Deep Learning Approach for Machine Reading Comprehension in Vietnamese

Viet-Sang Nguyen, Ngoc-Quy Tran
Faculty of Computer Sciences and Engineering
Ho Chi Minh University of Technology
Email: {nvietsang,motmaytinh}@gmail.com

(Thesis - 06/2019)

***Abstract*—In this paper, we introduce a modern approach which uses Deep Learning techniques for the problem of Question Answering (QA) in Vietnamese. The goal of this QA problem is to find the answer in a passage for a question about a certain fact existing in that passage. To do so, we utilise the ideas from some famous models which successfully worked in English. We also build a dataset with about 5000 samples in order to support the training phase. Our model achieves 61.0% and 76.6% corresponding to the two metrics, Exatch Match (EM) and F1-score. To the best of our knowledge, this is the first attempt in terms of applying Deep Learning techniques to QA problem in such a language as Vietnamese. Our results promise to open a new approach for problems of machine reading comprehension in Vietnamese.**

I. Introduction

Machine Comprehension (MC) is one of the most important problems in Natural Language Processing. Without a doubt, MC has brought us a large number of benefits through the ways we have applied it in reality. Nowadays, in a very busy world, people always expect all the responses are not only as quick as possible but also accurate at the highest level. We also know that it is really hard to satisfy both of these things at the same time. For example, in medicine, doctors can save much time thanks to applying MC to search for any expertise. Similarly, lawyers only take a short time to look for a rule. With MC, people using search engines get answers directly instead of a list of links to different sites which needed a moment to read.

Along with the development of Deep Learning in recent decades, a variety of models has been shown up to fiercely compete to achieve the best performance [1], [2]. In English, there exists a great number of datasets which support this challenge. In Vietnamese, there have been popular researches concerning to QA problem. However, those ones mainly are based on classical approach such as applying statistical models or ontology methods [3] [4]. Our approach, in an arbitrary manner, uses Deep Learning which is the most modern technique at this time, as the backbone to build a QA model and examine the effectiveness when applying to a certain language like Vietnamese.

In this paper, we introduce a new approach using Deep Learning technique for QA problem in Vietnamese. Given a context (passage) and a question which is posed about a certain fact in that context, our end-to-end model aims to find the answer in the context for that question. Following some popular models [1] [2], we use attention mechanisms as an important part in the model to obtain the awareness of the question in the passage. Besides, we also build a Vietnamese QA dataset consisting of nearly 5000 samples which is used for training and testing purposes. Questions in this dataset are posed on over 2000 Vietnamese Wikipedia paragraphs. Figure 1 shows some pairs of question and answer in the dataset. For evaluation, we use Exact Match (EM) and F1-score [5] as the metrics to measure the accuracy. Our end-to-end model

Context: Axit nitric loãng có thể cô đặc đến 68% với một hỗn hợp azeotropic 32% nước. Việc cô đặc hơn được thực hiện bằng cách chưng cất với axit sulfuric với vai trò là chất khử nước. Trong quy mô phòng thí nghiệm, cách chưng cất như thế phải được tiến hành bằng dụng cụ thủy tinh với áp suất thấp để tránh phân hủy axit này. Việc sản xuất axit nitric được thực hiện bằng công nghệ Ostwald do Wilhelm Ostwald phát minh.
(Dilute nitric acid can be concentrated up to 68 % with an azeotropic mixture of 32 % water. The more concentrated is made by distillation with sulfuric acid as a dehydrating agent. On a laboratory scale, such distillation must be carried out by means of glassware at low pressure to avoid decomposition of this acid. The production of nitric acid is made by the Ostwald technology invented by Wilhelm Ostwald.)

Question: Ai là người đã sáng chế ra công nghệ sản xuất axit nitric?
(Who invented the production technology of nitric acid?)
Answer: Wilhelm Ostwald

Question: Chất khử nước được dùng trong quá trình chưng cất axit nitric là gì?
(What is the dehydrating agent used in nitric acid distillation?)
Answer: axit sulfuric
(sulfuric acid)

Figure 1: A sample in Vietnamese QA dataset. The answer is a continuous span in the context.

adapts this dataset with promising results which are 61.0% in EM and 76.6% in F1-score. These results suggest that there is plenty of room for advancement in modeling and learning when we apply Deep Learning to solve the QA problem in such a language as Vietnamese.

This paper is organised as follows. Section II presents related work. Section III analyzes the architecture of our model. Section IV presents the analysis of the dataset. Experiments and main results are presented in Section V. Finally, Section VI concludes our work.

II. Related Work

Machine Comprehension and Question-Answering have played an important role in the field of Natural Language Processing. In English, there are plenty of datasets ([5], [6], [7], etc.) which highly support for research. Along with the richness of datasets, a large number of end-to-end models based on neural networks ([1], [2], [8]) have been proposed and achieved excellent performances. In general, theses models utilised popular techniques in Deep Learning such as Recurrent Neural Networks (RNNs), Convolution Neural Networks (CNNs), Long Short Term Memory architectures (LSTM). Besides, they also used attention mechanisms [9] as the core of model in order to embed the question into the context.

In the meantime, QA problem in Vietnamese has still not got impressive results yet. There exists some traditional approaches for Vietnamese QA. Vu Mai Tran et al. proposed a Vietnamese QA system by combining SnowBall system and semantic relation extraction using search engines [10]. The performance was promising with 89.7% precision and 91.4% ability to give the answers, however, the test was only experimented on traveling domain. Mai-Vu Tran et al. introduced a model of Vietnamese person named entity QA system [4], which used a method of analytical question by using CRF machine learning algorithm together with two strategies of answering automatically including indexed sentences database-based and Google search engine-based. If answers were not found in database, they would be pushed into Google search engine. Their results were 74.63% precision and 87.9% ability to give the answer. Dai Quoc Nguyen et al. proposed a Vietnamese QA system based on ontology [3] which allows user to query in natural language. This system consists of a question analysis module and an answer retrieval module. According to the authors, these two modules achieved an accuracy of 95% and 70% respectively. Dang-Tuan Nguyen et al. built a searching system aiming to search courses on the Vietnam OpenCourseWare Program. This system analyzes questions based on a set of defined syntax rules and create a query in SPAEQL to retrieve data on ontology-based.

Different from these current methods, we introduce a Deep Learning approach for QA problem in Vietnamese. In this approach, state-of-the-art techniques, which have successfully been applied in English models, are utilised in the stage of building our model. Besides, there is no public QA dataset for Vietnamese at this time. Hence, for helping to train the model, we build a Vietnamese QA dataset.

III. Model

First of all, we define the task of reading comprehension as follows. Given a context (passage) C consisting of n words, $\mathsf{C} = c_1, c_2, ..., c_n$ and and a question Q consisting of m words, $\mathsf{Q} = q_1, q_2, ..., q_m$ relating to a certain fact in the context. The goal is to derive a continuous span $\mathsf{A} = c_i, c_{i+1}, ..., c_{i+k}$ ($1 \leq i \leq k \leq n$) in the context C which represents the predicted answer for the question Q.

Figure 2 shows the architecture of the our model. There are totally 6 layers in this architecture, including Character Embedding Layer, Word Embedding Layer, Question and Context Encoder Layer, Attention Layer, Model Encoder Layer and Output Layer.

1) **Character Embedding Layer:** This layer aims to map each word to a vector using character-level representations. Words in C and Q are firstly converted to character-level embeddings. The number of character in a word is always fixed to be 16 by expanding or reducing. Regarding the expansion, we use the token `<NULL>` initialized with vector 0 to denote the expanded characters. Each character is represented by a trainable vector of dimension 64. We follow [11] to use Convolutional Neural Networks (CNN) in order to obtain the representations of these vectors. A word is now represented by a 2D matrix 16×64 considered as an input to the CNN. We then max-pool the output of the CNN over its width to obtain the character-level representation for that word.
2) **Word Embedding Layer:** This layer aims to map each word to a vector using the pre-trained model. Each word is represented by a vector of dimension 100. These vectors are fixed during training. Regarding to the words which do not have pre-trained vectors (out-of-vocab), we use the token `<OOV>` initialized with vector 0 to denote them.

   For each word, the two vectors of character-level embedding and word-level embedding are then concatenated. As the idea of [2], we pass this representation to a two-layer Highway Network [12].
3) **Question and Context Encoder Layer:** This layer aims to refine the representations of words in C and Q by the effect of surrounding words. The result of the previous concatenation is the input of this layer which is a vector of dimension $d = 164$ for an individual word. Similar to [1], we utilize encoder blocks to encode the question Q and the context C separately. In a encoder block, there are 4 convolution layers, a self-attention layer and a feed-forward layer. Following [1], we use depthwise separable convolutions [13] [14] in order to be efficient in term of memory and multi-head attention mechanism [9] in the self-attention layer. In this phase, each word is mapped to a vector of dimension $d = 128$ by the convolutions and this is also the output of this layer.

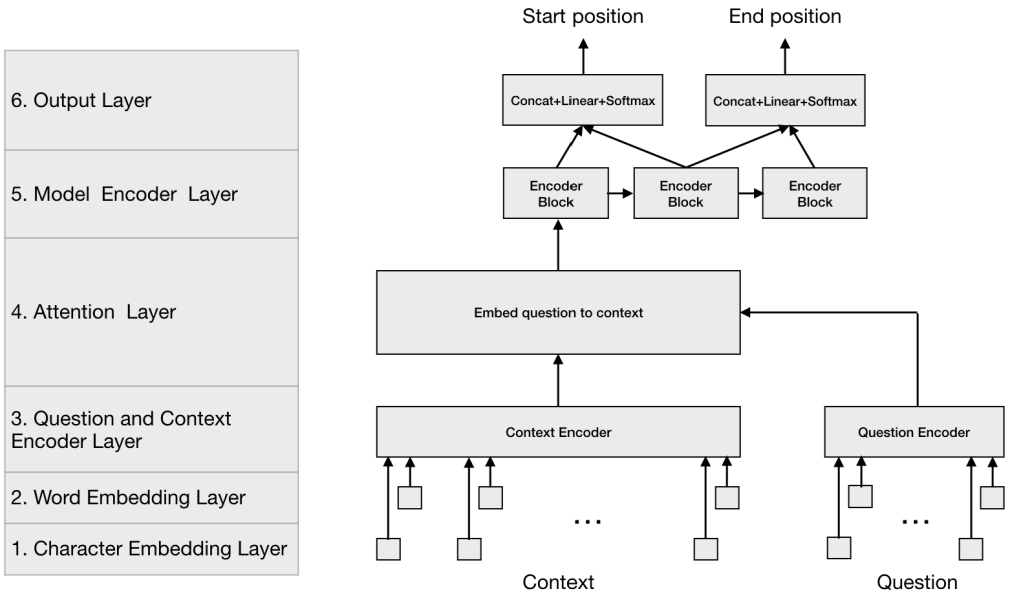


Figure 2: An Overview of Model Architecture

4) **Attention Layer:** This layer aims to incorporate the information in question into the context. This technique is as a standard in many modern models of question-answering problem [15] [16]. The attention is the combination of two directions as proposed in [2] including context-to-query attention (C2Q attention) and query-to-context (Q2C attention). These two directions of attention are both based on a similarity matrix $S \in \mathbb{R}^{n \times m}$, where S_{ij} is the similarity between the i-th word of C and the j-th word of Q. Notice that $\mathsf{C} \in \mathbb{R}^{d \times n}$ and $\mathsf{Q} \in \mathbb{R}^{d \times m}$ are the outputs of the previous layer.

$$S_{ij} = W_S[q, c, q \cdot c]$$

where W_S is a trainable parameter, $[,]$ is the concatenation, $\cdot$ is the element-wise computation, c and q are i-th column in C and j-th column in Q, respectively.

- C2Q attention: This creates a new representation of the context $\mathsf{C}' \in \mathbb{R}^{d \times n}$ where the most relevant words in Q to C are significantly embedded. For each row $S_{i:}$ in S, $\overline{S_{i:}} = \text{softmax}(S_{i:})$. Then i-th column in C' is computed by $\mathsf{C}'_{:i} = \sum_k \overline{S_{ik}} \mathsf{Q}_{:k}$.
- Q2C attention: This also creates a new representation of the context $\mathsf{C}'' \in \mathbb{R}^{d \times n}$, where words in Q having the highest similarity to each word in C are highlighted. We firstly compute $s = \text{softmax}(\max_{row}(S)$, where $\max_{row}(S)$ is to select the maximum similarity of each row in S, $s \in \mathbb{R}^n$. Then, i-th column in C'' is computed by $\mathsf{C}''_{:i} = s_i C_{:i}$, $1 \leq i \leq n$.

Similar to [2], the output in the end of this layer $M \in \mathbb{R}^{4d \times n}$ is the concatenation of C, C' and C'' as follows.

$$M_{:i} = [\mathsf{C}_{:i}; \mathsf{C}_{:i} \cdot \mathsf{C}'_{:i}; \mathsf{C}_{:i} \cdot \mathsf{C}''_{:i}]$$

5) **Model Encoder Layer:** This layer aims to encode the awareness of the question into the context. Following the architecture of QANet [1], we apply a stack of 3 encoder blocks as the Question and Context Encoder Layer. We denote that $M_1, M_2, M_3 \in \mathbb{R}^{d \times n}$ are the outputs of these 3 encoder blocks, respectively. Notice that the input of this layer is M.

6) **Output Layer:** This layer aims to compute the probabilities of each word in the context in terms of considering them as the starting position and the ending position of the predicted answer. We adopt the computations of [1] to derive these probabilities as follows.

$$p^1 = \text{softmax}(W_1[M_0; M_1])$$

$$p^2 = \text{softmax}(W_2[M_0; M_2])$$

where W_1 and W_2 are trainable parameters.
Denote that i-th sample (among N samples in dataset) has the labels of starting position y_i^1 and ending position y_i^2. We desire that probabilities of these two positions are maximized. Therefore, the loss funtion is defined by

$$L(\theta) = -\frac{1}{N} \sum_i^N \log p^1_{y_i^1} + \log p^2_{y_i^2}$$

where θ is the set of all trainable parameters.

IV. DATASET

To the best of our knowledge, at the time of writing, there is no public QA dataset for Vietnamese. Therefore, it is necessary to collect data in order to feed the model.

A. Data Collection

We use Project Nayuki's Wikipedia's internal to rank a large number of articles according to their popularity and hence crawl the top 10000 high-quality articles from Vietnamese Wikipedia. The idea of PageRanks[1] is that the more important (more high-quality) an article is, the larger number of references from other articles it has. We then randomly sample 238 articles and partition to paragraphs which contains less than 500 characters. These articles are cleaned by removing all of images, tables, links and symbols so as to be more informative. We finally obtain 2123 paragraphs which covers a wide range of topics such as music, sport, physics, geography, mathematical, society, and so on.

Next, we launch the procedure of annotation. At first, we guide all of our annotators about rules which they must follow to be sure that every sample of data is good enough to serve the QA problem. These rules are: (1) Question must relate to a fact in context and answer is a continuous span in context; (2) Grammar of question must be correct; (3) Annotators should use their own words to pose questions and avoid copying phrases in passages; (4) Annotators should use many types of questions to reduce biases; (5) Annotators are encouraged to pose hard questions which require information from different positions in paragraph to answer. We also prepare examples of right and wrong questions to help our annotators deeply understand. Each paragraph is randomly assigned to one of our annotators. Their duty is to read through and to completely understand the content. Then, they have to pose at least 3 questions per paragraph. After posing a question, annotators simply highlight the span corresponding to its answer. When having more than one candidates for the answer, the person who poses the question should make it clearer and more detail hence exactly one possible answer.

B. Data Analysis

To understand the properties and difficulty of the dataset in terms of answer types. Answers are categorized to one among 11 categories as shown in the table I. We use VnCoreNLP toolkit [17] to do this task. At first, we separate all the answers to two types: dates and others. Secondly, we determine whether the answers which are not dates can be considered as numeric or not. Then, non-numeric answers are continued categorizing to noun phrases, verb phrases, adjective phrases by using POS tag included in the toolkit. The proper noun phrases are futher split into three following groups: person, location and other entities. In the table I, we can see the diversity of answers. X% of data are made up by three different types of nouns, 10% data are made up by dates and numbers, adjective phrases and verb phrases account for 20% of the total data, and the rest (10%) are made up by clauses and other types.

[1]https://www.nayuki.io/page/computing-wikipedias-internal-pageranks

TABLE I: Categories of answers in dataset

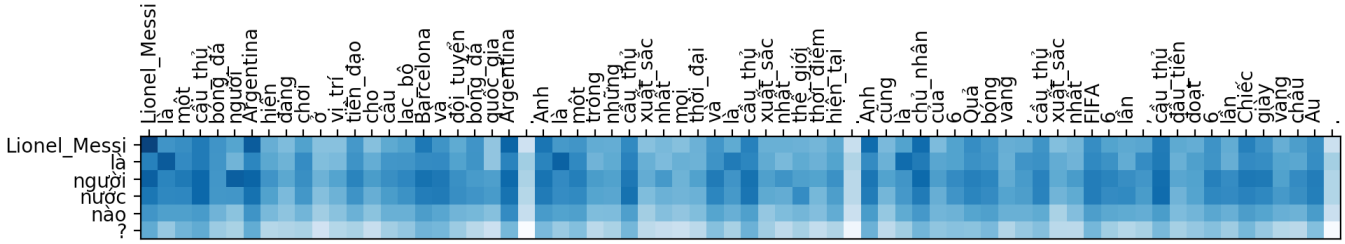
Answer type	**Ratio (%)**	**Examples**
Date & Time	9.22	ngày 3 tháng 2 năm 1930, thế kỷ 16 *(3 February 1930, 16th century)*
Person	9.95	Arthur Cayley, Đề Thám, Hoàng đế Napoléon III *(Arthur Cayley, De Tham, the King Napoleon III)*
Location	13.54	châu Âu, Hy Lạp, sông Mississippi *(European, Greece, Mississippi River)*
Noun Phrase	11.84	đại số, quân đội, bộ trưởng *(algebra, army, minister)*
Adjective Phrase	1.84	lớn hơn, mặn hơn, ngắn và dốc *(greater, more salty, short and sloping)*
Verb Phrase	1.77	làm việc qua đêm *(working overnight)*
Clause	9.78	các chiến thuyền thon dài có cánh buồm và ít nhất 24 mái chèo *(slender warships with sails and at least 24 paddles)*
Other Entity	11.48	Liên minh châu Âu, câu lạc bộ Barcelona *(European Union, Barcelona Football Club)*
Numeric	9.2	6000
Numeric with Units	2.94	100 km, 300 feet, 33 kg
Other	18.44	$P \subseteq NP \subseteq PP \subseteq PSPACE$

V. EXPERIMENT

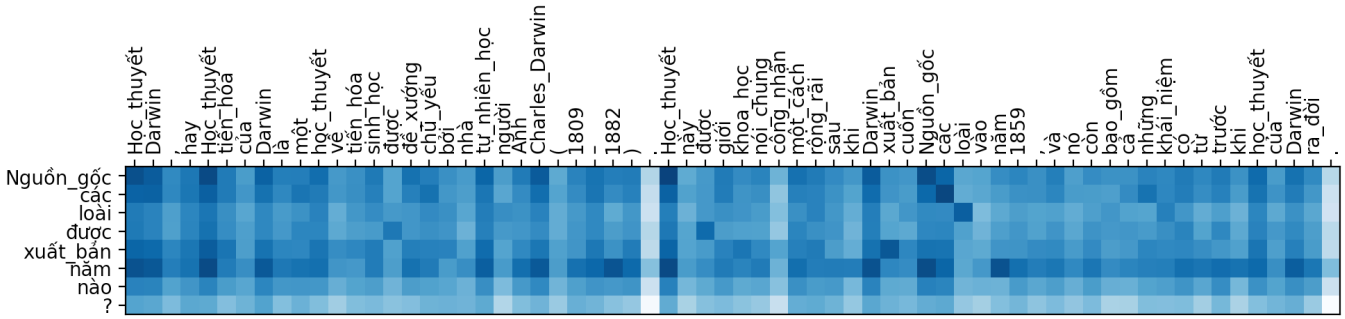
A. Implementation Details

In the stage of processing data, we use Pyvi[2] to tokenize passages and questions. Tokenizing in Vietnamese is significantly different from the one in English. A Vietnamese token can be a single word, e.g. đứng (stand), cây (tree), ngủ (sleep). However, majority of Vietnamese involves compound words which are usually combinations of two individual words, e.g. đường sắt (railway), cà chua (tomato), sân bay (airport). Lengths of context and question are fixed to be 200 tokens and 30 tokens, respectively. Those which have more or less tokens than these numbers are discarded or padded. For word embedding layer, we use the pretrained word vector of Kyubyong's project[3] in which each Vietnamese token is represented by a 100-D vector. We use Gensim [18] to read vectors from the results of Kyubyong's project, then transform them to Glove's format. [19].

The batch size is 8. The hidden size is set to 96 for all layers. We also apply dropout [20] rate of 0.1 for all layers except 0.05 for character embedding layer. Additionally, we use Adam [21] as the optimizer for the model. The learning rate is set to be 0.01. This model is implemented by using Pytorch [22]. Our experiments are executed on a device of GTX 1080. It takes nearly 2 hours for training.



(a) **Context:** Lionel Messi là một cầu thủ bóng đá người Argentina hiện đang chơi ở vị trí tiền đạo cho câu lạc bộ Barcelona và đội tuyển bóng đá quốc gia Argentina. Anh là một trong những cầu thủ xuất sắc nhất mọi thời đại và là cầu thủ xuất sắc nhất thế giới thời điểm hiện tại. Anh cũng là chủ nhân của 6 Quả bóng vàng, cầu thủ xuất sắc nhất FIFA 6 lần, cầu thủ đầu tiên đoạt 6 lần Chiếc giày vàng châu Âu.
(Lionel Messi is an Argentine footballer who currently plays as a striker for the Barcelona club and the Argentina national football team. He is one of the best players of all time and the best player in the world at the moment. He is also the owner of 6 Golden Balls, the best player in FIFA 6 times, the first player to win 6 times the European Golden Shoe.)
Question: Lionel Messi là người nước nào?
(What is the nationality of Lionel Messi?)
Answer: Argentina



(b) **Context:** Học thuyết Darwin, hay Học thuyết tiến hóa của Darwin là một học thuyết về tiến hóa sinh học được đề xướng chủ yếu bởi nhà tự nhiên học người Anh Charles Darwin (1809–1882). Học thuyết này được giới khoa học nói chung công nhận một cách rộng rãi sau khi Darwin xuất bản cuốn Nguồn gốc các loài vào năm 1859, và nó còn bao gồm cả những khái niệm có từ trước khi học thuyết của Darwin ra đời.
(The Darwinian doctrine, or Darwinian doctrine of evolution, is a doctrine of biological evolution primarily promoted by the English naturalist Charles Darwin (1809–1882). This doctrine was widely accepted by the scientific community after Darwin published the Origin of Species in 1859, and it also included concepts that predated Darwin's doctrine.)
Question: Nguồn gốc các loài được xuất bản năm nào?
(What year is the Origin of Species published?)
Answer: 1859

Hình 3: Examples of correct answers with attention matrices.

TABLE II: EM and F1-score of the model

	Number of samples	EM(%)	F1(%)
Train	4989	-	-
Dev	503	59.8	77.5
Test	487	61.0	76.6

B. Main Results

We use two metrics as SQuAD [5] including Exact Match (EM) and F1-score to evaluate the performance of the model. Considering a pair of prediction and groundtruth after being cleaned, EM gets the value of 1 if these two answers are exact the same and the value of 0, otherwise. F1-score measures portion of overlap between groundtruth and predicted answer. More specifically, we denote that overlapped tokens between them is O, number of tokens in the prediction and groundtruth are P and G, respectively. F1-score is computed as follows.

$$\text{precision} = \frac{O}{P}, \quad \text{recall} = \frac{O}{G}$$

$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

We randomly partition the dataset into a training set (80%), a development set (10%) and a testing set (10%). As we can see in table II, our model achieves 59.8% and 77.5% corresponding to EM and F1 in the development phase. These results in the testing phase are 61.0% and 76.6%, respectively. To the best of our knowledge, these are the results of first attempt in terms of applying Deep Learning to QA problem with Vietnamese. In the model architecture, attention matrix (similarity matrix), which represents the relations between

[2]Available at: https://pypi.org/project/pyvi/
[3]Available at: https://github.com/Kyubyong/wordvectors

TABLE III: Categories of errors

Error type	Ratio	Example
Incorrect answer boundary	60	**Context:** George Peacock là người sáng lập tư duy tiên đề trong số học và đại số. Augustus De Morgan phát kiến ra đại số quan hệ trong cuốn sách Syllabus of a Proposed System of Logic. Josiah Willard Gibbs phát triển đại số của các vectơ trong không gian ba chiều, và Arthur Cayley phát triển đại số của ma trận (đây là một đại số không giao hoán). *(George Peacock was the founder of axiomatic thinking in arithmetic and algebra. Augustus De Morgan discovered the relation algebra in the book Syllabus of a Proposed System of Logic. Josiah Willard Gibbs develops algebra of vectors in three dimensions, and Arthur Cayley develops algebra of matrices (this is a non-commutative algebra).)* **Question:** Đại số của ma trận còn gọi là gì? *(What is the algebra of the matrix?)* **Groundtruth:** đại số không giao hoán *(non-commutative algebra)*, **Prediction:** một đại số không giao hoán *(a non-commutative algebra)*
Syntactic complication	13	**Context:** Đường băng được đánh số theo hướng mà từ đó máy bay sẽ cất cánh hoặc hạ cánh, làm tròn tới 10 và chia cho 10. Ví dụ, "Đường băng Ba Sáu"sẽ có hướng 360 độ (nghĩa là hướng Bắc), "Đường băng Chín"có thể dùng để chỉ đường băng có hướng 94 độ (nghĩa là gần hướng Đông), và "Đường băng Một Bảy"cho hướng 168 độ. *(The runway is numbered in the direction from which the aircraft will take off or land, rounded to 10 and divided by 10. For example, "Three-Six runway" will have a 360-degree orientation (i.e, the North), "Nine runway" can be used to indicate a runway at 94-degree orientation (i.e, near the East), and "One-Seven runway" for 168-degree orientation.)* **Question:** Đường băng có hướng 168 độ được đọc như thế nào? *(How is a runway of 168-degree orientation called?)* **Groundtruth:** Đường băng Một Bảy *(One-Seven runway)*, **Prediction:** gần hướng Đông *(near the East)*
Multi-sentence relations	20	**Context:** Angelo thiết lập những quy tắc cơ bản về tư thế và cách di chuyển mà đến nay vẫn còn chi phối đấu kiếm hiện đại, mặc dù cách tấn công và phòng thủ của ông khác nhiều so với hiện nay. Mặc dù chủ ý của ông là chuẩn bị cho môn sinh cho chiến đấu thực, nhưng ông là huấn luyện viên đấu kiếm đầu tiên nhấn mạnh ích lợi về sức khỏe và thể thao của đấu kiếm hơn là sử dụng làm công cụ giết chóc. *(Angelo established the basic rules of posture and movement that still govern modern fencing to this day, although the ways of attack and defense are much different from today. Although his intention was to prepare students for real combat, he was the first fencing instructor to emphasize the health and sport benefits of fencing rather than using it as a tool of killing.)* **Question:** Ai là người huấn luyện viên đầu tiên coi trọng tác dụng về sức khoẻ và thể thao của môn đấu kiếm hơn sự giết chóc? *(Who was the first coach to value the health and sport effects of fencing more than killing?)* **Groundtruth:** Angelo *(Angelo)*, **Prediction:** làm công cụ giết chóc *(a tool of killing)*
Paraphrase	7	**Context:** Mới đây, Đại học Maryland được xếp hạng 54 toàn quốc trên tạp chí U.S. News and World Report, và hạng 18 trong số những đại học công lập. 31 chương trình giảng dạy được xếp vào Top 10 (bao gồm đại học và sau đại học), và 91 chương trình được xếp vào Top 25. Đại học chuyên xếp hạng các đại học thế giới, Đại học Shanghai Jiao Tong đã xếp Đại học Maryland hạng 37 trên thế giới và hạng 11 trong số những trường công lập của Mỹ. *(Recently, the University of Maryland was ranked 54 nationally in the U.S. magazine News and World Report, and 18 among public universities. 31 programs are ranked in the Top 10 (including undergraduate and graduate), and 91 programs are ranked in the Top 25. Shanghai Jiao Tong University, which specializes in the ranking of world universities, has ranked University attended Maryland at 37th in the world and 11th among American public schools.)* **Question:** Chất lượng của đại học Maryland đứng thứ mấy cả nước? *(What is the rank of the University of Maryland's University among the nation?)* **Groundtruth:** 54, **Prediction:** 10 (bao gồm đại học và sau đại học), và 91 *(10 (including undergraduate and graduate), and 91)*

question and context, plays an important role in a manner of successfully inferring the answer. Figure 3 shows two examples of attention matrices. As we can see, the closer meaning between words are, the higher similarity they have.

C. Error Analysis

We randomly choose 30 cases in which the model gives wrong answers. After analysis, these errors can be categorized to 4 main classes including incorrect boundaries when answering (60%), complications in syntax of the contexts (13%), relations between different sentences in answers (20%) and problems of paraphrase (7%). Table III shows some examples of these errors.

VI. Conclusion

In this paper, we propose a Deep Learning approach for QA problem in Vietnamese. We build an end-to-end model which has an ability to answer a question about a certain fact in a passage. We also build a Vietnamese QA dataset for training and testing. To the best of our knowledge, this is the first attempt to apply Deep Learning in Vietnamese QA problem. Our results promises to open a new direction for QA problem in such a language as Vietnamese. Future work involves enlarging the scale of dataset and enhancing the quality of the tokenizer as well as pretrained word embedding vectors.

References

[1] A. W. Yu, D. Dohan, T. Luong, R. Zhao, K. Chen, and Q. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," 2018. [Online]. Available: https://openreview.net/pdf?id=B14TlG-RW

[2] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *CoRR*, vol. abs/1611.01603, 2016. [Online]. Available: http://arxiv.org/abs/1611.01603

[3] D. Q. Nguyen, D. Q. Nguyen, and S. B. Pham, "A vietnamese question answering system," *2009 International Conference on Knowledge and Systems Engineering*, Oct 2009. [Online]. Available: http://dx.doi.org/10.1109/KSE.2009.42

[4] M.-V. Tran, D.-T. Le, X. T. Tran, and T.-T. Nguyen, "A model of Vietnamese person named entity question answering system," in *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*. Bali, Indonesia: Faculty of Computer Science, Universitas Indonesia, Nov. 2012, pp. 325–332. [Online]. Available: https://www.aclweb.org/anthology/Y12-1035

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics,

Nov. 2016, pp. 2383–2392. [Online]. Available: https://www.aclweb.org/anthology/D16-1264

[6] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," 2017.

[7] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang, "Ms marco: A human generated machine reading comprehension dataset," 2016.

[8] D. Lukovnikov, A. Fischer, and J. Lehmann, "Pretrained transformers for simple question answering over knowledge graphs," 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[10] V. M. Tran, V. D. Nguyen, O. T. Tran, U. T. T. Pham, and T. Q. Ha, "An experimental study of vietnamese question answering system," in *2009 International Conference on Asian Language Processing*, 2009, pp. 152–155.

[11] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: https://www.aclweb.org/anthology/D14-1181

[12] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *CoRR*, vol. abs/1505.00387, 2015. [Online]. Available: http://arxiv.org/abs/1505.00387

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357

[14] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," *CoRR*, vol. abs/1706.03059, 2017. [Online]. Available: http://arxiv.org/abs/1706.03059

[15] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1870–1879. [Online]. Available: https://www.aclweb.org/anthology/P17-1171

[16] D. Weissenborn, G. Wiese, and L. Seiffe, "Making neural QA as simple as possible but not simpler," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 271–280. [Online]. Available: https://www.aclweb.org/anthology/K17-1028

[17] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese natural language processing toolkit," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 56–60. [Online]. Available: https://www.aclweb.org/anthology/N18-5012

[18] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[19] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf