the objective function, q then undergoes a linear transformation followed by softmax to obtain probabilities of each predicted label $\widehat{y}$ given by

$$p_{\widehat{y}} = softmax(W_q q + b),$$

where W_q, and b denote transformation weights and bias, respectively. Defining y as the true label, the model is subsequently trained by minimizing the cross-entropy loss

$$L = -\sum_{n}^{N} y_n \log p_{\widehat{y}_n}$$

across a mini-batch of N samples. This enables it to find an optimal set of attention parameters λ^*, μ^* to generate the question representation

$$q = f(\lambda^*, \mu^* | Q).$$

Hereafter, following the naming convention in [7], the proposed hierarchical approach with vanilla attention mechanism is named as Hierarchical Attention Network (HAN), whereas the proposed HAN with TwAtt is referred as tHAN.

III. Dataset

Experiments are conducted on a subset of questions available in the Stack Overflow data dump[1]. While a subset of questions between 2011-2012 tagged with "Python" have been chosen, the proposed algorithm can be extended to other question tags. Leveraging on the collective wisdom of the Stack Overflow community, quality labels are computed mainly using *votes* described in this paragraph. To ensure that these questions have adequately been peer-reviewed, only questions with more than 1000 views are retained. Improvising on quality classes in [12], questions with a score of less than or equal to 0 are considered as *bad*, whereas questions having scores above the 3rd quantile are considered *very good*. The remaining questions are considered readable but do not possess exceptional properties that call for either recommendation or deletion; we therefore label them simply as *good*. To further enhance the quality of our dataset, questions marked by moderators as duplicates were removed from the dataset, whereas questions closed by moderators for reasons including off-topic, subjective and argumentative, not a real question, too localized are also considered as *bad* in the dataset. The above selection criteria results in a total of $55,380$ questions, comprising $12,710$ (23%) good, $34,461$ (62%) neutral and $8,209$ (15%) bad questions. To train the model, the questions are split into training and testing datasets using a 80:20 ratio via stratified sampling.

Data cleaning and pre-processing procedures similar to that of [13] have been applied to minimize out-of-vocabulary words. This includes the removal of programming language snippets, HTML tags and escape characters, URLs and numbers. However, short code snippets and camel-cased words are preserved and normalized because these may contain useful entities that contribute towards the question semantics. This dataset presents challenges in text analysis since it includes noisy text inherent in all user-generated texts, including spelling errors, abbreviations, and low-frequency technical words. Therefore, only the top $3,000$ occurring words are kept to allow the model to focus on the statistically-significant features. Additional experiments using more vocabulary words resulted in no significant difference in performance.

[1][Accessed: March 2019] https://archive.org/details/stackexchange

IV. Experiment setup

Both question title and body are sentence-tokenized and concatenated as input to the model. These words are provided to the proposed hierarchical approaches and existing methods to predict QQ. Existing baseline methods include:

- A linear ridge classifier [5] that employs topic model features at three different granularities;
- A CNN model consisting of two sets of 5×5 convolutional kernels and 2×2 max-pooling layers, followed by a fully-connected layer [2];
- A CNN with 3,4,5-width convolutional kernels is employed to extract contiguous n-gram features followed by a max-pooling layer and a fully-connected layer [14];
- A bi-directional LSTM (Bi-LSTM) max-pooling network [15] that extracts the most representative features in both forward and backward directions;
- Transformer (a state-of-the-art encoder neural network model) that employs multiple layers of self attention to generate contextual representations for each word. Similar to the implementation of $\text{BERT}_{\text{BASE}}$-classifier in [16], the encoded representation from the first timestep is used as the question representation, which is subsequently passed to a fully-connected layer for classification.

Word embeddings of all neural models are initialized with pre-trained `glove.6B.300d` GloVe [17] embeddings before fine-tuning during the training process. The sizes of the hidden unit of all neural encoder layers (except transformer) and the attention vector parameters u_w and u_s are tuned amongst $\{50, 100, 150, 200\}$. The transformer maintains 300 hidden units at every layer and utilizes five attention heads for feature extraction. For the TwAtt layer, an LDA topic model is trained with $\alpha = 0.01, \beta = 0.01$ for 100 passes over the dataset to obtain 20 question topics. Only the set of $K = 10$ words for the most representative topic in each question is used. These topic word embeddings v_k are also initialized similarly with GloVe but fine-tuned separately. Optimization is performed using Adam [18] with its initial learning rate tuned amongst $\{1 \times 10^{-6}, 3 \times 10^{-5}, 1 \times 10^{-5}\}$ and a weight decay of 1×10^{-5}. A grid search over each set of hyper-parameters was performed, combined with five-fold cross-validation for each set. The hyper-parameters and model parameters with the most stable losses and highest F1-score as observed across the five runs is considered the best-performing model and used for evaluation on the test dataset.

TABLE I
COMPARISON ANALYSIS FOR ALL THREE QUALITY CLASSES. METRICS ARE EXPRESSED IN PERCENTAGES %.

Models		Very Good			Good			Bad			Macro-Avg
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**F1**
Human subjects		31.52	41.87	33.66 $\pm$6.75	63.49	38.31	43.63 $\pm$22.00	66.67	25.36	35.06 $\pm$5.03	37.45 $\pm$9.52
Baselines	Linear [5]	35.46	14.87	21.01	58.74	86.47	69.96	38.46	1.22	2.36	31.11
	CNN [14]	27.42	1.34	2.55	62.31	98.88	**76.45**	20.00	0.18	0.36	26.45
	Bi-LSTM [15]	25.58	12.98	17.22	62.55	84.29	71.81	17.21	5.18	7.96	32.33
	Transformer [16]	24.09	18.21	20.74	62.64	77.28	69.19	15.98	6.33	**9.07**	33.00
Proposed Approach	HAN [7]	42.41	22.31	29.23	63.73	89.10	74.31	30.10	1.89	3.55	35.70
	tHAN (40-topics)	40.99	23.80	**30.11**	64.01	87.29	73.86	34.33	4.20	7.49	**37.15**

To quantify the label classification performance, precision, recall and the F1-score were employed. Recall is crucial for ensuring that most of the real positives of both *very good* and *bad* are correctly identified by the model. On the other hand, precision is important for the *good* class as it verifies the model against overconfidence in its predictions for this dominant class. A balance is sought between these two metrics via the F1-score (harmonic mean between recall and precision) to quantify classification performance for all three classes.

To further verify the performance, human subjects were consulted to establish a benchmark for the challenge in identifying quality of questions on Stack Overflow. Four experienced Python programmers were involved and each subject is given a set of stratified-sampled 100 questions from the test split. These subjects were briefed on the desirable characteristics of questions before starting their annotation processes.

V. RESULTS AND DISCUSSION

A. Comparison Analysis

QQ performance obtained from the baseline models and our proposed approaches are provided in Table I. Although the classification results of all three classes are provided, only those for *very good* and *bad* quality questions are of interest since the identification of these questions are important for maintaining the overall quality of site content. These models are categorized into three groups, namely human subjects, linear and sequential baselines, and the proposed hierarchical models. The lower-than-expected scores by human subjects is attributed to a high variation in their perception of question quality in the presence of the dominant *good* class. Our proposed hierarchical modelling of questions strikes a good balance between precision and recall, thus achieving the highest F1-scores of 37.15 in overall *macro-averaged* and of 30.11 in the individual *very good* class. In terms of overall performance quantified by *macro-averaged* F1-score, the linear and sequential neural models performed worse due to the generally lower precision scores of below 40% and 30% in both *very good* and *bad* classes, respectively. This is because many of these questions have been misclassified into the dominant *good* class. This highlights the limitation of CNN and Bi-LSTM sequential models in terms of their capability of modelling questions for quality, since all segments of the questions are considered equally without attention. Transformer, being one of the best sequential encoders, is able to outperform CNN and Bi-LSTM due to the effectiveness of self-attention in modelling context. Although it achieves the highest F1 for *bad* questions, the difference against the proposed tHAN model was not significant enough to compensate for its lower *macro-averaged* F1. The CNN model as described in [2] has also been implemented. However, it is found that the model has been developed for a customized labelling function that discriminates only between two classes, therefore performing modestly lower for this dataset.

By overcoming limitations of sequential modelling, the proposed hierarchical approach customized with topic words achieves the best results with over 4% improvement against the linear baseline method, represented by the transformer. The proposed approaches achieve comparable performance with human annotators in identifying the *very good* questions and overall F1-score. Two examples are presented in Fig. 1 to demonstrate the efficacy of a hierarchical model in selecting discriminative features at sentence level, which was effectively learned by the proposed approach. In the first example, the proposed model assigns a higher weightage to the first sentence that contains a detailed description of the technical problem from the asker. While an individual human annotator may not find this technical information as important for the problem as the community does, the model being trained on large amount of data is able to identify this as a crucial piece of information to determine a quality question. The second question, on the other hand, was written based on poor research. Our proposed model correctly identifies the sentence as an expression commonly found under such cases. In these two exemplar analyses, the hierarchical approach achieves higher performance compared to sequential ones. Similar trend was observed for most of the noisy text in the questions over the dataset.

B. Ablation analysis and effect of topic numbers

Arguably, topic words may be obtained simply from the asker-assigned question tags instead of words as indicated by latent topics trained from the LDA model. The effectiveness of topic words from both these sources is compared using

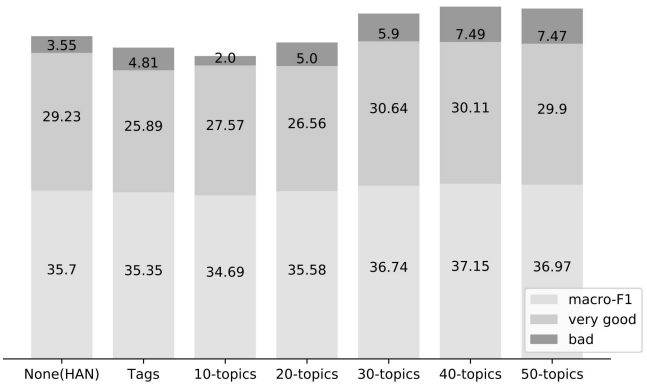

Fig. 4. Effect of number of trained topics on F1 (%) for HAN and tHAN.

a stacked bar chart corresponding to three categories of F1-scores in Fig. 4. In general, higher number of topics trained with LDA improves performance against HAN with none at all. This is due to the model achieving better context that is learned from the topic attention mechanism. Modelling better context therefore allows the model to learn a suitable sentence representation, which, as a consequence, relieves the burden of sentence attention module to select the important sentence for classification as in Fig. 1.

The use of question tags modestly reduced the performance of tHAN. This is because many questions are tagged with highly-specific technical jargon which do not frequently occur across questions, therefore making them unsuitable for modelling general contexts. The above problem was mitigated by introducing topic words from LDA. It can be observed that as the number of trained topics increases, the classification performance on QQ improves. This trend continues until 40 topics, after which performance starts to reduce. This is because an increase in topics learned introduces more diversity to segregate the common types of problems encountered. This can be seen from Table II in which words from the three most representative topics are presented. It can be observed that these topics form the general type of questions being asked on Stack Overflow; some common types of questions can be inferred as follows—Topic 1: asking about a problem involving an error code; Topic 2: appropriate ways of passing parameters in API documentations; Topic 3: installation and package issues. However, as the number of trained topics continues to increase to 50, overfitting occurs and some uncommon topics become too noisy to model question contexts. Using words from these topics therefore results in noisy features and, as a result, the overall performance decreases as observed from Fig. 4. These results show that using words from topic models at an appropriate level is effective at guiding the model towards learning semantic properties at the sentence and word levels to form an effective representation for QQ.

Overall, the proposed hierarchical approach outperforms the state-of-the-art transformer encoders with significantly fewer parameters. This underpins the fact that extracting features via hierarchical selection is important for QQ in noisy user-generated questions. This model is particularly more effective when coupled with a topic attention module that introduces global information about common topic structures in the corpus. We note that the performance for identifying bad quality questions is still modestly poor since many were classified as the *good* questions.

TABLE II
TOP TOPICS LEARNED USING THE BEST MODEL.

Topic 1	error, code, python, get, trying, following, using, problem, tried, help
Topic 2	type, argument, arguments, pass, parameters, parameter, default, documentation, passing, set
Topic 3	python, install, import, module, installed, lib, py, path, packages, version

VI. Conclusion

We propose a neural network architecture that automatically evaluates QQ on community question answering sites. User-generated texts on these sites are often noisy and require customized processing methods to extract relevant features from only the salient parts. To address this issue, a hierarchical model is proposed to aggregate relevant information over textual features at word and sentence levels using neural attention. In addition, a new attention mechanism is developed. This mechanism introduces global topical information from the corpus trained via topic models to complement the hierarchical model. Topic words are useful to distinguish between problem contexts that serve as useful information for the model to vary its attention scheme during the processing of a question. Experiments conducted on the Stack Overflow dataset show that our proposed approach is effective at exploiting these additional features that represent a given question, which, as a consequence, outperforms existing QQ prediction approaches without the use of any platform social indicators as features.

Acknowledgment

This work was conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc. and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

References

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *Proc. Int. Conf. Web Search and Data Mining*, 2008, pp. 183–194.

[2] Y. Zheng et al., "Quality prediction of newly proposed questions in CQA by leveraging weakly supervised learning," in *Proc. Adv. Data Mining and Appl.*, 2017, pp. 655–667.

[3] S. Krishnan, J. Patel, M. J. Franklin, and K. Goldberg, "A methodology for learning, analyzing, and mitigating social influence bias in recommender systems," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 137–144.

[4] A. Baltadzhieva and G. Chrupaa, "Predicting the quality of questions on Stackoverflow," in *Proc. Recent Advances Natural Lang. Process.*, 2015, pp. 32–40.

[5] S. Ravi, B. Pang, V. Rastogi, and R. Kumar, "Great question! Question quality in community Q&A," in *Proc. Int. AAAI Conf. Web and Social Media*, 2014, pp. 426–435.

[6] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[7] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2016, pp. 1480–1489.

[8] T. Luong, H. Pham, C. Manning, "Effective approaches to attention-based neural machine translation" in *Proc. Conf. Empirical Methods in Natural Lang. Process. (EMNLP)*, 2015, pp. 1412-1421.

[9] D. Bahdanau, K. Cho, Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993-1022, 2003.

[11] A. Vaswani, "Attention is all you need," in *Proc. Advances Neural Information Process. Syst.*, 2017, pp. 5998–6008.

[12] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, and D. Fullerton, "Improving low quality Stack Overflow post detection," in *Proc. IEEE Int. Conf. Software Maintenance and Evolution*, 2014, pp. 541–544.

[13] A. Shirani, B. Xu, D. Lo, T. Solorio, and A. Alipour, "Question relatedness on Stack Overflow: The task, dataset, and corpus-inspired models," arXiv:1905.01966 [cs], 2019.

[14] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods in Natural Lang. Process. (EMNLP)*, 2014.

[15] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods in Natural Lang. Process. (EMNLP)*, 2017, pp. 670–680.

[16] T. Wolf *et al.*, "HuggingFace's transformers: State-of-the-art natural language processing," arXiv:1910.03771 [cs], 2019.

[17] J. Pennington, R. Socher, and C. Manning, "Glove: global vectors for word representation," in *Proc. Conf. Empirical Methods in Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[18] D. P. Kingma, J. Ba, "Adam: a method for stochastic optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.