A Deep Neural Network Framework for English Hindi Question Answering

DEEPAK GUPTA, ASIF EKBAL, and PUSHPAK BHATTACHARYYA, Department of Computer Science and Engineering, Indian Institute of Technology Patna, India

In this article, we propose a unified deep neural network framework for multilingual question answering (QA). The proposed network deals with the multilingual questions and answers snippets. The input to the network is a pair of factoid question and snippet in the multilingual environment (English and Hindi), and output is the relevant answer from the snippet. We begin by generating the snippet using a graph-based language-independent algorithm, which exploits the lexico-semantic similarity between the sentences. The soft alignment of the question words from the English and Hindi languages has been used to learn the shared representation of the question. The learned shared representation of question and attention-based snippet representation are passed as an input to the answer extraction layer of the network, which extracts the answer span from the snippet. Evaluation on a standard multilingual QA dataset shows the state-of-the-art performance with 39.44 Exact Match (EM) and 44.97 F1 values. Similarly, we achieve the performance of 50.11 Exact Match (EM) and 53.77 F1 values on Translated SQuAD dataset.

CCS Concepts: • **Information systems** → **Retrieval tasks and goals**;

Additional Key Words and Phrases: Question answering, gated recurrent units, neural networks, attention mechanism, low-resourced languages, snippet generation, character embedding

ACM Reference format:
Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. A Deep Neural Network Framework for English Hindi Question Answering. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19, 2, Article 25 (November 2019), 22 pages.
https://doi.org/10.1145/3359988

1 INTRODUCTION

With the abundance of digital information on the web, the need for accessing the precise information has increased tremendously during the past few years. However, the information is not only limited to a particular language—the web is full of multilingual information. A multilingual question answering (MQA) system can extract the precise answer(s) to a given question from the various sources of information, regardless of the language of the question or the information sources. Such a system facilitates the users to interact and receive the query-specific information

Asif Ekbal acknowledges Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

Authors' addresses: D. Gupta, A. Ekbal, and P. Bhattacharyya, Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, Bihar, 801103, India; emails: {deepak.pcs16, asif, pb}@iitp.ac.in.

2375-4699/2019/11-ART25 $15.00
https://doi.org/10.1145/3359988

Table 1. Sample Multilingual Questions, Answers, and Snippet from Documents on a Given Domain (Tourism)

English Snippet (information): Shimla is the capital of Himachal Pradesh and was also the summer capital in pre-independence India. Covering an area of 25 sq km at a height of 7,238 ft Shimla is surrounded by pine, deodar and oak forests.

Hindi Snippet (information): शिमला, एक ख़ूबसूरत हिल स्टेशन है जो हिमाचल प्रदेश की राजधानी है।
Trans: Shimla is a beautiful hill station, which is the capital of Himachal Pradesh.

Ques(1): हिमाचल प्रदेश की राजधानी क्या है?
(**Trans:** What is the capital of Himachal Pradesh?)
Answer(s): [Shimla, शिमला (**Trans:** Shimla)]
Ques(2): What is the capital of Himachal Pradesh?
Answer(s): [Shimla, शिमला (**Trans:** Shimla)]

Ques(3): शिमला का क्षेत्रफल कितना है?
(**Trans:** How much area is covered by Shimla?)
Answer(s): [25 sq km]
Ques(4): What is the height of Shimla from sea level?
Answer(s): [7,238 ft]

from various multilingual information sources, which may not be available in their native languages. Let us consider the following example from Table 1:

Ques: शिमला का क्षेत्रफल कितना है?
(**Trans:** *What is the area of Shimla?*).

Even though the answer to this question is not available in Hindi (HI) information source, it can be retrieved (**25 sq km**) from the English (EN) source. The linguistic diversities (e.g., morphological, lexical, syntactical) across the languages of a question, document, and answer, further add to the challenge of an MQA system. An efficient MQA system provides the facility to retrieve the answers across multilingual information sources.

Indian languages are *not so fortunate* in terms of resources, tools, and their performance [AP et al. 2014]. Hence, in this work, we propose and develop an MQA system that can leverage the benefit of utilizing resources and tools available in a *fortunate* language, such as English. Towards this, we utilize the popular English QA dataset, SQuAD [Rajpurkar et al. 2016] to generate our synthetic English-Hindi dataset. In the recent work on English/Hindi QA [Sahu et al. 2012; Sekine and Grishman 2003; Stalin et al. 2012], the focus is on passage extraction by considering only lexical similarity. It does not take into account the semantic information to curate the probable sentences where the answer could lie. This set of curated sentences is also known as a snippet. The snippets are automatically anchored around the question terms. First, we propose a snippet generation algorithm, the inputs to the algorithm are a question and a set of documents, and output(s) is (are) the most probable sentence(s) supporting the evidence containing the answer(s). The algorithm takes into account the semantic information with lexical similarity to rank the probable sentences by considering its relevance to the question. Along with this, we represent the sentences of documents as a graph, where each pair of the sentences are linked based on their lexico-semantic similarity (obtained through word embeddings) towards the question. Recently, Joty et al. [2017] proposed an adversarial network to rank the community question under the cross-lingual setting. Gupta et al. [2018a] proposed an approach (neural based) for question generation and question answering in English-Hindi code-mixed scenario. However, the deep neural architecture has not yet been explored for the multilingual QA, especially to extract/generate the answer.

We propose a unified deep neural network framework to retrieve the multilingual answer by exploring the attention-based recurrent neural network to generate the adequate representation of multilingual question and snippets. We utilize the soft-alignment of words from English and Hindi question to generate a single shared representation of questions. The effectiveness of the proposed system is demonstrated to extract the answer of an English and/or Hindi question from English and/or Hindi snippet. Our experiments on a recently released multilingual QA dataset show that our proposed model achieves the state-of-the-art performance. For multilingual settings, our model has shown significant performance improvement over the baselines.

The major contributions of this work are as follows: **(i)** we propose a unified end-to-end deep neural network model for multilingual QA, where question and answer can be either in English or Hindi or both; **(ii)** we introduce a language-independent snippet generation algorithm by leveraging the property of a word embedding; **(iii)** we introduce a technique to learn the shared representation of question from different languages; and **(iv)** we build a model that achieves the state-of-the-art performance for multilingual QA.

2 RELATED WORK

In the work of Sorokin and Gurevych [2017], entity linking is performed prior to forming a SPARQL query. A convolutional neural network is employed for this purpose. The recent trend is to use an end-to-end machine learning approach, for *simple questions dataset* [Bordes et al. 2014]. This work, further extended by He and Golub [2016], makes use of specific characters instead of words as input. Yin et al. [2016] use attentive convolutional networks, and Ture and Jojic [2017] used simple recurrent networks for QA. In recent years, plenty of machine reading comprehension (MRC) models have been developed.

A Bi-Directional Attention Flow (in short, BiDAF) network for reading comprehension is proposed in Seo et al. [2017]. BiDAF consists of a hierarchical architecture to encode the context representation at different levels of granularity. It encodes the words in question and context by three different levels of embeddings: character, word, and contextual. The selling point of the architecture is the use of bi-directional attention flow from a query (question) to paragraph and vice versa, which provides complementary information to each other. With the help of bi-directional attention, they compute the query-aware context (paragraph) representation. The attention operation is performed at each time step and to obtain an attended vector. The obtained attended vector and representations from the previous layers is passed to the next layer in the architecture.

A two-stage network for question answering is proposed by Tan et al. [2018]. The first stage deals with the extraction of relevant span (evidence) to the question from the document. The second stage of the network is responsible for synthesizing the answer from the extracted sentences. The first stage of the network is a multi-task model focused on **(1)** evidence extraction and **(2)** passage ranking. The authors choose a passage ranking task for better evidence prediction. The synthesized model is a seq2seq learning framework [Sutskever et al. 2014] to generate the answer by using the extracted evidence as an additional feature to the model.

Match-LSTM model [Wang and Jiang 2017] proposed a neural-based solution for machine comprehension task. The proposed framework is based on the match-LSTM and Pointer Net Vinyals et al. [2015] to point the answer in the given input context or passage. The model provides two different ways to obtain the answer: *sequence* and *boundary*. In the *sequence* model, proposed architecture predicts the sequence of answer tokens. In the *boundary* model, it only predicts the start and end indices of the answer in the original passage. The words present between the start and end indices are considered to be the answer sequence. The *boundary* model performs better compared to the *sequence* model. Recently, Hu et al. [2018] introduced the reinforced mnemonic reader for MRC tasks. The proposed model improves the attention mechanism by introducing a re-attention mechanism to re-compute the current attentions. In addition tho this, the authors also introduced the dynamic-critical reinforcement learning, which dynamically decides the reward need to be maximized.

The QANet model [Yu et al. 2018] is different from the other neural-based approaches for reading comprehension. The majority of the approaches exploit the RNNs (LSTM or GRU) and attention mechanism. Unlike the other approaches, QANet focused on convolution and self-attention technique.

However, most of these existing studies are in resource-rich languages such as English, which is difficult to port into the other relatively low-resource language (Hindi). In the literature, we see very few attempts to multilingual QA [Bowden et al. 2007; Forner et al. 2008; Giampiccolo et al. 2007; Matteo et al. 2001; Olvera-Lobo and Gutiérrez-Artacho 2011]. The majority of these works made use of machine translation, where question and/or documents in less-resourced languages were translated to the resource-rich language(s) like English. The motivation has been to utilize the resources and tools available in resource-rich languages. García Santiago and Olvera-Lobo [2010] described the main characteristics of multilingual QA systems. Further, they analyzed the quality of the output produced by the machine translation systems (Google Translator,[1] Promt[2] and Worldlingo[3]). The obtained results show the potential in the context of multilingual question answering.

AP et al. [2014] proposed Correlational Neural Networks (CorrNet) to learn the shared representation for the two different aspects (view) of the data. CorrNet maximizes the correlation among the different views of the data when they are projected in a common subspace. The proposed approach does not rely on word-level alignment to learn the bilingual representation. The proposed auto-encoder-based approach learns the representations of bag-of-words of aligned sentences, within and between languages. This cross-language learning representation is useful for multilingual question answering. Deep Canonical Correlation Analysis (DCCA) [Andrew et al. 2013] is another method to learn nonlinear transformations of two views of data. Similar to CorrNet, DCCA also learns the resulting representations are linearly correlated. The DCCA is the non-linear extension of the linear method, canonical correlation analysis (CCA) [Hardoon et al. 2004]. In a different line of research, Das et al. [2016] proposed an approach called SCQA design to find semantic similarity between the two questions. The approach is based on the architecture of Siamese Convolutional Neural Network. The proposed network consists of two convolutional neural networks with shared parameters and a loss function (contrastive) joining them. The aim of the proposed model is to project the semantically similar questions close to each other and dissimilar questions far from each other in the semantic space. There are some other existing works [Gupta et al. 2018; Maitra et al. 2018] on semantic question matching in line with Das et al. [2016].

In another work of community question answering the quality of the answer is predicted using the technique proposed in Suggu et al. [2016] by proposing "Deep Feature Fusion Network (DFFN)," which takes advantage of fusion of two features: the hand-crafted and neural network–based features. The DFNN architecture takes the question-answer pair and associated metadata as inputs and provides the neural network–based feature as the output. It also has the capability to generate the hand-crafted features with the help of various external resources. These two features are fused by projecting the new features into a different vector space with the help of fully connected network. The network assesses the quality of the answer given a question.

[1] https://translate.google.com/.

[2] https://www.online-translator.com/.

[3] http://www.worldlingo.com/microsoft/computer_translation.html.

There have been a very few initiatives with a focus on Hindi QA [Kumar et al. 2005; Sahu et al. 2012; Stalin et al. 2012]. Sekine and Grishman [2003] proposed an English-Hindi cross-lingual QA system using a translation-based approach. But none of these attempts is on English-Hindi multilingual QA.

In our earlier attempt [Deepak Gupta and Bhattacharyya 2018], we have proposed a multilingual QA setup involving English and Hindi. However, our current work significantly differs from this in terms of the following points: **(i)** the current work leverages the rich English QA dataset SQuAD [Rajpurkar et al. 2016] to build an efficient and elegant deep learning model for English-Hindi QA, while the earlier work [Deepak Gupta and Bhattacharyya 2018] deals with information retrieval (IR)-based solution for the English Hindi QA; **(ii)** in this work, we propose a snippet generation algorithm for the passage retrieval, but our earlier work [Deepak Gupta and Bhattacharyya 2018] makes use of a simple heuristic-based scoring; **(iii)** instead of relying on English translation of Hindi question, as we have done in Deepak Gupta and Bhattacharyya [2018], we propose here a mechanism to encode the multilingual question in single shared representation; and **(iv)** our current network is able to handle the question and passage from both the languages without translating them into a single language as in Deepak Gupta and Bhattacharyya [2018].

3 PROPOSED MODEL FOR MULTILINGUAL QA

We propose a unified deep neural network–based approach for multilingual QA. The proposed network, while training, takes as an input the triplets of $< question, snippet, answer >$ for both English and Hindi languages. The trained model can take the multilingual question and snippet[4] as inputs and is able to provide the answer, irrespective of the language of the question or snippet.

We have conducted experiments with two datasets: **(1)** Translated SQuAD and **(2)** Multilingual QA. The multilingual QA dataset consists of the documents containing the passages against each question. We generate the snippet from the whole document in a question-focused summarization fashion. In the case of Translated SQuAD dataset, the paragraph (snippet) containing the answer is available for each question. The proposed algorithm for snippet generation is described as follows:

3.1 Snippet Generation

In snippet generation module, we attempt to extract the sentence(s) that contain the possible answer(s). It is a preliminary step in question answering (QA) system, which reduces the search space of answer from a document containing multiple paragraphs/sentences to a few sentences answer. In the literature, snippet generation is closely related to the task of retrieving candidate answer passage or sentences. Towards this, Tymoshenko and Moschitti [2015] exploit the syntactic parsers (shallow and deep) to obtain the syntactic and semantic structure for the task of candidate answer passage re-ranking. Yang et al. [2016b] proposed a learning to rank approach for answer sentence retrieval. They use the combination of different features such as semantic, context, and text matching features to learn using the models MART [Friedman 2001], LambdaMART [Wu et al. 2010], and Coordinate Ascent (CA) [Metzler and Bruce Croft 2007]. Recently, Yang et al. [2016a] built a neural matching model based on attention mechanism to rank the short answer sentences. A ranking answers model proposed by Yang et al. [2016a] achieved the satisfactory performance without any hand-crafted features. These approaches deal with mono-lingual question/passages and achieve good performance for ranking the candidate sentences containing the answer.

However, in our work, we have question and document in multilingual forms. The existing deep learning–based approaches [Tymoshenko and Moschitti 2015; Yang et al. 2016a, 2016b] may not be feasible in our work, because of the following reasons: **(a)** requires sufficient amount of labelled

[4] In this work, we use the term "snippet" to represent the paragraph containing the answer.

data to train the model and **(b)** the model should have the capability to process the multilingual inputs. Therefore, in this work, we propose an unsupervised approach with the flexibility to deal with the language-independent question/passage.

Our snippet generation algorithm is motivated from the passage retrieval task [Otterbacher et al. 2009], where graph-based query-focused summarization technique is used to retrieve the relevant passage. For a given question q and a set of sentences $S = \{s_1, s_2, \ldots, s_n\}$, the proposed algorithm calculates the relevance score to each sentence $s \in S$ with respect to the question, as shown below:

$$p(s|q) = d\frac{rel(s,q)}{\sum_{p \in C} rel(p,q)} + (1-d)\sum_{v \in C}\frac{rel(s,v)}{\sum_{z \in v} rel(z,v)}p(v|q), \tag{1}$$

where d is termed as "question bias" factor and $C = S - \{s\}$.

The first component of Equation (1) determines the relevance of sentence s to the question q and the second component finds out its relevance to the other sentence. The term d is a trade-off between the two components in the equation and is determined empirically.[5] We force the system to give more importance to the relevance of the question by providing a higher value of d in the 1. The Equation (1) is computed with the help of power method as discussed in Otterbacher et al. [2009]. The term $rel(X, Y)$ is the standard relevance score, which can be computed as follows:

$$\begin{aligned} V_{X(Y)} &= \sum_{w \in X(Y)} log(1 + tf_{w,X(Y)}) * idf_w * Ma_w, \\ rel(X,Y) &= cosine(V_X, V_Y). \end{aligned} \tag{2}$$

Here, $tf_{w,X(Y)}$ is the frequency of word w in $X(Y)$, idf_w is the inverse document frequency of word w. $M \in \mathbb{R}^{d \times |V|}$ is the d dimensional word embedding matrix of vocabulary V word w represented by their one hot vector representation a_w. The terms, V_X and V_Y are the lexico-semantic representation of the entities X and Y, respectively. The vector $V_{X(Y)}$ is normalized to avoid the biasness towards long sentence. The sentences are ranked based on their relevance to the user's question. The top-most ranked three sentences are considered as the candidate to belong to a snippet in our proposed multilingual network. Whenever the system encounters the question in Hindi and documents are in English or vice versa, it translates the Hindi text into English using the Google translator.[6] We use the English-Hindi multilingual embedding trained via the technique discussed in Smith et al. [2017], which helps the snippet generation technique to consider the multilingual words.

In this work, we attempt to solve the multilingual question answering problem, especially in English-Hindi languages. Our proposed method employs a unified deep neural network–based model, with the capability of processing the English and Hindi question/document/snippet and providing the answer. The proposed model consists of multiple layers and is trained with English and Hindi question and documents simultaneously. We train question and snippet for both the languages simultaneously, as we want to adopt the cross-lingual and multilingual settings in a unified model.

In an ideal unified multilingual QA model, the model should have the capability of processing multilingual inputs (question, snippet) and providing the answer, irrespective of the language of question or snippet. To build a multilingual QA model, which is close the ideal multilingual QA model, we propose the QA model. The model is having the capability of processing the multilingual inputs via the *Multilingual Sentence Encoding* layer. We introduce the *Shared Question Encoding* layer, which generates the shared representation of multilingual question. We achieve the

[5] The value of d is set to 0.8 in our experiment.

[6] https://translate.google.com/.

capability of processing the multilingual question via this layer. We introduce an attention-based *Snippet Encoding* layer, which is necessary to encode the question-aware snippet representation. Since we deal with the two languages, English and Hindi, the desired answer can be from any of the two languages. To provide this support in our model, we utilize two pointer networks—one will point and index the answer from English snippet and the other from the Hindi snippet.

Our model consists of multiple layers and is trained with English and Hindi question and document simultaneously. The reason to train question and snippet from both the languages simultaneously is to adopt cross-lingual and multilingual settings in a unified model. The first *Multilingual Sentence Encoding* layer encodes the question and snippet, which are in English and/or Hindi. This layer exploits the multilingual embedding to represent the multilingual words from question and snippet. The word representation is used by Bi-GRU to generate the representation of question and snippet. Our model consists of the *Shared Question Encoding* layer, which takes the English and Hindi question representation and generates the shared representation of the question. We generate the shared representation of question, because the English and Hindi questions are the same asked in different languages. The shared representation is generated by the soft-alignment of words between English and Hindi questions. The *Snippet Encoding Layer* is a self-matching layer that provides the flexibility to dynamically collect information for each word by exploiting the information of the whole snippet. Finally, we have *Answer Extraction Layer*, which is based on the pointer network, which points the start and end answer indices from the snippet. We now describe the individual components of the proposed neural network model as follows:

3.2 Multilingual Sentence Encoding Layer

This layer is responsible to encode the multilingual question and snippet. Given an English question $Q_e = \{w_1^{Q_e}, \ldots, w_{m_e}^{Q_e}\}$, English snippet $S_e = \{w_1^{S_e}, \ldots, w_{n_e}^{S_e}\}$, Hindi question $Q_h = \{w_1^{Q_h}, \ldots, w_{m_h}^{Q_h}\}$, and English snippet $S_h = \{w_1^{S_h}, \ldots, w_{n_h}^{S_h}\}$, word-level embeddings $\{x_t^{Q_e}\}_{t=1}^{m_e}$, $\{x_t^{S_e}\}_{t=1}^{n_e}$, $\{x_t^{Q_h}\}_{t=1}^{m_h}$ and $\{x_t^{S_h}\}_{t=1}^{n_h}$ are generated from pre-trained multilingual word embedding table. To tackle the out-of-vocabulary (OOV) words, we employ character-level embedding $\{c_t^{Q_e}\}_{t=1}^{m_e}$, $\{c_t^{S_e}\}_{t=1}^{n_e}$, $\{c_t^{Q_h}\}_{t=1}^{m_h}$ and $\{c_t^{S_h}\}_{t=1}^{n_h}$. The character-level embeddings are generated by taking the final hidden states of a bi-directional gated recurrent units (Bi-GRU) Chung et al. [2014] applied to embeddings of characters in the token. The final representation of each word $u_t^{Q_e}$ $(u_t^{Q_h})$ of English (Hindi) question and snippet $u_t^{S_e}$ $(u_t^{S_h})$ are obtained as follows:

$$\begin{aligned} u_t^{Q_k} &= \text{Bi-GRU}\left(u_{t-1}^{Q_k}, \left[x_t^{Q_k} \oplus c_t^{Q_k}\right]\right), \\ u_t^{S_k} &= \text{Bi-GRU}\left(u_{t-1}^{S_k}, \left[x_t^{S_k} \oplus c_t^{S_k}\right]\right), \end{aligned} \tag{3}$$

where $k \in \{e, h\}$ denotes the English(e) and Hindi(h) languages, $\oplus$ is the concatenation operator.

3.3 Shared Question Encoding Layer

In this layer, we obtain a shared representation of the encoded English $\{u_t^{Q_e}\}_{t=1}^{m_e}$ and Hindi question $\{u_t^{Q_h}\}_{t=1}^{m_h}$. Basically, we obtain the shared representation via soft-alignment of words [Rocktäschel et al. 2016; Wang et al. 2017] between English and Hindi questions. Since both the questions are same irrespective of their languages, it contains the same information across the languages. With the help of soft-alignment of words between the questions of both languages, we obtain a better representation of a given question (in a language), which considers the same information in other languages. Given English and Hindi question representation $\{u_t^{Q_e}\}_{t=1}^{m_e}$ and $\{u_t^{Q_h}\}_{t=1}^{m_h}$, at first we obtain the English *question-aware* Hindi question representation:

$$v_t^{Q_h} = \text{Bi-GRU}\left(v_{t-1}^{Q_h}, p_t^{Q}\right), \tag{4}$$

where p_t^Q is an attention-based pooling vector. It is calculated as follows:

$$k_j^t = \mathrm{V}^T \tanh\left(\left[W_u^{Q_e} W_u^{Q_h} W_v^{Q_h}\right]\left[u_j^{Q_e} u_t^{Q_h} v_{t-1}^{Q_h}\right]^T\right),$$
$$p_t^Q = \sum_{i=1}^{m_e}\left(\exp\left(k_i^t\right) \Big/ \sum_{j=1}^{m_e} \exp\left(k_j^t\right)\right) u_i^{Q_e}, \tag{5}$$

where V^T is a weight vector, $W_u^{Q_e}$, $W_u^{Q_h}$, and $W_v^{Q_h}$ are the weight matrices.

To compute the representation ($v_t^{Q_h}$) at time t of Hindi question (Equation (4)) using Bi-GRU, we concatenate the pooling vector p_t^Q with the representation ($v_{t-1}^{Q_h}$) at time $(t-1)$. The pooling vector is computed by weighted representation of Hindi question representation $u_t^{Q_e}$ at time t in Equation (5). The Hindi question representation is computed by considering the English question representation; therefore, we called it English *question-aware* Hindi question representation. Similarly, we compute the *Hindi question-aware* English question representation $v_t^{Q_e}$. The shared question representation is obtained by concatenating both the language aware question representations. The final question representation will be $\{v_t^Q\}_{t=1}^{(m_e+m_h)} = \{v_t^{Q_e}\}_{t=1}^{m_e} \oplus \{v_t^{Q_h}\}_{t=1}^{m_h}$.

3.4 Snippet Encoding Layer

The snippet encoding generated from the sentence encoding layer (cf. Section 3.2) does not account question information. To incorporate the question information into the snippet representation, we follow the attention-based recurrent neural network (RNN). We generate the snippet representation of both English and Hindi by taking the shared question information into account. The English snippet representation can be calculated by:

$$v_t^{S_e} = \text{Bi-GRU}\left(v_{t-1}^{S_e}, c_t^{S_e}\right), \tag{6}$$

where $c_t^{S_e}$ is an attention-based pooling vector that can be derived *via* the following equations:

$$k_j^t = \mathrm{V}^T \tanh\left(\left[W_v^Q W_u^{S_e} W_v^{S_e}\right]\left[v_j^Q u_t^{S_e} v_{t-1}^{S_e}\right]^T\right),$$
$$c_t^{S_e} = \sum_{i=1}^{m_e+m_h}\left(\exp\left(k_i^t\right) \Big/ \sum_{j=1}^{m_e+m_h} \exp\left(k_j^t\right)\right) v_i^Q, \tag{7}$$

where W_v^Q, $W_u^{S_e}$, and $W_v^{S_e}$ are the learnable weight matrices. The snippet representation $v_t^{S_e}$ dynamically incorporates aggregated matching information from the whole question. Similarly, we compute the Hindi snippet representation $v_t^{S_h}$. To capture the context information while generating the snippet representation, we introduce an additional layer similar to Wang et al. [2017]. The context plays an important role in discovering the answer from a snippet. This additional layer matches the obtained snippet representation from the *snippet encoding layer* against itself. This layer provides the facility to dynamically collect evidence from the whole snippet for the words in a snippet. It encodes the evidence relevant to the current snippet word and its matching question information into the snippet representation. The final snippet representation for the English snippet can be computed as follows:

$$p_t^{S_e} = \text{Bi-GRU}\left(p_{t-1}^{S_e}, \left[v_t^{S_e}, c_t^{S_e}\right]\right), \tag{8}$$

where $c_t^{S_e}$ is an attention-based pooling vector for the entire English snippet, it is computed in the following manner:

$$\begin{aligned} k_j^t &= \mathrm{V}^T \tanh\left(\left[W_{p'}^{S_e} W_{p''}^{S_e}\right]\left[v_j^{S_e} v_t^{S_e}\right]^T\right), \\ c_t^{S_e} &= \sum_{i=1}^{n_e}\left(\exp\left(k_i^t\right) \Big/ \sum_{j=1}^{n_e} \exp\left(k_j^t\right)\right) v_i^{S_e}, \end{aligned} \tag{9}$$

where $W_{p'}^{S_e}$ and $W_{p''}^{S_e}$ are the learnable weight matrices. We compute the snippet representation for the Hindi snippet following the same way. The final snippet representations that we obtain are $\{p_t^{S_e}\}_{t=1}^{n_e}$ and $\{p_t^{S_h}\}_{t=1}^{n_h}$ for English and Hindi, respectively.

3.5 Answer Extraction Layer

We utilize the pointer network proposed by Vinyals et al. [2015] to extract the answer from the snippet. We use two pointer networks, one to select start (a_e^{start}) and end (a_e^{end}) index of answer from the English snippet and another from the Hindi snippet. Given the English snippet representation $\{p_t^{S_e}\}_{t=1}^{n_e}$, with the help of attention mechanism, networks select the start and end indices of the answer. The hidden state of pointer network is calculated by $h_t^{a_e} = \text{Bi-GRU}(h_{t-1}^{a_e}, c_t^{S_e})$, where $c_t^{S_e}$ is the attention pooling vector. It can be computed as follows:

$$\begin{aligned} k_j^t &= \mathrm{V}^T \tanh\left(\left[W_p^{S_e} W_h^{a_e}\right]\left[p_j^{S_e} h_{t-1}^{a_e}\right]^T\right), \\ a_i^t &= \exp\left(k_i^t\right) / \sum_{j=1}^{n_e} \exp\left(k_j^t\right), \\ c_t^{S_e} &= \sum_{i=1}^{n_e} a_i^t p_i^{S_e}, \\ a_e^t &= argmax(a_1^t, .., a_{n_e}^t). \end{aligned} \tag{10}$$

At first step ($t = 1$) network will predict a_e^{start} and the next step it will predict a_e^{end}. In a similar way, we compute a_e^{end}. Following Equation (10), the answer index a_h^{start} and a_h^{end} from the Hindi snippet are extracted. The structure of the model is depicted in Figure 3.

4 EXPERIMENTS

4.1 Experimental Setup

We perform experiments in six different multilingual settings.

(1) $\mathrm{Q_E} - \mathrm{S_{E+H}}$: The question is in *English* and the answer exists in both *English* and *Hindi* snippets. The model has to retrieve the answer from both the snippets. This setting is equivalent to cross-lingual and multilingual evaluation setup of QA.

(2) $\mathrm{Q_H} - \mathrm{S_{E+H}}$: The question is in *Hindi* and the answer exists in both *English* and *Hindi* snippets. The model has to retrieve the answer from both the snippets. This setting is equivalent to cross-lingual and multilingual evaluation setup of QA.

(3) $\mathrm{Q_E} - \mathrm{S_E}$: Both question and answer are in *English*. The model has to retrieve the answer from the *English* snippet. This setting is equivalent to the monolingual evaluation setup of QA.

(4) $\mathrm{Q_H} - \mathrm{S_H}$: Both question and answer are in *Hindi*. The model has to retrieve the answer from the *Hindi* snippet. This setting is equivalent to the monolingual evaluation setup of QA.

(5) $\mathbf{Q_E - S_H}$: The question is in *English* and the answer exists in the *Hindi* snippet. The model has to retrieve the answer from the Hindi snippet. This setting is equivalent to cross-lingual evaluation setup of QA.

(6) $\mathbf{Q_H - S_E}$: The question is in *Hindi* and the answer exists in the *English* snippet. The model has to retrieve the answer from the English snippet. This setting is also equivalent to cross-lingual evaluation setup of QA.

It should be noted that we train our model with the bi-triplet $< question_e, snippet_e, answer_e >$ and $< question_h, snippet_h, answer_h >$ input from the English and Hindi languages, respectively. Both the triplets have the same information in two different languages. The proposed network is trained to minimize the sum of the negative log probability of the ground truth start and end indices of the answers in both the languages by the predicted probability distributions of the model. By training the network with the bi-triplet of both the languages, the network learns to handle the different settings of multilingual question and snippet. At the time of evaluation, when the network receives question or snippet from one language, we replicate the same for the other language to keep the inputs compatible with the model.

For experiments, we use the publicly available *fastText* [Bojanowski et al. 2017] pre-trained English and Hindi word embeddings of dimension 300. For multilingual word embedding, we align monolingual vectors of English and Hindi in a unified vector space using a learned linear transformation matrix [Smith et al. 2017]. We use the Stanford CoreNLP [Manning et al. 2014] to pre-process all the English sentences. The model with character-level embeddings of dimension 45 shows the highest performance on the validation set. The optimal dimension of hidden units for all the layers is set to 45 in the experiment. We exploit two layers of Bi-GRU to compute character embedding and three layers to obtain the question and snippet representation, respectively. Mini-batch gradient decent (batch size of 50) with the AdaDelta optimizer [Zeiler 2012] is used to train the network with a learning rate of 1. The network is trained for 70 epochs. The hyper-parameters are tuned using a validation dataset.

4.2 Datasets

We use two different multilingual question answering datasets in our experiment to evaluate the performance of the proposed model. Both the datasets are available online.[7]

4.2.1 Translated SQuAD Dataset. We translate 18,454 random English <*question, passage, answer*> triplets from Squad dataset [Rajpurkar et al. 2016] into Hindi. These translated triplets ensure that the answer is a substring of passage. We divide this dataset into train, validation, and test sets. We use a set of 10,454 QA pairs in English and Hindi for training the network. Another set of 2K QA pairs are used to validate the system performance over every epoch. We use a set of 6K QA pairs for evaluating the system performance.

4.2.2 Multilingual QA Dataset. We use the MQA dataset released by Deepak Gupta and Bhattacharyya [2018] to evaluate the model. The detailed statistics of this dataset are given in Table 2. This dataset also provides us with the source documents where the answer exists for the questions. In the practical scenario, we only have a question and need to retrieve its answer from the different documents, not necessarily in the same language as that of the question. With this fact in mind, we perform the experiments by different multilingual settings (cf. Section 4.1). For each question, we generate the snippet following the approach discussed in Section 3.1. This dataset is only used for evaluating the model performance. To compare the performance between

[7] https://bit.ly/2MEkrTQ.

Table 2. Statistics of the Multilingual QA Dataset

Domains	$Q_E - S_E$	$Q_H - S_H$	$Q_E - S_H$	$Q_H - S_E$	$Q_E - S_{E+H}$	$Q_H - S_{E+H}$	Overall
Tourism	456	403	456	403	422	422	1,703
History	110	126	110	126	1,118	1,118	2,472
Diseases	81	33	81	33	48	48	210
Geography	55	29	55	29	174	174	432
Economics	25	14	25	14	682	682	1,403
Environment	9	2	9	2	226	226	463
Overall	**736**	**607**	**736**	**607**	**2,670**	**2,670**	**6,683**

the different multilingual settings, we could only use the data samples listed in the category of $Q_E - S_{E+H}$ and $Q_H - S_{E+H}$.

4.3 Evaluation Scheme

We evaluate the system performance using Exact Match (EM) and F1 metrics following Rajpurkar et al. [2016]. For multilingual setting $Q_E - S_{E+H}$ and $Q_H - S_{E+H}$, we count the correct prediction only when the model produces the correct answer from both the snippets. For the rest of the experimental settings, we count the correct prediction when the model produces the correct answer from the particular snippet.

4.4 Baselines

4.4.1 IR-based QA Model. We develop a translation-based baseline model for the comparison. This baseline is adopted from the state-of-the-art models in English-Hindi QA as proposed by Deepak Gupta and Bhattacharyya [2018]. This baseline is related to the translation-based IR approaches [Forner et al. 2008; Giampiccolo et al. 2007; Matteo et al. 2001] developed for multilingual QA focused on European languages. We also translate Hindi question and articles into English. The details of the component used in this baseline are as follows:

- Document Processing: This step is dealing with the processing of the paragraphs (articles). First, we translate Hindi questions and Hindi articles into English by using the Google Translator.[8] Thereafter, we use the snippet generation algorithm to generate the snippets for each question as proposed in Section 3.1.
- Question Processing: Question processing step consists of two sub-steps: **(1)** *question classification* and **(2)** *query formulation*. We classify each question with the question classes proposed by Li and Roth [2002]. Question class provides us the semantic constraint on the sought-after answer. We adopted the question classification system proposed by Deepak Gupta and Bhattacharyya [2018]. The system classifies each question into coarse and fine classes.

  In the query Formulation step, we obtain the Part-of-Speech (PoS) tags for each question using Stanford PoS tagger.[9] Query is formulated by concatenating all the noun, verb, and adjective words in the same order in which it appears in the question.
- Candidate Answer Extraction: The output of question classification guides the candidate answer extraction step to extract the probable answer from the passage. First, we tag the

[8] https://translate.google.com.
[9] https://nlp.stanford.edu/software/tagger.shtml.

passage with Stanford named entity tagger.[10] Thereafter, we make a list of all the entities (along with the sentence in which it appears) whose entity type is the same as of question classification. The obtained entity list will be considered as the candidate answers.

- Candidate Answer Scoring: In this step, each candidate answer will be assigned a score. As each candidate answer is also associated with their sentence, we calculate the score for each of the candidate answer sentences (A). We calculate the score for each of the candidate answer sentences (A). We use the following scoring techniques to score each candidate answer:

(1) **Term Coverage (TC)**: It computes the number of words that are common in query terms candidate answer sentence. We also normalized it w.r.t. the length of the query (number of words in the query).

(2) **Proximity Score (PS)**: We compute the shortest span that covers the query words contained in the candidate answer sentence. We also normalized it w.r.t. the length of the query.

(3) **Coverage Score (CS)**: First, we compute the coverage of n-gram ($n = 1, 2, 3, 4$) between the query and the candidate answer sentence. Thereafter, the coverage score between a query (q) and a candidate answer sentence (S) is computed as follows:

$$NGCoverage(q, S, n) = \frac{\sum_{ng_n \in S} Count_{common}(ng_n)}{\sum_{ng_n \in q} Count_{query}(ng_n)}, \tag{11}$$

$$NGScore(q, S) = \sum_{i=1}^{n} \frac{NGCoverage(q, S, i)}{\sum_{i=1}^{n} i}. \tag{12}$$

(4) **Word-vector Similarity (WS)**: We represent query and candidate answer sentence using the semantic vector obtained from the word embedding. A similarity score is computed using the cosine similarity between the semantic vector of query and candidate answer. The semantic vector is formulated as follows:

$$\mathtt{SemVec}(X) = \frac{\sum_{t_i \in X} \mathtt{W}(t_i) \times \text{tf-idf}_{t_i}}{number\ of\ look-ups}, \tag{13}$$

where X is query q or candidate answer sentence S, $\mathtt{W}(t_i)$ is the word vector of word t_i. *number of look-ups* represents the number of words in the question for which pre-trained word embeddings[11] are available.

The weighted aggregate score for each candidate answer (A) is computed as follows:

$$S(Q, A) = W_1 * TC + W_2 * PS + W_3 * CS + W_4 * WS \tag{14}$$

Here, W_k is the learning weights for k^{th} scoring. Optimal weight values[12] are chosen based on the system performance on the validation dataset. We choose a candidate having the maximum score as our final answer.

4.4.2 RNN-based QA Model. Similar to the IR-based baseline, we translate[13] the Hindi question and snippet into English. The question and snippet encodings are performed as discussed in Section 3.2. Thereafter, we incorporate the question information into snippet by applying the attention mechanism similar to Equations (6) and (7) to regenerate the snippet representation. This snippet representation of a word (from snippet) at time t is fed to a feed-forward neural network.

[10] http://nlp.stanford.edu:8080/ner/process.

[11] https://code.google.com/archive/p/word2vec/.

[12] Optimal weights are found to be (0.31, 0.18, 0.39, 0.12).

[13] In all baseline models translation is performed using Google translation.

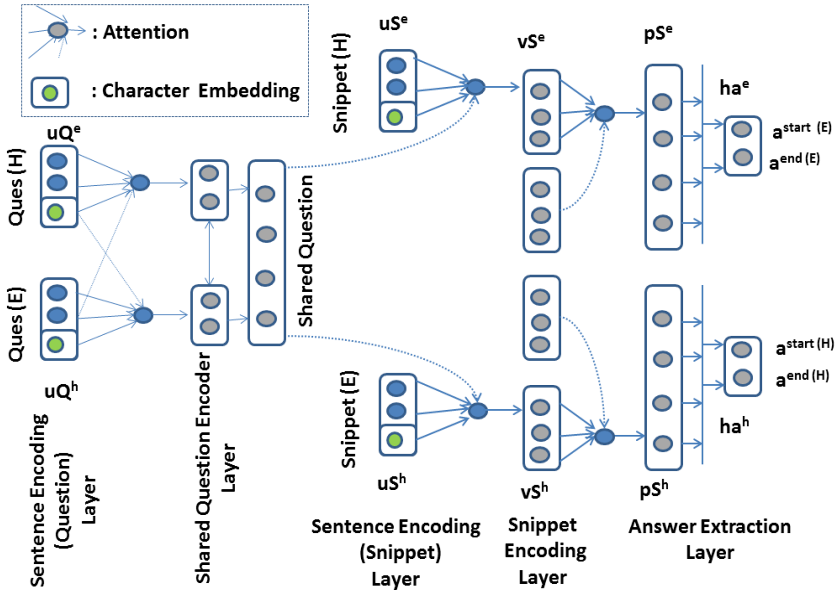

Fig. 1. Structure of the proposed unified deep neural network for MQA. The notations are the same as described in Section 3.

This network computes the vectors of probability score p_t. The length of the probability vector is set to 3, representing the BIO encoding (B-beginning, I-intermediate, and O-outside) of the answer. This model is similar to the attention-based QA-LSTM model proposed by Tan et al. [2015], but instead of computing the similarity between question and snippet as in Tan et al. [2015], we classify the token at time t from the snippet into “B-answer,” “I-answer,” and “O.”

4.4.3 Monolingual (English) QA Model. This baseline is similar to the monolingual version of the proposed network (cf. Section 3). In the first layer of this baseline model, the English question and snippet are encoded as discussed in Section 3.2. As we are dealing with only one language, *shared question encoding layer* is not existing in this particular baseline model. The output of *sentence encoding layer* is passed to the *snippet encoding layer* (cf. Section 3.4). Finally, *answer extraction layer* (cf. Section 3.5) predicts the start and end indices of the answer from the snippets.

4.4.4 Monolingual (Hindi) QA Model. We propose the fourth baseline similar to the monolingual (English) baseline. The structure of the proposed model is depicted in Figure 1. The input question and snippet are in the Hindi language. Hyperparameters of both monolingual models are kept the same as the multilingual model.

4.4.5 Deep Canonical Correlation Analysis (Deep CCA). Deep CCA [Andrew et al. 2013] computes representations of the two views by passing them through multiple stacked layers of nonlinear transformation. We experiment with Deep CCA by treating English and Hindi question representations as two different views of the same question. In our experiment, we use four layers of GRU network to compute the representation of both the views. Basically, from our proposed model, we replace the *Shared Question Encoding* layer with Deep CCA, which computes the shared representation by taking the two question views (representation) as inputs. The goal is to jointly learn parameters for both views such that the correlation between the final obtained

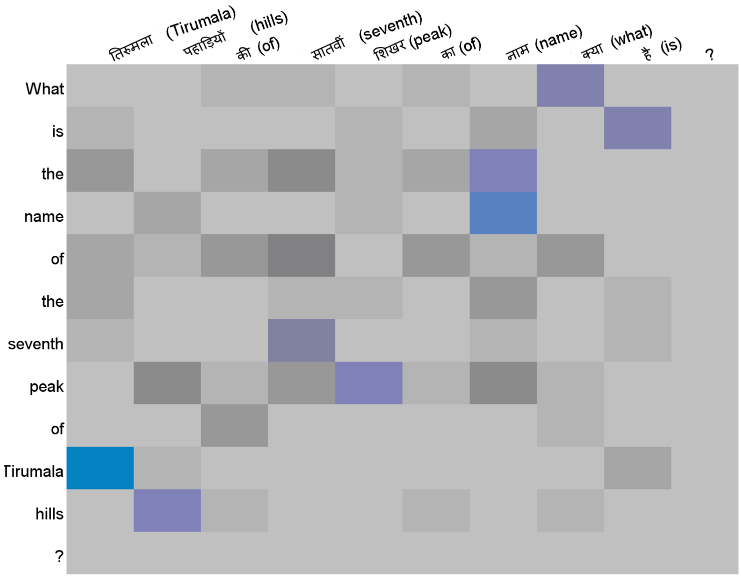

Fig. 2. The soft alignment of words between the same questions in two different languages. The learned attention weight is shown here. It is clearly seen that the model learns the same words across languages.

representations is as high as possible. The hyperparameters of the Deep CCA model are kept the same as the proposed multilingual model.

5 RESULTS AND ANALYSIS

We evaluate the performance of the proposed snippet generation algorithm in terms of mean reciprocal rank (MRR). We achieve the MRR values of 95.48% as compared to the standard Biased LexRank [Otterbacher et al. 2009] of 91.71% on the ground truth passage provided in the multilingual QA dataset. We show the evaluation results on MQA for the multilingual question answering and Translated SQuAD dataset in Table 4 and Table 6 for multilingual QA and Translated SQuAD dataset, respectively. The proposed model achieves 7.23 and 11.7 absolute F1 point increments over the attention-based RNN baseline for the multilingual QA and Translated SQuAD datasets, respectively. Similarly, the proposed model achieves 5.86 and 5.14 absolute F1 point increments over the Deep CCA baseline for the multilingual QA and Translated SQuAD datasets, respectively. Statistical t-test confirms this improvement to be statistically significant (t-test, $p < 0.05$). We observe that $Q_H - S_{E+H}$ performs slightly lower than $Q_E - S_{E+H}$. It may be because of the smaller-size corpus used for generating the Hindi embeddings. To ensure the quality of translation from Google Translate, we perform human evaluation of the Google translation. We randomly choose 100 question-snippet pairs from English (SQuAD) dataset and translate them to Hindi. For translation, we employ two annotators having expertise in both English and Hindi. We computed the BLEU score [Papineni et al. 2002] and found the score as 72.13.

5.1 Analysis and Discussion

In this section, we present the analysis of the results obtained in terms of the effect of shared question encoding and the ablation study. In addition to this, we also compare the quality of answer extracted using the proposed multilingual model and Deep CCA model.

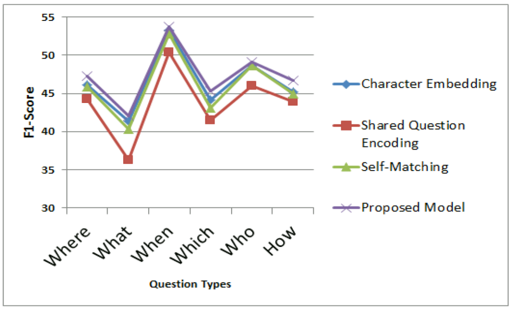

Fig. 3. Effect of model components on the various types of questions from MQA dataset.

Table 3. Results of Ablation Study (by Removing One Model Component at a Time) on Both the Datasets

Models	**Multilingual QA**		**Translated SQuAD**	
	EM	F1	EM	F1
Proposed Model	39.44	44.97	50.11	53.77
-Shared Question Embedding	35.62	41.18	46.37	49.91
-Character Embeddings	38.12	43.26	48.84	52.53
-Self Matching	37.23	42.39	48.02	51.84

5.1.1 Effect of Shared Question Encoding. This layer learns the word or phrase of the question that needs to be given more focus with respect to the question of the other language while generating the question representation. We show in Figure 2 through attention weight that the model learns to align the same/similar words from the questions across the languages (English and Hindi). The effect of shared question representation is evident while we look at the Monolingual (English) and Monolingual (Hindi) baselines performance in Table 4 and Table 6, respectively. Both of these baselines do not have shared question encoding layer. The Monolingual (Hindi) model favors the question and snippet that are in Hindi, and it shows the comparable performance close to the RNN-based baseline for the English question and/or snippet ($Q_E - S_E$, $Q_E - S_{E+H}$). We also observed quite a similar trend for the Monolingual (English) baseline model. The evaluation shows that the proposed multilingual system performs better in all the multilingual settings compared to the monolingual baselines.

5.1.2 Ablation Study. We carefully observe the effect of various components of the model. We show the ablation study in terms of EM and F1 score on the multilingual QA dataset in Table 3. This shows the contribution of important components very clearly. The analysis reveals that shared question encoding represents the questions of two languages very effectively by aggregating the information from the questions. The character embedding helps the model to overcome the out-of-vocabulary words and short words, which are often in Hindi question and snippet. The self-matching of snippet assigns more weights to the words (in a snippet) that are related to the question and the context in which the answer appears. We extend our experiment by analyzing the model performance on the various question types such as *what, where, when, how, which, who*. Figure 3 shows the impact (in terms of F1 score) of model components (by removing a component at a time) on different types of questions of multilingual QA dataset. Our model achieves the best performance on *"when"* type questions. Because *"when"* type questions generally look for "date" and "time" as the answer. However, for "*what*" type questions, the model achieves comparatively low F1 score. This is because ‘ ‘*what*" type of questions look for a long phrase as the answer. The study reveals that the shared question encoding has the higher impact on the performance of the model for all types of questions.

We have translated the question/snippet in baseline 1 and baseline 2 only. We did not translate the question/snippet in our proposed model. The Monolingual (English) and Monolingual (Hindi) model are trained on the question and snippet from the English and Hindi languages, respectively. In the $Q_E - S_{E+H}$ and $Q_H - S_{E+H}$ settings, the model receives the cross-lingual inputs. Therefore, the monolingual model could not achieve as good performance as our proposed multilingual model. The proposed model has the shared question encoder and has the capability of processing

Table 4. Performance Comparison of Proposed MQA Model (on Multilingual QA Dataset) with the Various Baseline Models

	Models	$Q_E - S_E$	$Q_H - S_H$	$Q_E - S_H$	$Q_H - S_E$	$Q_E - S_{E+H}$	$Q_H - S_{E+H}$	**Overall**
		EM (F1)	**EM (F1)**	**EM (F1)**	**EM (F1)**	**EM (F1)**	**EM (F1)**	**EM (F1)**
Baselines	IR-based QA	33.46 (39.81)	32.63 (38.12)	30.24 (32.94)	27.67 (30.04)	32.17 (39.67)	30.78 (37.97)	31.15 (36.42)
	RNN-based QA	37.18 (41.74)	34.75 (40.32)	32.14 (33.85)	28.22 (29.61)	35.49 (41.85)	33.79 (39.12)	33.59 (37.74)
	Monolingual (Hindi)	36.12 (42.67)	41.38 (47.79)	30.97 (33.54)	28.41 (30.08)	38.31 (44.61)	38.71 (44.94)	35.65 (40.60)
	Monolingual (English)	44.17 (49.35)	35.52 (41.11)	31.23 (33.97)	29.11 (31.71)	39.18 (46.64)	35.17 (41.29)	35.73 (40.67)
	Deep CCA	41.21 (43.48)	37.79 (40.23)	31.62 (33.89)	30.34 (32.65)	39.76 (42.23)	38.23 (42.19)	36.49 (39.11)
Proposed Multilingual		44.78 (50.27)	41.46 (48.14)	34.68 (37.89)	33.41 (37.02)	42.28 (49.01)	40.06 (47.49)	39.44 (44.97)

the cross-lingual and multilingual inputs. This is the reason why the proposed model achieves the improvements on $Q_E - S_{E+H}$ and $Q_H - S_{E+H}$ settings compared to the monolingual (English) and monolingual (Hindi) model.

We observe that the model performance on multilingual QA dataset is relatively lower as compared to the Translated SQuAD multilingual dataset. This is because the model is trained on the Translated SQuAD multilingual dataset and learns the diverse answers from the dataset, which may not exist in multilingual QA dataset. Due to the unavailability of any other MQA (EN-HI) dataset, we cannot make any direct comparison. However, our IR-based baseline is the re-implementation of the state-of-the work [Sekine and Grishman 2003] on EN-HI cross-lingual QA and obtains significantly better performance compared to the state-of-the-art model. Most of the available French/German-English dataset (CLEF) is small in size and developed in the cross-lingual setting. However, the dataset used in this work, provide the monolingual, cross-lingual and multilingual settings. To the best of our knowledge, in multilingual settings, where for a given multilingual question, the corresponding answer needs to be extracted from various multilingual snippets, has not yet been addressed in the literature.

5.2 Qualitative Analysis

We qualitatively analyze the answers predicted by the proposed system. The examples are shown in Table 5. The analysis shows that the proposed system performs very well for the question that is looking for the named entity type answer. Our further analysis reveals that the proposed system performs exceptionally well to identify the "*number*," "*date*," "*quantity*," and "*person name*" types of answers.

We closely analyze the major sources of errors in Section 5.3. The model learns to identify the semantically similar words in snippet, and sometimes it predicts the semantically similar words as the answer. We compare the performance of the CCA-based model to the proposed model—both quantitatively and qualitatively. We show the question, snippet, along with their answers predicted from the proposed model and Deep CCA in Table 5. The Deep CCA model suffers from the out-of-context answers. In cross-lingual setups ($Q_H - Q_E$) and ($Q_E - S_H$), the Deep CCA model does not perform well compared to the proposed model. We also observe that Deep CCA model extracts the long sentence answer. The Deep CCA model tries to maximize the correlation between English and Hindi representation and learns the shared question representation. While maximizing the correlation Deep CCA focuses on the question representation as a single vector. In contrast, our shared question encoding layer tries to find the alignment between the English and Hindi question representation by considering each word from English and Hindi question. In addition, our model generates the shared question representation by considering the English-aware Hindi and Hindi-aware English representation (cf. Section 3.3).

Table 5. Examples of Question, Snippet, Gold Answer, and the Predicted Answer Using Deep CCA and Our Proposed Model

Question (1): Which company adopted the ASA scale in 1946?

Snippet: General Electric switched to use the ASA scale in 1946. Meters manufactured since February 1946 were equipped with the ASA scale -LRB- labeled “ Exposure Index “ -RRB- already. For some of the older meters with scales in “ Film Speed “ or “ Film Value “ -LRB- e.g. models DW-48, DW-49 as well as early DW-58 and GW-68 variants -RRB-, replaceable hoods with ASA scales were available from the manufacturer …

Gold Answer: General Electric

Answer using Deep CCA: DW-48

Answer using Proposed Model: General Electric

Question (2): एलजीबीटी के अधिकारों के लिए कौन सा मील का पत्थर माना जाता है?

Trans: Which landmark is considered the spark for LGBT rights?

Snippet: The Statue of Liberty National Monument and Ellis Island Immigration Museum are managed by the National Park Service and are in both the states of New York and New Jersey. … Hundreds of private properties are listed on the National Register of Historic Places or as a National Historic Landmark such as, for example, the Stonewall Inn in Greenwich Village as the catalyst of the modern gay rights movement.

Gold Answer: Stonewall Inn

Answer using Deep CCA: Governors Island National Monument

Answer using Proposed Model: Stonewall Inn in Greenwich Village

---

Question (3): How did naturalism effect the greater world?

Snippet: …But as the 19th-century went on, European fiction evolved towards realism and naturalism, the meticulous documentation of real life and social trends. Much of the output of naturalism was implicitly polemical, and influenced social and political change, but 20th century fiction and drama moved back towards the subjective, emphasising unconscious motivations and social and environmental pressures on the individual.…

Gold Answer: influenced social and political change

Answer using Deep CCA: primacy of individual experience

Answer using Proposed Model: social and political developments

---

Question (4): ज़ार अलेक्जेंडर ने चोपिन को क्या दिया?

(**Trans**: What did Tsar Alexander I give to Chopin?)

Snippet: सितंबर 1823 से 1826 तक चोपिन वारसा लिसेयुम में भाग लिया जहां उन्होंने अपने पहले वर्ष के दौरान चेक संगीतकार विल्हेम वार्फ़ेल से अंग सबक प्राप्त किय ज़ार ने उसे एक हीरे की अंगूठी प्रस्तुत किया 10 जून 1825 को बाद के ईओलोमेलोडिकॉन कॉन्सर्ट में चोपिन ने अपने रोंडो ओप का प्रदर्शन किया

(**Trans:** From September 1823 to 1826 Chopin attended the Warsaw Lyceum, where he received organ lessons from the Czech musician Wilhelm Wurfel during his first year. Tsar presented him with a diamond ring. At a subsequent eolomelodicon concert on 10 June 1825, Chopin performed his Rondo Op)…

Gold Answer: हीरे की अंगूठी

Answer using Deep CCA: रोंडो ओप (**Trans:** Rondo Op)

Answer using Proposed Model: हीरे की अंगूठी (**Trans:** diamond ring)

Question (5): Who is responsible for appointing the Lieutenant Governor of the Union Territory of Delhi?

Snippet: The head of state of Delhi is the Lieutenant Governor of the Union Territory of Delhi, appointed by the President of India on the advice of the Central government and the post is largely ceremonial, as the Chief Minister of the Union Territory of Delhi is the head of government and is vested with most of the executive powers.

Gold Answer: President of India

Answer using Deep CCA: Lieutenant Governor

Answer using Proposed Model: President of India

Question (6): What particle is associated with the yellowing of newspapers?

Snippet: Paper made from mechanical pulp contains significant amounts of lignin, a major component in wood. In the presence of light and oxygen, lignin reacts to give yellow materials, which is why newsprint and other mechanical paper yellows with age …

Gold Answer: lignin

Answer using Deep CCA: lignin

Answer using Proposed Model: lignin

The answers are shown in red.

Table 6. Performance Comparison of Proposed MQA Model (on Test Set of Translated SQuAD Dataset) with the Various Baseline Models

Models		$Q_E - S_E$	$Q_H - S_H$	$Q_E - S_H$	$Q_H - S_E$	$Q_E - S_{E+H}$	$Q_H - S_{E+H}$	**Overall**
		EM (F1)	**EM (F1)**	**EM (F1)**	**EM (F1)**	**EM (F1)**	**EM (F1)**	**EM (F1)**
Baselines	IR-based QA	35.17 (37.78)	32.87 (36.55)	31.45 (33.13)	28.12 (30.69)	34.67 (36.54)	31.22 (35.16)	32.25 (34.97)
	RNN-based QA	44.68 (45.51)	41.24 (44.71)	33.27 (36.89)	31.59 (33.86)	42.56 (46.94)	39.33 (44.54)	38.77 (42.07)
	Monolingual (Hindi)	43.78 (47.41)	49.81 (53.27)	35.01 (38.78)	37.14 (41.85)	47.77 (51.29)	48.18 (52.21)	43.61 (47.46)
	Monolingual (English)	52.49 (56.11)	43.17 (48.37)	41.54 (35.53)	33.11 (37.54)	52.38 (56.61)	45.11 (49.35)	44.63 (47.25)
	Deep CCA	44.78 (50.27)	41.46 (48.14)	42.04 (46.68)	40.84 (44.86)	51.19 (53.38)	45.06 (48.49)	44.28 (48.63)
Proposed Multilingual		53.15 (57.29)	51.34 (53.87)	45.34 (50.24)	44.19 (48.21)	54.38 (58.39)	52.27 (54.67)	50.11 (53.77)

5.3 Error Analysis

We closely analyze the outputs on multilingual QA dataset and come up with the following observations:

(1) The system suffers to predict the correct answer, where the answer entity is the anaphor or cataphor in the snippet. E.g.,
Q: *What is the part of the Adam's Bridge?*,
Gold Answer: Pamban Island
Snippet: ***Pamban Island*** *is situated in the Gulf of Mannar between India and Srilanka …* ***It*** *is a part of the Adam's Bridge.*
As shown in the example the word "**it**" (pronoun) is referring to the phrase **"Pamban Island,"** and these two words are far apart (in terms of the number of words between these two words) in the passage. Therefore, the model could not identify the correct referred phrase **"Pamban Island."** Resolving such pronouns in the snippet before passing it into the network should lead to performance improvements.

(2) Sometimes the system predicts the wrong answer from the snippet. This generally happens in case named entity (NE) appears in the vicinity. E.g.,
Q: *How far is the Taj Mahal from New Delhi?*
Gold Answer: 230 KM;
Predicted Answer: 310 KM
Snippet: *Taj is located within the distance of 310 km and* ***230 Km*** *from Lucknow and national capital New Delhi respectively …*
In this example, there are two numbers (*310 km* and *210 km*) that appear very near in the snippet. The network fails to correctly map the associated number (**230 km**).

(3) While analyzing the outputs of snippet generation, we observe that during translation of Hindi sentences in snippet generation, some synonym words and named entities are incorrectly translated. E.g., **Q:** *When Mahatma Gandhi visited Darjeeling?*
The prompt translation of documents: "..*Mahatma Gandhi traveled to Darjeeling in 1925 …*". The word *visited* has been replaced with *traveled*, so the snippet generation algorithm ranks it lower in order.

(4) Our proposed network sometimes could not identify the correct start or end index of the answer in the snippet. It contributes to the major sources of errors. The example of this type of error is shown as the question (2) in Table 5. This phenomenon is observed more often in cross-lingual settings. The prediction of end index can be improved by providing the predicted start index information to the network before making the prediction of end index.

(5) The network could not provide an answer where the reasoning across multiple sentences is required. We also observe the similar behavior, signifying that the network fails to provide the correct answer, where the answer and the headwords (query) in the question are far apart (2 to 3 sentences away). Example:
Q: *The climate of Greece in the Northwest is known as what?*
Snippet: *The mountainous areas of Northwestern Greece -LRB- parts of Epirus, Central Greece, Thessaly, Western Macedonia -RRB- as well as in the mountainous central parts of Peloponnese – including parts of the regional units of Achaea, Arcadia and Laconia – feature an Alpine climate with heavy snowfalls. . . . . Snowfalls occur every year in the mountains and northern areas, and brief snowfalls are not unknown even in low-lying southern areas, such as Athens.*
Gold Answer: Alpine climate
Predicted Answer: Western Macedonia

In this example, the model has to perform the reasoning across multiple sentences to conclude the correct answer. This type of error can be addressed by the multi-step of reasoning similar to the work of Das et al. [2019].

(6) One of the limitations of the network is that it does not correctly identify the answer of short descriptive questions started with “*why*’ or ‘*how.*” In these types of errors, the network could not predict the correct answer indices. It is because the network has to predict the correct phrase that is not limited to only a noun, verb, or adjective phrase. The prediction of the complex phrase is difficult as compared to the prediction of the named entities. Example:
Q: *Why did they miss that competition?,*
Snippet: *It is very rare for top clubs to miss the competition, although it can happen in exceptional circumstances. Defending holders Manchester United did not enter the 1999 – 2000 FA Cup, as they were already in the inaugural Club World Championship, with the club stating that entering both tournaments would overload their fixture schedule and make it more difficult to defend their Champions League and Premiership titles. The club claimed that they did not want to devalue the FA Cup by fielding a weaker side. The move benefited United as they received a two-week break and won the 1999 – 2000 league title by an 18-point margin, although they did not progress past the group stage of the Club World Championship . . .*
Gold Answer: The club claimed that they did not want to devalue the FA Cup by fielding a weaker side.
Predicted Answer: their handling of the situation

(7) The network also suffers to find the correct answer in cross-lingual setup ($Q_E - S_H$) where the answer words are not named entity and consist of a descriptive answer. Example:
Q: कई चीनी सैनिकों की एक बड़ी चिंता क्या थी?
Trans: *What was a great concern of many Chinese troops?,*
Snippet: *.. In late April Peng Dehuai sent his deputy, Hong Xuezhi, to brief Zhou Enlai in Beijing. What Chinese soldiers feared, Hong said, was not the enemy, but that they had nothing to eat, no bullets to shoot, and no trucks to transport them to the rear when they were wounded. Zhou attempted to respond to the PVA 's logistical concerns by increasing Chinese production and improving methods of supply, but these efforts were never completely sufficient. At the same time, large-scale air defense training programs were carried out, and the Chinese Air Force began to participate in the war from September 1951 onward.*
Gold Answer: they had nothing to eat
Predicted Answer: supply

6 CONCLUSION

In this article, we have proposed a unified deep neural network technique for multilingual question answering. The proposed model is a generic framework with the flexibility of being adaptable to any number of languages. To provide the input snippet (if not available) to the proposed network, we introduce an effective language-independent snippet generation algorithm. Our snippet generation algorithm exploits the lexico-semantic similarity between the sentences. The soft alignment of the question words from the English and Hindi languages has been used to learn the shared representation of the question. The learned shared representation of question and attention-based snippet representation are passed as an input to the answer extraction layer of the network, which extracts the answer span from the snippet.

We achieve state-of-the-art performance on the multilingual benchmark QA dataset. Evaluation shows that our proposed model attains 39.44 Exact Match (EM) and 44.97 F1 values. In the future, we will work towards addressing the specific concerns to improve the system performance. We would also like to handle the descriptive and multi-step reasoning questions under the multilingual environment.

REFERENCES

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *Proceedings of the International Conference on Machine Learning*. 1247–1255.

Sarath Chandar A. P., Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 1853–1861.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Ling.* 5 (2017), 135–146. DOI: https://doi.org/10.1162/tacl_a_00051.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. Association for Computational Linguistics, 615–620. Retrieved from: http://www.aclweb.org/anthology/D14-1067.

Mitchell Bowden, Marian Olteanu, Pasin Suriyentrakorn, Thomas d'Silva, and Dan Moldovan. 2007. Multilingual question answering through intermediate translation: LCC's PowerAnswer at QA@ CLEF 2007. In *Proceedings of the Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 273–283.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of the NIPS Workshop on Deep Learning*.

Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. 2016. Together we stand: Siamese networks for similar question retrieval. In *Proceedings of the 54th Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 378–387. DOI: https://doi.org/10.18653/v1/P16-1036

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *Proceedings of the International Conference on Learning Representations*. Retrieved from: https://openreview.net/forum?id=HkfPSh05K7.

Asif Ekbal, Deepak Gupta, Surabhi Kumari, and Pushpak Bhattacharyya. 2018. MMQA: A multi-domain multi-lingual question-answering framework for English and Hindi. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)* (7-12). European Language Resources Association (ELRA).

Pamela Forner, Anselmo Peñas, Eneko Agirre, Iñaki Alegria, Corina Forăscu, Nicolas Moreau, Petya Osenova, Prokopis Prokopidis, Paulo Rocha, Bogdan Sacaleanu et al. 2008. Overview of the CLEF 2008 multilingual question answering track. In *Proceedings of the Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 262–295.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 29, 5 (2001), 1189–1232.

María Dolores García Santiago and María Dolores Olvera-Lobo. 2010. Automatic web translators as part of a multilingual question-answering (QA) system: Translation of questions. *Transl. J.* 14, 1 (2010).

Danilo Giampiccolo, Pamela Forner, Jesús Herrera, Anselmo Peñas, Christelle Ayache, Corina Forascu, Valentin Jijkoun, Petya Osenova, Paulo Rocha, Bogdan Sacaleanu et al. 2007. Overview of the CLEF 2007 multilingual question answering track. In *Proceedings of the Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 200–236.

Deepak Gupta, Pabitra Lenka, Asif Ekbal, and Pushpak Bhattacharyya. 2018a. Uncovering code-mixed challenges: A framework for linguistically driven question generation and neural-based question answering. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 119–130. DOI: https://doi.org/10.18653/v1/K18-1012

Deepak Gupta, Rajkumar Pujari, Asif Ekbal, Pushpak Bhattacharyya, Anutosh Maitra, Tom Jain, and Shubhashis Sengupta. 2018. Can taxonomy help? Improving semantic question matching using question taxonomy. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, , 499–513. Retrieved from: https://www.aclweb.org/anthology/C18-1042.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.* 16, 12 (2004), 2639–2664.

Xiaodong He and David Golub. 2016. Character-level question answering with attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1598–1607. Retrieved from: https://aclweb.org/anthology/D16-1166.

Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. Retrieved from: http://www.ijcai.org/proceedings/2018/.

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. 2017. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL'17)*. 226–237.

Praveen Kumar, Shrikant Kashyap, Ankush Mittal, and Sumit Gupta. 2005. A Hindi question answering system for E-learning documents. In *Proceedings of the 3rd International Conference on Intelligent Sensing and Information Processing (ICISIP'05)*. IEEE, 80–85.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics, (COLING'02)*. Association for Computational Linguistics, 1–7.

Anutosh Maitra, Shubhashis Sengupta, Abhisek Mukhopadhyay, Deepak Gupta, Rajkumar Pujari, Pushpak Bhattacharya, Asif Ekbal, and Tom Geo Jain. 2018. Semantic question matching in data constrained environment. In *Text, Speech, and Dialogue*, Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala (Eds.). Springer International Publishing, Cham, 267–276.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Meeting of the Association for Computational Linguistics: System Demonstrations*. 55–60.

Bernardo Magnini Matteo, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2001. Multilingual question/answering: The DIOGENE system. In *Proceedings of the 10th Text Retrieval Conference*.

Donald Metzler and W. Bruce Croft. 2007. Linear feature-based models for information retrieval. *Inf. Retr.* 10, 3 (June 2007), 257–274. DOI: https://doi.org/10.1007/s10791-006-9019-z

María-Dolores Olvera-Lobo and Juncal Gutiérrez-Artacho. 2011. *Multilingual Question-Answering System in Biomedical Domain on the Web: An Evaluation*. Springer Berlin, 83–88. DOI: https://doi.org/10.1007/978-3-642-23708-9_10

Jahna Otterbacher, Gunes Erkan, and Dragomir R. Radev. 2009. Biased LexRank: Passage retrieval using random walks with question-based priors. *Inform. Proc. Manag.* 45, 1 (2009), 42–54.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 311–318.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2383–2392. DOI: https://doi.org/10.18653/v1/D16-1264

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of the International Conference on Learning Representations (ICLR'16)*.

Shriya Sahu, Nandkishor Vasnik, and Devshri Roy. 2012. Prashnottar: A Hindi question answering system. *Int. J. Comput. Sci. Inform. Technol.* 4, 2 (2012), 149.

Satoshi Sekine and Ralph Grishman. 2003. Hindi-English cross-lingual question-answering system. *ACM Tran. Asian Lang. Inform. Proc.* 2, 3 (2003), 181–192.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations*.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations, and the inverted softmax. In *Proceedings of the International Conference on Learning Representations*.

Daniil Sorokin and Iryna Gurevych. 2017. End-to-end representation learning for question answering with weak supervision. In *Proceedings of the Semantic Web Evaluation Challenge*. Springer, 70–83.

Shalini Stalin, Rajeev Pandey, and Raju Barskar. 2012. Web based application for Hindi question answering system. *Int. J. Electron. Comput. Sci. Eng.* 2, 1 (2012), 72–78.

Sai Praneeth Suggu, Kushwanth Naga Goutham, Manoj K. Chinnakotla, and Manish Shrivastava. 2016. Hand in glove: Deep feature fusion network architectures for answer quality prediction in community question answering. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, 1429–1440. Retrieved from: https://www.aclweb.org/anthology/C16-1135.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112. Retrieved from: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-Net: From answer extraction to answer synthesis for machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'18).*

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. Retrieved from: *arXiv preprint arXiv:1511.04108* (2015).

Ferhan Ture and Oliver Jojic. 2017. No need to pay attention: Simple recurrent neural networks work! In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2866–2872. Retrieved from: https://www.aclweb.org/anthology/D17-1307.

Kateryna Tymoshenko and Alessandro Moschitti. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1451–1460.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2692–2700. Retrieved from: http://papers.nips.cc/paper/5866-pointer-networks.pdf.

Shuohang Wang and Jing Jiang. 2017. Machine comprehension using match-LSTM and answer pointer. In *Proceedings of the International Conference on Learning Representations (ICLR'17).*

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 189–198.

Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Inform. Retr.* 13, 3 (2010), 254–270.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016a. aNMM: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 287–296.

Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W. Bruce Croft, Jiafeng Guo, and Falk Scholer. 2016b. Beyond factoid QA: Effective methods for non-factoid answer sentence retrieval. In *Proceedings of the European Conference on Information Retrieval*. Springer, 115–128.

Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, and Yi Chang. 2016. Ranking relevance in Yahoo search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, 323–332. DOI: https://doi.org/10.1145/2939672.2939677.

Adams Wei Yu, David Dohan, Thang Luong, Rui Zhao, Kai Chen, and Quoc Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*. Retrieved from: https://openreview.net/pdf?id=B14TlG-RW.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. Retrieved from: *arXiv preprint arXiv:1212.5701* (2012).

Received July 2018; revised April 2019; accepted August 2019