BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 1
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0008

Question Analysis towards a Vietnamese Question Answering System in the Education Domain

Ngo Xuan Bach[1], *Phan Duc Thanh*[1], *Tran Thi Oanh*[2]

[1]*Department of Computer Science, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam*
[2]*VNU International School, Vietnam National University, Hanoi, Vietnam*
E-mails: bachnx@ptit.edu.vn phanducthanh1997@gmail.com oanhtt@isvnu.vn

***Abstract*:** *Building a computer system, which can automatically answer questions in the human language, speech or text, is a long-standing goal of the Artificial Intelligence* (*AI*) *field. Question analysis, the task of extracting important information from the input question, is the first and crucial step towards a question answering system. In this paper, we focus on the task of Vietnamese question analysis in the education domain. Our goal is to extract important information expressed by named entities in an input question, such as university names, campus names, major names, and teacher names. We present several extraction models that utilize the advantages of both traditional statistical methods with handcrafted features and more recent advanced deep neural networks with automatically learned features. Our best model achieves 88.11% in the F_1 score on a corpus consisting of 3,600 Vietnamese questions collected from the fan page of the International School, Vietnam National University, Hanoi.*

***Keywords*:** *Question analysis, question answering, convolutional neural networks, bidirectional long-short term memory, conditional random fields.*

1. Introduction

Question Answering (QA), a subfield of Information Retrieval (IR) and Natural Language Processing (NLP), aims to build computer systems, which can automatically answer questions of users in a natural language. These systems are widely applied in more and more fields such as e-commerce, business, and education. Nowadays, students everywhere carry their mobile phone/laptop with them. It helps students to connect with the world. Therefore, as a trend, universities need to develop their own QA system to foster students' engagement anytime and anywhere. This brings multiple benefits to both students and universities. For students, they can easily get information about a university/college such as degrees, programs, courses, lecturers, campus, admission conditions, and scholarships. For universities, it helps in recruiting new students by facilitating the students in seeking out a

college/university's information; in ensuring constant communication: provide instant for multi-users with 24/7/365 feedback especially in admission periods; and creating a universally accessible website for the university.

There are two main approaches to build a QA system: 1) Information Retrieval (IR) based approach, and 2) knowledge-based approach. An IR-based QA system consists of three steps. First, the question is processed to extract important information (question analysis step). Next, the processed question serves as the input for information retrieval on the Word Wide Web (WWW) or on a collection of documents. Answer candidates are then extracted from the returned documents (answer extraction step). The final answer is selected among the candidates (answer selection step). While an IR-based QA method finds the answer from the WWW or a collection of (plain) documents, a knowledge-based QA method computes the answer using existing knowledge bases in two steps. The first step, question analysis, is similar to the one in an IR-based system. In the next step, a query or formal representation is formed from extracted important information, which is then used to query over existing knowledge bases to retrieve the answer.

Question analysis, the task of extracting important information from the question, is a key step in both IR-based and knowledge-based question answering. Such information will be exploited to extract answer candidates and select the final answer in an IR-based QA system or to form the query or formal representation in a knowledge-based QA system. Without extracted information in the question analysis step, the system could not "understand" the question and, therefore, fails to find the correct answer. A lot of studies have been conducted on question analysis. Most of them fall into one of two categories: 1) question classification or intent detection [9, 12, 17, 18] and 2) Named Entity Recognition (NER) in questions [2, 20]. While question classification determines the type of question or the type of the expected answer, the task of NER aims to extract important information expressed by named entities in the questions.

In this work, we deal with the task of Vietnamese question analysis in the education domain. Given a Vietnamese question. Our goal is to extract named entities in the question, such as university names, campus names, department names, major names, lecturer names, numbers, school years, time, and duration. Table 1 shows examples of questions, named entities in those questions, and their translations in English. The outputs of the task can be exploited to develop an online, web-based or mobile app, QA system. We investigate several methods to deal with the task, including traditional probabilistic graphical models like Conditional Random Fields (CRFs) and more advanced deep neural networks with Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks. Although CRFs can be used to train an accurate recognition model with a quite small annotated dataset, we need a manually designed feature set. Recent advanced deep neural networks have been shown to be powerful models, which can achieve very high performance with automatically learned features from raw data. Neural networks, however, are data hungry. They need to be trained on a quite large dataset, which is challenging for the task in a specific domain. To overcome such challenges, we introduce a recognition models that integrates multiple neural network layers for

learning word and sentence representations, and a CRF layer for inference. By utilizing both automatically learned and manually engineered features, our models outperform competitive baselines, including a CRF model and neural network models that use only automatically learned features.

Table 1. Examples of Vietnamese questions and named entities in the education domain

No	Questions	Entities
1	Học phí **[**ngành kế toán**][**năm nay**]** bao nhiêu ạ? How much is the tuition fee of the **[**Accounting Program**][**this year**]** ?	– ngành kế toán (Accounting Program): a major/program name – năm nay (this year): time
2	**[**Sinh viên năm nhất**]** học ở **[**Nguy Như**]** hay **[**Thanh Xuân**]** ạ? Do**[**freshmen**]** study at **[**Nguy Nhu**]** or **[**Thanh Xuan**]**?	– Sinh viên năm nhất (freshmen): the academic year of students (first year) – Nguy Như (Nguy Nhu): a campus name – Thanh Xuân (Thanh Xuan): a campus name
3	Cho em hỏi số điện thoại của **[**cô Ngân**]** ở **[**phòng đào tạo**]** ạ? Could you please tell me the phone number of **[**Ms.Ngan**]** from the **[**Training Department**]**?	– cô Ngân (Ms. Ngan): the name of a staff – phòng đào tạo (Training Department): a department name
4	Điều kiện để nhận **[**học bổng Yamada**]** là gì ạ? What are the conditions for **[**Yamada Scholarship**]**?	– học bổng Yamada (Yamada scholarship): the name of a scholarship program

Our contributions can be summarized in the following points: 1) we present several models for recognizing named entities in Vietnamese questions, which combine traditional statistical methods and advanced deep neural networks with a rich feature set; 2) we introduce an annotated corpus for the task, consisting of 3,600 Vietnamese questions collected from the online forum of the VNU International School. The dataset will be made available at publication time; and 3) we empirically verify the effectiveness of the proposed models by conducting a series of experiments and analyses on that corpus. Compared to previous studies [2, 5, 15, 21, 24, 25], we focus on the education domain and exploit advanced machine learning techniques, i.e. deep neural networks.

2. Related work

2.1. Question analysis

Prior studies on question analysis can roughly be divided into two classes: 1) question classification and 2) Named Entity Recognition (NER) in questions.

Question Classification. Several approaches have been proposed to classify questions, including rule-based methods [18], statistical learning methods [9], deep neural network methods [12, 17], and transfer learning methods [16]. Madabushi and Lee [18] present a purely rule-based system for question classification which achieves 97.2% accuracy on the TREC 10 dataset [27]. Their system consists of two steps: 1) extracting relevant words from a question by using the question structure; 2) classifying the question based on rules that associate extracted words to concepts. Huang, Thint and Qin [9] describe several statistical models for question

classification. Their models employ support vector machines and maximum entropy models as the learning methods, and utilize a rich linguistic feature set including both syntactic and semantic information. As a pioneer work, Kim [12] introduces a general framework for sentence classification using CNNs. By stacking several convolutional, max-over-time pooling, and fully connected layers, the proposed model achieves impressive results on different sentence classification tasks. Following the work of Kim [12], Ma et al. [17] propose a novel model with group sparse CNNs. Ligozat [16] presents a transfer learning model for question classification. By automatically translating questions and labels from a source language into a target language, the proposed method can build a question classification in the target language without any annotated data.

NER in Questions. NER is a crucial component in most QA systems. Molla, Zaanen and Smith [20] present an NER model for question answering that aims at higher recall. Their model consists of two phases, which uses hand-written regular expressions and gazetteers in the first phase and machine learning techniques in the second phase. Bach et al. [2] describe an empirical study on extracting important information in transportation law questions. Using conditional random fields [13] as the learning method, their model can extract 16 types of information with high precision and recall. Abujabal et al. [1], Costa [4], Sharma et al. [22], Srihari and Li [23] are some examples, among a lot of QA systems that we cannot list, that exploit an NER component. In addition to studies on building QA systems, several works have been conducted to provide benchmark datasets for the NER task in the context of QA [11, 19]. Mendes, Coheur and Lobo [19] introduce nearly 5,500 annotated questions with their named entities to be used as training corpus in machine learning-based NER systems. Kilicoglu et al. [11] describe a corpus of consumer health questions annotated with named entities. The corpus consists of 1548 questions about diseases and drugs, which contains 15 broad categories of biomedical named entities.

2.2. Vietnamese question answering

Several attempts have been made to build Vietnamese QA systems. Tran et al. [24] describe an experimental Vietnamese QA system. By extracting information from the WWW, their system can answer simple questions in the travel domain with high accuracy. Nguyen, Nguyen and Pham [21] present a prototype for an ontology-based Vietnamese QA system. Their system works like a natural language interface to a relational database. Tran et al. [25] introduce another Vietnamese QA system focusing on *Who*, *Whom*, and *Whose* questions, which require an answer as a person name. Tran et al. [26] introduce a learning-based approach for Vietnamese question classification which utilizes two kinds of features bag-of-words and keywords extracted from the Web. Some studies have been conducted to build a Vietnamese QA system in the legal domain [2, 5]. While Duong and Bao-Quoc [5] focus on simple questions about provisions, processes, procedures, and sanctions in law on enterprises, Bach et al. [2] deal with questions about the transportation law. The most recent work on this field is the one of Le-Hong and Bui [15], which proposes an end-to-end factoid QA system for Vietnamese. By combining both statistical

models and ontology-based methods, their system can answer a wide range of questions with promising accuracy.

To the best of our knowledge, this is the first work on machine learning-based Vietnamese question analysis as well as question answering in the education domain.

3. Recognition models

Given a Vietnamese input question represented as a sequence of words $s = w_1 w_2 \ldots w_n$ where n denotes the length (in words) of s, our goal is to extract all the named entities in the question. A named entity is a word or a sequence of consecutive words that provides information about campuses, lecturers, subjects, departments, and so on. Such important information clarifies the question and need to be extracted to answer to the question.

Our task belongs to information extraction, a subfield of natural language processing which aims to extract important information from text. We cast our task as a sequence tagging problem, which assigns a tag to each word in the input sentence to indicate whether the word begins a named entity (tag B), is inside (not at the beginning) a named entity (tag I), or outside all the named entities (tag O). Table 2 shows two examples of tagged sentences in the IOB notation. For example, the tag B-MajorName indicates that the word begins a major name, while the tag I-ScholarName indicates that the word is inside (not at the beginning) a scholarship name.

Table 2. Examples of tagged sentences using the IOB notation

Học_phí/**O** ngành/**B-MajorName** kế_toán/**I-MajorName** năm/**B-Datetime** nay/**I-Datetime** bao_nhiêu/**O** ạ/**O**?/**O** **(**How much is the tuition fee of the Accounting Program this year?)
Điều_kiện/**O** để/**O** nhận/**O** học_bổng/**B-ScholarName** Yamada/**I-ScholarName** là/**O** gì/**O** ạ/**O**?/**O** (What are the conditions for Yamada Scholarship?)

In the following we present our models for solving the above sequence tagging task, including a CRF-based model and more advanced models with deep neural networks. The CRF-based model exploits a traditional but powerful sequence learning method (i.e., conditional random fields) with manually designed features, which can be used as a strong baseline to compare with our neural models.

3.1. CRF-based model

Our baseline model uses Conditional Random Fields (CRFs) [13], which have been shown to be an effective framework for sequence tagging tasks, such as word segmentation, part-of-speech tagging, text chunking, information retrieval, and named entity recognition. Unlike hidden Markov models and maximum entropy Markov models, which are directed graphical models, CRFs are undirected graphical models (as illustrated in Fig. 1). For an input sentence represented as a sequence of words $s = w_1 w_2 \ldots w_n$, CRFs define the conditional probability of a tag sequence t given s as follows:

$$p(t|s,\lambda,\mu) = \frac{1}{Z(s)} \exp\left(\sum_j \lambda_j f_j(t_{i-1}, t_i, s, i) + \sum_k \mu_k g_k(t_i, s, i)\right),$$

where $f_j(t_{i-1}, t_i, s, i)$ is a transition feature function, which is defined on the entire input sequence s and the tags at positions i and $i - 1$; $g_k(t_i, s, i)$ is a state feature function, which is defined on the entire input sequence s and the tag at position i; λ_j and μ_k are model parameters, which are estimated in the training process; $Z(s)$ is a normalization factor.

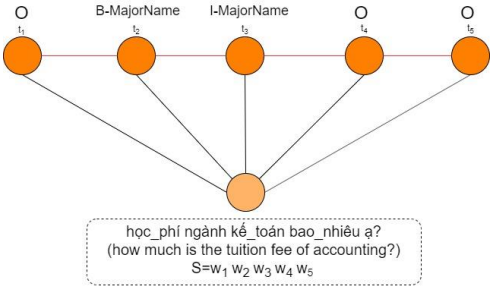

Fig. 1. Recognition model with linear-chain conditional random fields

Our CRF-based model encodes different types of features as follows:

- ***n*-grams.** We extract all position-marked n-grams (unigrams, bigrams, and trigrams) of words in the window of size 5 centered at the current word.
- **POS tags.** We extract *n*-grams of POS tags in a similar way.
- **Capitalization patterns.** We use two features for looking at capitalization patterns (the first letter and all the letters) in the word.
- **Special character.** We use a feature to check whether the word contains a special character (hyphen, punctuation, dash, and so on).
- **Number.** We use a feature to check whether the word is a number.

3.2. Neural recognition model

As illustrated in Fig. 2, our neural network-based model consists of three stages: word representation, sentence representation, and inference.

- **Word representation.** In this stage, the model employs several neural network layers to learn a representation for each word in the input question. The final representation incorporates both automatically learned information at the character and word levels and handcrafted features extracted from the word. We consider two variants of the model; one uses CNNs and the other exploits BiLSTM networks to learn the word representation. The detail of the two variants will be described in the following sections.
- **Sentence Representation.** In this stage, BiLSTM networks are used to modeling the relation between words. Receiving the word representations from the previous stage, the model learns a new representation for each word that incorporates the information of the whole question. Previous studies [3] show that by stacking several BiLSTM layers, we can produce better representations. We, therefore, also

use two BiLSTM layers in this stage. The detail of BiLSTM networks will be presented in the following sections.

- **Inference.** In this stage, the model receives the output of the previous stage and generates a tag (in the IOB notation) at each position of the input question. We consider two variants of the models; one uses the softmax function and the other exploit CRFs. While the softmax function computes a probability distribution on the set of all possible tags at each position of the question independently, CRFs can look at the whole question and utilize the correlation between the current tag and neighboring tags.

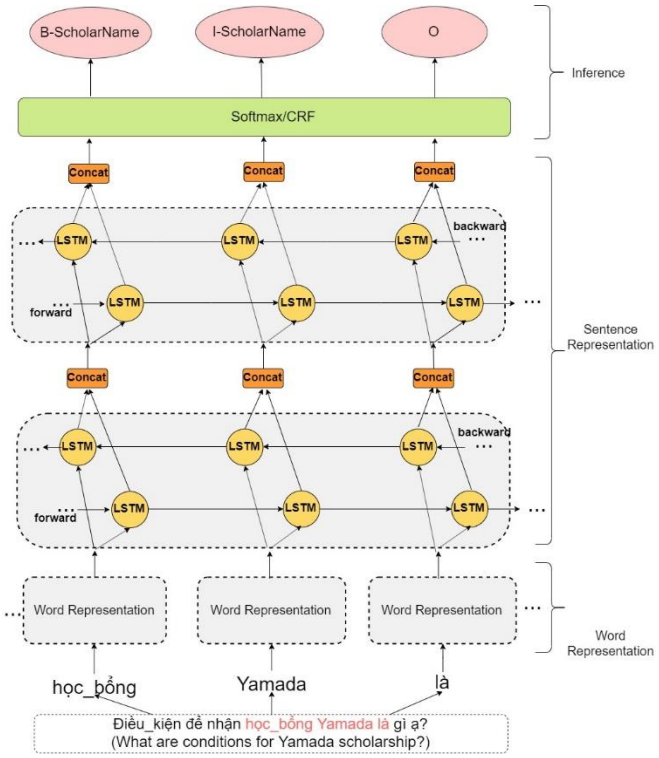

Fig. 2. General architecture of neural recognition models

We now describe our two methods to produce the word representation for each word in the input question. The first method employs CNNs, and the other one uses BiLSTM networks. For notation, we denote vectors with bold lower-case, matrices with bold upper-case, and scalars with italic lower-case.

3.2.1. Word representation using CNNs

As shown in Fig. 3, our word representations employ both handcrafted and automatically learned features.

- **Handcrafted features.** We use the POS tag of the word and multiple features that check whether the word contains special characters, whether the word is a number, and look at capitalization patterns of the word.
- **Automatically learned features.** We use both word embeddings and character embeddings. Convolutional neural networks are then used to extract features from the matrix formed from character embeddings.

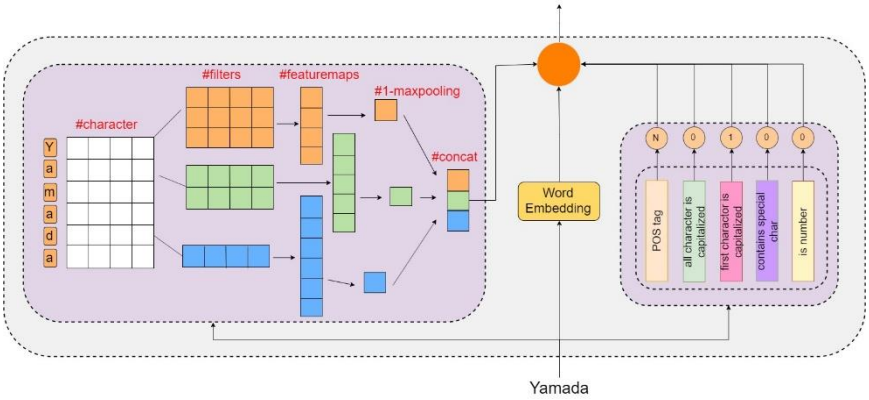

Fig. 3. Word representation using CNNs

The final representation of a word is the concatenation of three components: 1) character representations (the output of the CNNs); 2) the word embedding; 3) the embeddings of handcrafted features. Word embeddings, character embeddings, and the embeddings of handcrafted features are initialized randomly and learned during the training process.

In the following, we give a brief introduction to CNNs and describe how to use them to produce our word representations.

Convolutional neural networks [14] are one of the most popular deep neural network architectures that have been applied successfully to various fields of computer science, including computer vision [10], recommender systems [29], and natural language processing [12]. The main advantage of CNNs is the ability to extract local features or local patterns from data. In this work, we apply CNNs to extract local features from groups of characters or sub-words.

Suppose that we want to learn the representation of a Vietnamese word consisting of a sequence of characters $c_1 c_2 \dots c_m$, where each character c_i is represented by its d-dimensional embedding vector $\mathbf{x}_i$ and m denotes the length (in character) of the word. Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ denotes the embedding matrix, which is formed from the embedding vectors of m characters. We first apply a convolution filter $\mathbf{H} \in \mathbb{R}^{w \times d}$ of height w and width d ($w \leq m$) on $\mathbf{X}$, with stride height of 1. We then apply a tanh operator to generate a feature map $\mathbf{q}$. Specifically, let $\mathbf{X}_i$ be the submatrix consisting of w rows of $\mathbf{X}$ starting at the i-th row, we have

$$\mathbf{q}[i] = \tanh(\langle \mathbf{X}_i, \mathbf{H} \rangle + b),$$

where $\mathbf{q}[i]$ is the i-th element of $\mathbf{q}$, $\langle .,. \rangle$ denotes the Frobenius inner product, tanh is the hyperbolic tangent activation function, and b is a bias.

Finally, we perform max-over-time pooling to generate a feature f that corresponds to the filter $\mathbf{H}$:

$$f = \max_i \mathbf{q}[i].$$

By using h filters $\mathbf{H}_1, \ldots, \mathbf{H}_h$ with different height w, we will generate a feature vector $\mathbf{f} = [f_1, \ldots, f_h]$, which serves as the character representation of our model.

3.2.2. Word representation using BiLSTM networks

As illustrated in Fig. 4, our second method to produce the word representation is similar to the first method presented in the previous section, except that we now use BiLSTM networks to learn the character representation instead of using CNNs.

In the following, we give a brief introduction to BiLSTM networks and explain how to apply them to character embeddings for producing the character representation of the whole word. Note that the process of applying BiLSTM networks to the word representations in the sentence representation stage is similar.

Besides CNNs, Recurrent Neural Networks (RNNs) [6] are one of the most popular and successful deep neural network architectures, which are specifically designed to process sequence data such as natural languages. Long Short-Term Memory (LSTM) networks [8] are a variant of RNNs, which can deal with the long-range dependency problem by using some gates at each position to control the passing of information along the sequence.

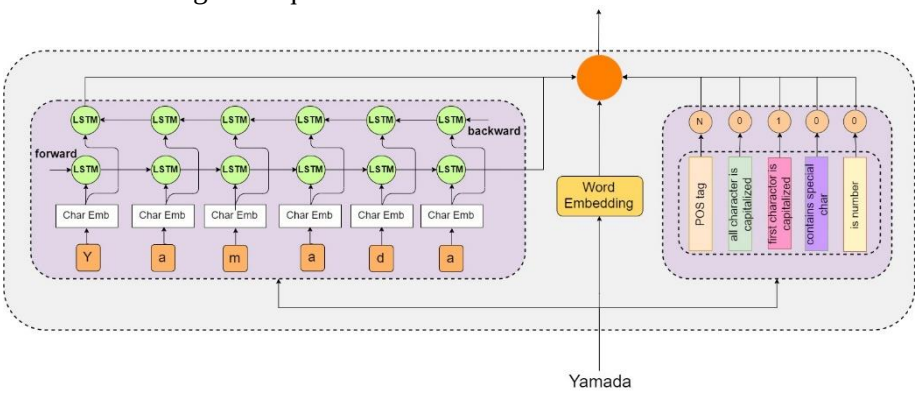

Fig. 4. Word representation using BiLSTM networks

Recall that we want to learn the representation of a word represented by $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$, where $\mathbf{x}_i$ is the character embedding of the i-th character and m denotes the length (in characters) of the word. At each position i, the LSTM network generates an output $\mathbf{y}_i$ based on a hidden state $\mathbf{h}_i$

$$\mathbf{y}_i = \sigma(\mathbf{U}_y \mathbf{h}_i + \mathbf{b}_y),$$

where the hidden state $\mathbf{h}_i$ is updated by several gates, including an input gate $\mathbf{I}_i$, a forget gate $\mathbf{F}_i$, an output gate $\mathbf{O}_i$, and a memory cell $\mathbf{C}_i$ as follows:

$$\mathbf{I}_i = \sigma(\mathbf{U}_\mathrm{I} \mathbf{x}_i + \mathbf{V}_\mathrm{I} \mathbf{h}_{i-1} + \mathbf{b}_\mathrm{I}),$$
$$\mathbf{F}_i = \sigma(\mathbf{U}_\mathrm{F} \mathbf{x}_i + \mathbf{V}_\mathrm{F} \mathbf{h}_{i-1} + \mathbf{b}_\mathrm{F}),$$
$$\mathbf{O}_i = \sigma(\mathbf{U}_\mathrm{O} \mathbf{x}_i + \mathbf{V}_\mathrm{O} \mathbf{h}_{i-1} + \mathbf{b}_\mathrm{O}),$$
$$\mathbf{C}_i = \mathbf{F}_i \odot \mathbf{C}_{i-1} + \mathbf{I}_i \odot \tanh(\mathbf{U}_\mathrm{C} \mathbf{x}_i + \mathbf{V}_\mathrm{C} \mathbf{h}_{i-1} + \mathbf{b}_\mathrm{C}),$$
$$\mathbf{h}_i = \mathbf{O}_i \odot \tanh(\mathbf{C}_i)$$

In the above equations, σ and ⊙ denote the element-wise softmax and multiplication operator functions, respectively; **U**, **V** are weight matrices, **b** are bias vectors, which are learned during the training process.

LSTM networks are used to model sequence data from one direction, usually from left to right. To capture the information from both directions, our model employs Bidirectional LSTM (BiLSTM) networks [7]. The main idea of BiLSTM networks is that it integrates two LSTM networks, one moves from left to right (forward LSTM) and the other one moves in the opposite direction, i.e. from right to left (backward LSTM). Specifically, the hidden state $\mathbf{h}_i$ of the BiLSTM is the concatenation of the hidden states of two LSTMs.

4. Dataset

4.1. Data collection and pre-processing

To build the dataset, we collected questions from the fan page of the International School, Vietnam National University, Hanoi (VNU-IS) in 7 years, from 2012 to 2018. The raw sentences are very noisy. Many of them contain unformal words, slang, abbreviations, foreign language words, grammatical errors, and words without tone marks. Vietnamese words usually contain tone marks such as a, ă, â, à, á, ả, ã, ạ, ằ, ắ, ẳ, ẵ, ặ, ầ, ấ, ẩ, ẫ, ậ. For some reasons (typing speed-up or habit), however, many Vietnamese people do not use tone marks in unformal text, especially on social networks.

We conducted some pre-processing steps as follows:

- **Sentence removal.** We removed a question if all words in the question are non-standard Vietnamese words (foreign language words, abbreviations, without tone marks, or grammatical errors). We also discarded questions which contain less than three words.
- **Word segmentation.** A Vietnamese word consists of one or more syllables separated by white spaces. We used Pyvi (https://pypi.org/project/pyvi/) to segment Vietnamese questions into words.
- **Part-of-speech tagging.** We also used Pyvi to assign a part-of-speech tag to each word in a question.

Finally, we got a set of 3,600 pre-processed Vietnamese questions, which were used to build our dataset.

4.2. Data annotation

We investigated the questions and determined named entity types, which provide important information to answer the questions. Table 3 lists fourteen entity types, which have been chosen and annotated, including university names, campus names, department names, lecturer names, major names, subject names, document names, scholarship names, admission types, major modes, duration, date times, and numbers. Those entity types are also most frequently asked by students.

Table 3. The list of entity types

No	Entity Type	Explanation
1	UniName	The name of a university/school or an expression that refers to a university/school (Vietnam National University; VNU; Our school)
2	CampusName	The name of a campus or an expression that refers to a campus (Xuan Thuy Campus; Campus 1)
3	DeptName	The name of a department or club (Admission Department; Student Volunteer Club)
4	TeacherName	The name of a lecturer or a staff (Ms. Thuy; Mr. To)
5	MajorName	The name of a major/program (Management Information Systems; Business Administration)
6	SubjectName	The name of a subject/course (Algebra; Java Programming; Technical English)
7	DocName	The name of a document (Tuition Fee Reduction Application Form; Enrollment Application Form)
8	ScholarName	The name of a scholarship (Yamada Scholarship; POSCO Scholarship; Encouraging Study Scholarship)
9	AdmissionType	An admission type (National High School Examination; Entrance Examination)
10	MajorMode	The name of a major mode (Regular Program; International Affiliate Program)
11	KYears	The year of students in the university/school (freshman; second-year students; K15 students)
12	Duration	A period of time (a semester; a month; a year)
13	Datetime	A specific date/time (last year; next Sunday; tomorrow)
14	Number	Numbers (1; 2; 2019)

Three annotators were asked to annotate fourteen entity types on the pre-processed questions. Two of them, undergraduate students of computer sciences, annotated data first. Then, the third annotator, an undergraduate student of management information systems who also is the admin of the fan page of the VNU-IS, re-examined and made the final decision on disagreement. To measure the agreement between annotators we used the Kappa coefficient. The Kappa coefficient of our corpus was 0.76, which usually is interpreted as almost excellent agreement.

4.3. Data statistics

Tables 4, 5 show statistical information on our dataset. Totally, we have 3,600 annotated questions with the average length in words is 11.14. On average, each question contains about 1.02 entity with the average length of 3.04 words. The most popular entities include UniName (952), MajorName (733), Datetime (509), MajorMode (241), ScholarName (219), and AdmissionType (200).

Table 4. Statistical information on the dataset

Number of questions	3,600
Average length (in words) of questions	11.14
Average number of entities per question	1.02
Average length (in words) of entities	3.04

Table 5. Statistical information on entity types

No	Entity type	Quantity		No	Entity Type	Quantity
1	UniName	952		8	ScholarName	219
2	CampusName	119		9	AdmissionType	200
3	DeptName	39		10	MajorMode	241
4	TeacherName	38		11	KYears	30
5	MajorName	733		12	Duration	80
6	SubjectName	120		13	Datetime	509
7	DocsName	171		14	Number	197

5. Experiments

5.1. Evaluation methods

We randomly divided the dataset into five folds and conducted 5-fold cross-validation tests. To measure the performance of recognition models, we used precision, recall, and the F_1 score. Let's take the entity type UniName as an example. Precision, recall, and the F_1 score for this entity type can be computed as follows:

$$\text{Precision} = \frac{\#\text{correctly recognized UniName entities}}{\#\text{recognized UniName entities}},$$

$$\text{Recall} = \frac{\#\text{correctly recognized UniName entities}}{\#\text{actual UniName entities}},$$

$$F_1 = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}.$$

5.2. Models to compare

We conducted experiments to compare the performance of the models presented in Table 6 using the method described in Section 5.1. The baseline model uses CRFs with manually designed features. Our purpose is to investigate the task by using a traditional statistical learning model

Table 6. Models to compare

Model	Word layer	Sentence layer	Inference layer
Baseline			CRFs
CNNs-BiLSTM-Softmax	CNNs	BiLSTM	Softmax
CNNs-BiLSTM-CRFs	CNNs	BiLSTM	CRFs
BiLSTM-BiLSTM-Softmax	BiLSTM	BiLSTM	Softmax
BiLSTM-BiLSTM-CRFs	BiLSTM	BiLSTM	CRFs

Note that for each of neural models (CNNs-BiLSTM-Softmax, CNNs-BiLSTM-CRFs, BiLSTM-BiLSTM-Softmax, BiLSTM-BiLSTM-CRFs), we conducted experiments with two variants of the model: 1) using only automatically learned features; 2) using both automatically learned and manually designed features. The

purpose is to investigate the impact of manually designed features on the performance of neural models.

5.3. Model training

We trained the baseline model using CRF^{++}, an open-source CRF toolkit implemented by Taku Kudo (https://github.com/taku910/crfpp). For deep neural networks, we used NCRF^{++}, an implementation of neural sequence labeling models by Yang and Zhang [28]. We set the dimensions of word embeddings and character embeddings to 100 and 30, respectively. All deep neural models were trained using the standard stochastic gradient descent algorithm with batch size of 8. The learning rate was initialized $\eta_0 = 0.015$ and updated on each epoch of training $\eta_t = \frac{\eta_0}{1+\rho*t}$, where $\rho = 0.05$ is the decay rate and t denotes the number of epochs completed.

One problem that usually occurs during the training process of deep neural networks is overfitting. This is a phenomenon in which the network memorizes the training data very well, but could not generalize to unseen samples. In such situations, the network can produce a very small error on the training dataset, but makes a large error on test data. An effective solution for this problem is dropout, a regularization technique by dropping out units of neural networks to prevent complex co-adaptations on training data. In this work, we also applied dropout with the rate of 0.5 to both word representation and sentence representation stages to reduce overfitting.

5.4. Experimental results

5.4.1. CRFs

We first conducted experiments with the baseline model using CRFs. As shown in Table 7, our model achieved good F_1 scores on most entity types. The best entity types include teacher names (92.25%), university/school names (89.07%), date time (87.88%), numbers (86.18%), subject names (85.49%), major names (85.49%), scholarship names (82.94%), and admission types (82.38%). This is reasonable because most of those entity types have a high frequency in the dataset: university/school names (952), major names (733), date time (509), scholarship names (219), admission types (200), and numbers (197). The entity type of teacher names is an interesting case. Although it appears only 38 times in the dataset, we got a very high F_1 score of 92.25%. The reason may be that teacher names contain capital letters on all their syllables and usually start with prefixes such as “Ms.” and “Mr.”. Entity types with the lowest F_1 scores include school years (55.58%), document names (63.14%), and department names (65.84%). Two of them have a very low frequency in the dataset, school years (30) and department names (39). Although document names appear 171 times in the dataset, entities of this type are usually long and complicated, which results in a low F_1 score. On average, our model achieved 88.62%, 79.34%, and 83.72% in precision, recall, and the F_1 score, respectively.

Table 7. Experimental results of the baseline model

No	Entiy type	Precision (%)	Recall (%)	F_1 (%)
1	UniName	90.83	87.37	89.07
2	CampusName	89.55	66.22	76.14
3	DeptName	86.67	53.09	65.84
4	TeacherName	95.56	89.17	92.25
5	MajorName	91.20	80.44	85.49
6	SubjectName	92.67	79.56	85.62
7	DocsName	82.61	51.09	63.14
8	ScholarName	87.07	79.19	82.94
9	AdmissionType	86.44	78.68	82.38
10	MajorMode	77.73	67.73	72.39
11	KYears	76.00	43.81	55.58
12	Duration	84.92	67.70	75.34
13	Datetime	91.51	84.53	87.88
14	Number	82.51	90.19	86.18
Average		**88.62**	**79.34**	**83.72**

5.4.2. Neural models vs. CRFs

Next, we conducted experiments using neural models to compare with the CRF model. Precision, recall, and the F_1 scores (on average of all entity types) of neural extraction models are shown in Table 8. Our first observation is that all the variants of the neural models outperformed the CRF model by a large margin. This shows the power and effectiveness of the neural models with automatically learned features for the task. Our best model using bidirectional LSTM and CRFs with both automatically learned and handcrafted features achieved 88.11% in the F_1 score, which improved 4.39% compared with the baseline model. The next observation is the impact of the handcrafted features. All the variants using both the automatically learned and handcrafted features got better results than the similar ones using only the automatically learned features. The results also confirmed the effectiveness of using CRFs at the inference layer compared with using the softmax function.

Table 8. Experimental results of neural extraction models

Model	Features	Precision (%)	Recall (%)	F_1(%)
CRFs	Handcrafted	88.62	79.34	83.72
CNNs-BiLSTM-Softmax	Automatically learned	86.93	84.70	85.80
	+ Handcrafted	87.13	86.13	86.63
CNNs-BiLSTM-CRFs	Automatically learned	**88.70**	86.40	87.54
	+ Handcrafted	88.33	87.35	87.84
BiLSTM-BiLSTM-Softmax	Automatically learned	85.97	85.15	85.56
	+ Handcrafted	86.47	85.96	86.21
BiLSTM-BiLSTM-CRFs	Automatically learned	87.27	86.72	86.99
	+ Handcrafted	88.20	**88.02**	**88.11**

Table 9 shows experimental results of our best model in detail, which outperformed the baseline model on all types of entities. Especially, the improvements were significant on some difficult types, including school years (20.24%), document names (14.88%), department names (10.74%), major modes (7.23%), and campus names (6.25%).

Table 9. Experimental results of the best model in detail. The last column shows the results of the baseline model with CRFs

No	Entity type	Precision (%)	Recall (%)	F_1 (%)	F_1 (%) (**CRFs**)
1	UniName	91.71	94.03	92.85 (+3.78)	89.07
2	CampusName	84.74	80.17	**82.39 (+6.25)**	**76.14**
3	DeptName	75.81	77.36	**76.58 (+10.74)**	**65.84**
4	TeacherName	93.00	92.50	92.75 (+0.50)	92.25
5	MajorName	89.70	89.65	89.68 (+4.19)	85.49
6	SubjectName	86.25	85.56	85.91 (+0.29)	85.62
7	DocsName	80.94	75.30	**78.02 (+14.88)**	**63.14**
8	ScholarName	87.23	89.70	88.45 (+5.51)	82.94
9	AdmissionType	87.29	78.24	82.52 (+0.14)	82.38
10	MajorMode	80.50	78.76	**79.62 (+7.23)**	**72.39**
11	KYears	88.81	66.14	**75.82 (+20.24)**	**55.58**
12	Duration	77.43	78.31	77.86 (+2.52)	75.34
13	Datetime	88.91	92.16	90.51 (+2.63)	87.88
14	Number	88.43	88.95	88.69 (+2.51)	86.18
	Average	88.20	88.02	88.11 (+4.39)	83.72

5.5. Error analysis

Table 10 shows some examples that our model fails to extract correct entities and our explanations. Common cases include: 1) our model could not recognize abbreviated entities; 2) our model could not recognize all words in long entities; 3) our model included some noisy words in the extracted entities; 4) our model recognized a long entity as two short entities.

Table 10. Some error cases

Gold standards	Predictions	Comments
[IB**]** học ở đâu ạ? where to learn **[**IB**]**?	IB học ở đâu ạ? where to learn IB?	Our model could not recognize the entity because of abbreviation or missing prefix
Bao nhiêu suất học bổng đã tặng cho sinh viên **[**năm ngoái**]** ạ? How many scholarships were given to students **[**last year**]**?	Bao nhiêu suất học bổng đã tặng cho **[**sinh viên năm ngoái**]** ạ? How many scholarships were given to **[**students last year**]**?	Our model recognized some noisy words
Cho em xin **[**mẫu đơn đăng ký học lại**]** với ạ? May I have **[**re-enrollment application form**]**, please?	Cho em xin **[**mẫu đơn đăng ký học**]** lại với ạ? May I have re-**[**enrollment application form**]**, please?	Our model could not recognize all words in a long entity
Điểm chuẩn **[**chương trình liên kết đào tạo do đại học Keuka cấp bằng**]** bao nhiêu ạ? What is the matriculation score of **[**the joint training program awarded by Keuka University**]**?	Điểm chuẩn **[**chương trình liên kết đào tạo**]** do **[**đại học Keuka**]** cấp bằng bao nhiêu ạ? What is the matriculation score of **[**the joint training program**]** awarded by **[**Keuka University**]**?	Our model recognized a long enity as two short entities
Em đang là **[**học sinh lớp 12**]** ạ I'm a **[**12th grade student**]**	Em đang là học sinh **[**lớp 12**]** ạ I'm a **[**12th grade**]** student	Our model could not recognize all words in a long entity

6. Conclusion

We have presented in this paper an empirical study on question analysis, the first and crucial step towards an automatic Vietnamese question answering system in the education domain. By integrating traditional statistical models and deep neural networks which can utilize both manually engineered and automatically learned features, our proposed models can accurately extract fourteen types of important information from Vietnamese questions. Our work, however, has some limitations that we discuss in the following. First, our work is institute-specific, i.e., VNU International School, and domain-specific, i.e., the education domain. The dataset needs to be updated if we want to build a similar system for other schools/universities or a system that is expected to answer questions from multidisciplinary domains. Second, due to budget limit, our annotated corpus is quite small with 3,600 sentences. The system could be better if we had a larger dataset which covers a wide range of questions. Finally, our work focuses only on question analysis, not a full question answering system. As future work, we plan to improve the performance of extraction models with state-of-the-art deep neural networks such as attention-based architectures. We also aim at building a QA system, which can automatically answer questions from Vietnamese students at Vietnam National University, Hanoi.

***Acknowledgements*:** This research is funded by Vietnam National University, Hanoi (VNU) under Project number QG.19.59.

References

1. A b u j a b a l, A., R. S. R o y, M. Y a h y a, G. W e i k u m. ComQA: A Community-Sourced Dataset for Complex Factoid Question Answering with Paraphrase Clusters. – In: Proc. of NAACL-HLT, 2019, pp. 307-317.
2. B a c h, N. X., L. T. N. C h a m, T. H. N. T h i e n, T. M. P h u o n g. Question Analysis for Vietnamese Legal Question Answering. – In: Proc. of 9th International Conference on Knowledge and Systems Engineering (KSE'17), 2017, pp. 154-159.
3. B a c h, N. X., T. K. D u y, T. M. P h u o n g. A POS Tagging Model for Vietnamese Social Media Text Using BiLSTM-CRF with Rich Features. – In: Proc. of 16th Pacific Rim International Conferences on Artificial Intelligence (PRICAI'19), Part III, 2019, pp. 206-219.
4. C o s t a, L. F. Esfinge – a Question Answering System in the Web Using the Web. – In: Proc. of European Chapter of the Association for Computational Linguistics, 2006, pp. 127-130.
5. D u o n g, H. T., H. B a o-Q u o c. A Vietnamese Question Answering System in Vietnam's Legal Documents. – In: Proc. of 13th International Conference on Computer Information Systems and Industrial Management Applications, 2014, pp. 186-197.
6. E l m a n, J. L. Finding Structure in Time. – Cognitive Science, Vol. **14**, 1990, No 2, pp. 179-211.
7. G r a v e s, A., J. S c h m i d h u b e r. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. – Neural Networks, Vol. **18**, 2005, No 5-6, pp. 602-610.
8. H o c h r e i t e r, S., J. S c h m i d h u b e r. Long Short-Term Memory. – Neural Computing, Vol. **9**, 1997, No 8, pp. 1735-1780.
9. H u a n g, Z., M. T h i n t, Z. Q i n. Question Classification Using Head Words and Their Hypernyms. – In: Proc. of Conference on Empirical Methods in Natural Language Processing, 2008, pp. 927-936.

10. I o a n n i d o u, A., E. C h a t z i l a r i, S. N i k o l o p o u l o s, I. K o m p a t s i a r i s. Deep Learning Advances in Computer Vision with 3D Data: A Survey. – ACM Computing Surveys, Vol. **50**, 2017, No 2, pp. 1-38.
11. K i l i c o g l u, H., A. B. A b a c h a, Y. M r a b e t, K. R o b e r t s, L. R o d r i g u e z, S. S h o o s h a n, D. D e m n e r-F u s h m a n. Annotating Named Entities in Consumer Health Questions. – In: Proc. of 10th International Conference on Language Resources and Evaluation (LREC'16), 2016, pp. 3325-3332.
12. K i m, Y. Convolutional Neural Networks for Sentence Classification. – In: Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP'14), 2014, pp. 1746-1751.
13. L a f f e r t y, J., A. M c C a l l u m, F. P e r e i r a. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. – In: Proc. of ICML, 2001, pp. 282-289.
14. L e C u n, Y., B. E. B o s e r, J. S. D e n k e r, D. H e n d e r s o n, R. E. H o w a r d, W. E. H u b b a r d, L. D. J a c k e l. Handwritten Digit Recognition with a Back-Propagation Network. – In: Proc. of Advances in Neural Information Processing Systems, 1990, pp. 396-404.
15. L e-H o n g, P., D. T. B u i. A Factoid Question Answering System for Vietnamese. – In: Proc. of 2018 Web Conference Companion, Track: First International Workshop on Hybrid Question Answering with Structured and Unstructured Knowledge, 2018, pp. 1049-1055.
16. L i g o z a t, A. L. Question Classification Transfer. – In: Proc. of 51st Annual Meeting of the Association for Computational Linguistics (ACL'13), 2013, pp. 429-433.
17. M a, M., L. H u a n g, B. X i a n g, B. Z h o u. Group Sparse CNNs for Question Classification with Answer Sets. – In: Proc. of 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), 2017, pp. 335-340.
18. M a d a b u s h i, H. T., M. L e e. High Accuracy Rule-Based Question Classification Using Question Syntax and Semantics. – In: Proc. of 26th International Conference on Computational Linguistics (COLING'16), 2016, pp. 1220-1230.
19. M e n d e s, A. C., L. C o h e u r, P. V. L o b o. Named Entity Recognition in Questions: Towards a Golden Collection. – In: Proc. of 7th International Conference on Language Resources and Evaluation (LREC'10), 2010, pp. 574-580.
20. M o l l a, D., M. V. Z a a n e n, D. S m i t h. Named Entity Recognition for Question Answering. – In: Proc. of Australasian Language Technology Workshop, 2006, pp. 51-58.
21. N g u y e n, D. Q., D. Q. N g u y e n, S. B. P h a m. A Vietnamese Question Answering System. – In: Proc. of International Conference on Knowledge and Systems Engineering (KSE'09), 2009, pp. 26-32.
22. S h a r m a, V., N. K u l k a r n i, S. P. P o t h a r a j u, G. B a y o m i, E. N y b e r g, T. M i t a m u r a. BioAMA: Towards an End to End BioMedical Question Answering System. – In: Proc. of BioNLP Workshop, 2018, pp. 109-117.
23. S r i h a r i, R., W. L i. A Question Answering System Supported by Information Extraction. – In: Proc. of 6th Conference on Applied Natural Language Processing, 2000, pp. 166-172.
24. T r a n, V. M., V. D. N g u y e n, O. T. T r a n, U. T. T. P h a m, T. Q. H a. An Experimental Study of Vietnamese Question Answering System. – In: Proc. of International Conference on Asian Language Processing (IALP'09), 2009.
25. T r a n, V. M., D. T. L e, X. T. T r a n, T. T. N g u y e n. A Model of Vietnamese Person Named Entity Question Answering System. – In: Proc. of 26th Pacific Asia Conference on Language, Information and Computation (PACLIC'12), 2012, pp. 325-332.
26. T r a n, D. H., C. X. C h u, S. B. P h a m, M. L. N g u y e n. Learning Based Approaches for Vietnamese Question Classification Using Keywords Extraction from the Web. – In: Proc of International Joint Conference on Natural Language Processing (IJCNLP'13), 2013, pp. 740-746.
27. V o o r h e e s, E. M. Question Answering in TREC. – In: Proc. of 10th International Conference on Information and Knowledge Management (CIKM'01), 2001, pp. 535-537.
28. Y a n g, J., Y. Z h a n g. NCRF++: An Open-Source Neural Sequence Labeling Toolkit. – In: Proc. of ACL-System Demonstrations, 2018, pp. 74-79.
29. Z h a n g, S., L. Y a o, A. S u n, Y. T a y. Deep Learning Based Recommender System: A Survey and New Perspectives. – ACM Computing Surveys, Vol. **52**, 2019, No 1, pp. 1-38.

Received: 17.12.2019; Second Version: 21.02.2020; Accepted: 25.02.2020 (fast track)