ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Nguyễn Trọng Chỉnh

TUYỂN TẬP

CÁC CÔNG TRÌNH NGHIÊN CỨU

ĐỀ TÀI LUẬN ÁN

MÔ HÌNH VÀ PHƯƠNG PHÁP LẬP LUẬN ĐỂ TRẢ LỜI CÁC CÂU HỎI "TẠI SAO" DỰA TRÊN CÁCH TIẾP CẬN PHÂN TÍCH DIỄN NGÔN TIẾNG VIỆT

TP HỒ CHÍ MINH – NĂM 2022

DANH MỤC CÁC CÔNG TRÌNH NGHIÊN CỨU

[CT.1] Nguyen Trong Chinh, Dang Tuan Nguyen: Construction of Vietnamese Argument Annotated Dataset for Why-Question Answering Method. ICTCC 2016: 124-132. DOI: 10.1007/978-3-319-46909-6_12

[CT.2] Nguyen Trong Chinh, Dang Tuan Nguyen: Towards an Argument-Based Method for Answering Why-question in Vietnamese Language. NICS 2016: 130-134. DOI: 10.1109/NICS.2016.7725637

[CT.3] Nguyen Trong Chinh, Dang Tuan Nguyen: Towards Building Vietnamese Discourse Treebank. SoICT 2017: 63-69. DOI: 10.1145/3155133.3155200

[CT.4] Nguyen Trong Chinh, Dang Tuan Nguyen: Building a Vietnamese Dataset for Natural Language Inference Models. FDSE 2021: 185-199. DOI: 10.1007/978-981-16-8062-5_12

[CT.5] Nguyen Trong Chinh, Dang Tuan Nguyen: A Vietnamese Answer Extraction Model Based on PhoBERT. ACOMP 2021: 112-119. DOI: 10.1109/ACOMP53746.2021.00022

[CT.6] Nguyen Trong Chinh, Dang Tuan Nguyen: "Elementary Discourse Unit Segmentation for Vietnamese Texts". In Int. J. of Intelligent Information and Database Systems. 2022. DOI: 10.1504/IJIIDS.2022.124090 (**Scopus – Q4**)

[CT.7] Nguyen Trong Chinh, Dang Tuan Nguyen: "Building a Discourse-Argument Hybrid System for Answering Vietnamese Why-questions". In Computational Intelligence and Neuroscience. 2021. DOI:10.1155/2021/6550871 (**SCIE – Q1, IF: 3.633**)

[CT.8] Nguyen Trong Chinh, Dang Tuan Nguyen: "Building a Vietnamese Dataset for Natural Language Inference Models". In SN Computer Science. 2022. DOI: 10.1007/s42979-022-01267-x

Phan Cong Vinh
Leonard Barolli (Eds.)

LNICST 168

Nature of Computation and Communication

Second International Conference, ICTCC 2016
Rach Gia, Vietnam, March 17–18, 2016
Revised Selected Papers

ICST
Springer

Construction of Vietnamese Argument Annotated Dataset for Why-Question Answering Method

Chinh Trong Nguyen and Dang Tuan Nguyen

Faculty of Computer Science
University of Information Technology, VNU-HCM
Ho Chi Minh city, Vietnam
{chinhnt,dangnt}@uit.edu.vn

Abstract. In this paper, the method of building a Vietnamese Argument Annotated Dataset (VAAD) is presented. This dataset contains argumentative data which can be used to answer the why-questions. Therefore, it is important to discover the characteristics of the answers of why-questions to develop why-question answering method by using causal relations between texts. In addition, this dataset can be used to generate the testing dataset for evaluation of answering method. In order to build the dataset, a process of four steps is proposed after studying relevant problems. To briefly evaluate the method, an experiment is conducted to show the applicability of the method in practice.

Keywords: discourse analysis, why-question answering, Vietnamese Argument Annotated Dataset.

1 Introduction

At present, the development of question answering systems for Vietnamese language can be founded on researched solutions of answering the factoid questions [13, 14, 15, 16]. These solutions are mostly based on knowledge mining techniques therefore they need a large annotated corpus to train, to evaluate and to develop.

Although why-questions are rarely asked, 5% of all questions asked according to the observation of Hovy [1], they seem to be the important type of question because their answers, found by causal relations in discourse structures instead of the bag of words in texts, provide the reasons about problems. Therefore, building a why-question answering (why-QA) system for Vietnamese language has been conducted. However, the Vietnamese corpus for researching why-question answering methods is lacked. Although TREC has developed testing datasets for question answering systems for many years, the datasets mostly contain factoid questions and they are written in English. At present, it is important to build a large Vietnamese annotated dataset for researching and testing why-QA.

For the above reasons, a Vietnamese Argument Annotated Dataset (VAAD) for why-questions should be built to develop why-QA answering methods. The dataset should be suitable for developing many answering methods and evaluation. In this paper, the process of building VAAD for why-questions is presented in five sections. Section 1 introduces the exigence of developing VAAD. Section 2 explores some problems related to building the dataset. According to these problems, the annotation format of Vietnamese VAAD and the building process is presented section 3. Then, the experiment of building the dataset is presented in section 4. At the end, some conclusions are drawn in section 5.

2 Related works

The methods of question answering can be divided into two approaches that are knowledge mining, as in [13, 14, 15, 16], and knowledge annotation, as in [17]. The methods based on knowledge mining techniques have the advantage of information redundancy from the internet. The redundancy of information can be utilized to propose question answering methods which do not need to use complex natural language processing techniques. Therefore, many researches in question answering have focused on this approach.

According to the knowledge mining approach, developing question answering methods need large datasets to discover the patterns which are used to find the candidate answers. These datasets are also used to test the question answering methods. These datasets should be not only collected but also annotated into a specific format. The format of a dataset depends on the feature analyzed by the researching methods. For example, Saint-Dizier's dataset in [12] is annotated by using Rhetorical Structure Theory (RST) [7] because the question answering method is based on the argumentation which is identified in discourse structure of the document.

In why-QA, the question answering method can be divided into two types: cue-based method and discourse-based method. The cue-based methods are developed with clues as in [11] or with cue words and paragraph retrieval techniques as in [2]. They have the simplicity in analysis but the results are quite low because the semantic features have not been analyzed yet. In contrast, the discourse-based methods are developed with discourse structure of the document as in [4, 5, 6, 12]. In this type, the methods have to use the context of the sentences in a document to build the relations between them. These relations express the intention of the writer. Among these relations, the causal relations between sentences form the writer's argument structures. The discourse-based methods need more complicated analysis but their results are more relevant to the questions than the cue-based ones. Despite of the differences, these types of answering method need why-QA datasets for training and testing. These datasets have to be built for each research project because there are no appropriate dataset for all purposes.

In discourse structure of document, there are two approaches of representation. In the RST representation [7], a document is a “tree of spans”. Each span, which can be a clause, a sentence or a paragraph, links to another span following rhetorical relations to form a larger span. These spans are still presented in text therefore they are easy to

search. In the Discourse Representation Theory (DRT) [18], a document is a set of Discourse Representation Structure (DRS) which is a group of first-order logic expressions. These representations can be used to reason in order to find new information, however it is complex to build a set of DRS from a document in natural language. In other aspect of discourse structure, the visual structure of a document also affects its discourse structure as Power shown in [8].

3 Building VAAD for developing why-QA method

The purpose of building the VAAD is to develop why-QA methods. These methods can be cue-based or discourse-based approaches therefore the dataset should be annotated in a simple format so that it can be used easily. In addition, the dataset can be used to generate testing sets by transforming the result parts of causal relations in to why-questions. For example, the causal relation "Tom is not allowed to ride a bicycle because Tom is young" has the result part "Tom is not allowed to ride a bicycle". Thus, a why-question "why Tom is not allowed to ride a bicycle?" can be built by transforming the result part. In order to make more complex why-questions, synonyms or similar semantic phrases can be used to expand the original result parts.

The process of building VAAD dataset has four steps that are documents collecting, argument annotating, patterns extracting and argument annotated fragments collecting

3.1 Documents collecting

During the process of collecting documents containing arguments, the observations show that there are many news posts or comments without any arguments in them. These news posts or comments are often about new products, instructions, sports news. In order to collect documents containing arguments, Google[1] is used to search for document containing phrases which are more likely to appear in an argument, such as "tại sao" (“why”), "công dụng của" (“the use of”), "hạn chế của" (“the disadvantages of”). Then, the links in google search results are extracted and used to download the origin web pages. After that, the scripts, banners, etc. of the web pages are eliminated and the texts of main content of the web pages are extracted. These texts form a dataset for annotating in the next step.

3.2 Argument annotating

According to the simplicity of the RST representation, the dataset is annotated follow these rules:

- All spans which are not in any argument are unchanged.
- Spans, which are in a certain argument, are place in a pair of symbols "[" and "]"

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

- A span which is an argument is annotated as follow: causal part and result part are place in a pair of symbols "{" and "}" in which they follow a notation of their role in the argument; the cue phrase which informs the type of causal relation is unchanged. Figure 1 illustrates an annotated argument fragment.
- An argument can be a part of another argument as shown in Figure 2.

[{CIRCUMSTANCE Theo nghiên cứu công bố trên tạp chí khoa học PNAS hồi tháng 5 của CSIRO (Tổ chức Nghiên cứu Khoa học và Công nghiệp Liên bang Australia), hải sâm là nguồn dược liệu và thực phẩm có giá trị cao tại thị trường châu Á**}**. ***Do đó,*** **{OUTCOME** nó đang bị đánh bắt quá mức**}]**

(source: VnExpress.net)

Fig. 1. A structure of an argument annotated fragment. The bold words are the roles of two parts in a causal relation (CIRCUMSTANCE - OUTCOME). The bold, italic words, "Do đó" (therefore) is a cue phrase indicates the circumstance - result relation.

[{CIRCUMSTANCE Bóng đè là một hiện tượng tâm sinh lý điển hình của hệ thống tính năng cơ thể. **[{CIRCUMSTANCE** Nó được ví như hệ thống "Rơle" trong kỹ nghệ, nhằm bảo vệ cơ thể bằng cách vô hiệu hóa những mệnh lệnh "tái sinh" từ hệ điều khiển đến hệ thống vận động trong lúc cơ thể đang được duy trì ở trạng thái "nghỉ" **-} *do vậy*** **{OUTCOME** sự "đè nén" ở đây không có thực thể mà chỉ là hiệu ứng do "cái bóng" gây ra mà thôi**}]**. **[{CIRCUMSTANCE** Mệnh lệnh "tái sinh" chỉ là "mệnh lệnh ảo" được não bộ tái hiện lại, hoặc "sáng tác ra" trong giai đoạn ta đang ngủ**}**, ***vì vậy*** **{OUTCOME** mệnh lệnh loại này chỉ được "chiếu thử" lên màn hình của não bộ mà không được thực thi bởi các cơ quan chức năng của cơ thể**}]}**.

Chính vì vậy, **{OUTCOME** trong suốt giai đoạn mộng mị của giấc ngủ, hoặc trong lúc bị "bóng đè", cơ thể vẫn được duy trì trạng thái "nằm yên" bởi các cơ bắp bỗng nhiên bị "mất điện" nhằm ngăn cản các hành động có thể diễn ra theo kịch bản phiêu lưu quái dị và lãng mạn của não bộ vẽ vời ra**}]**.

(source: VnExpress.net)

Fig. 2. An argument can be a part of another argument. In this figure, the first paragraph is the causal part and the second paragraph is the result part of an argument. There are two arguments in the first paragraph.

By using these rules, the arguments in document are easy to extract. In addition, if there is any further language analysis needed, it can be applied easily to discover more precise patterns. In this format, the causal relations in RST is divided into four types according to [4]: rationale - effect, purpose - outcome, circumstance - outcome and means - outcome.

3.3 Patterns extracting

After identifying arguments by annotating the causal relations. The patterns containing cue phrases and some specific marks such as periods, commas, new-lines are also identified. A causal relation can be an inner-sentence, an inter-sentence or an inter-paragraph relation.

In an inner-sentence relation, as in Figure 3 all parts of the relation are bounded in two periods and they do not contain any period. In an inter-sentence relation, as in Figure 1 above, there is only one period; and in an inter-paragraph relation, as in Figure 2 above, there are one more new-line symbols.

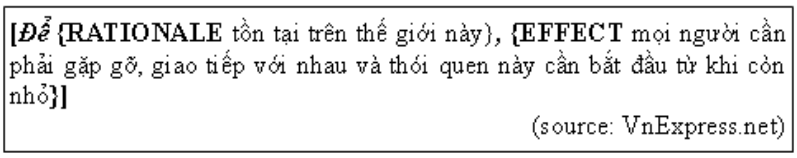


Fig. 3. The inner-sentence relation in which all parts of the relation are bounded in two periods and there is no period in all parts of the relation.

In this step, the cue phrases are used as core feature to identify the argument because the cue phrase have stably meaning of discourse function as shown in [7, 9]. Therefore, the patterns are manually identified and used to extract arguments having the same patterns in websites to enrich the dataset

3.4 Argument annotated fragments collecting

By using the patterns discovered in step 3, a crawler is used to fetch the news posts on websites to extract the argument annotated fragment. By using the crawler, the process of building VAAD is reduced greatly in cost of manually collecting and annotating. However, this method has a disadvantage of not collecting arguments of new patterns. The extracted arguments of collected news posts are automatically annotated with the proposed format according to the patterns which are used to extract them.

4 Experiment

In order to evaluate the method of building VAAD, 34 articles are collected according to step 1 and annotated as describing in step 2. Then, the 49 argument fragment patterns, as shown in Table 1 are manually identified. Then, these patterns are represented in regular expressions to collect argument fragments.

Table 1. The list of manually identified cue phrases.

Phrase	Relation type

... . Vì vậy,	inter-sentence
... . Bởi vậy, ...	inter-sentence
... . Vì thế ...	inter-sentence
... . Đi u này làm cho ...	inter-sentence
... Do đó, ...	inter-paragraph
... do ...	inner-sentence
Nhờ ..., ...	inter-sentence
... .Thế nên	inter-sentence
... . Kết quả ...	inter-sentence
... .Vì vậy ...	inter-sentence
... . Do vậy, ...	inter-sentence
Đ ..., ...	inner-sentence
..., chính vì vậy ...	inner-sentence
... . Do vậy ...	inter-sentence
... do vậy ...	inner-sentence
... . Vì lẽ đó, ...	inter-sentence
... là nguyên nhân chính dẫn tới ...	inner-sentence
Do ... mà ...	inner-sentence
... . Đi u này khi n ...	inter-paragraph
... là do ...	inner-sentence
... cho nên ...	inner-sentence
..., do vậy ...	inner-sentence
... Chính vì vậy, ...	inter-paragraph
... Vậy, ...	inter-paragraph
... dẫn đến ...	inner-sentence
... vì ...	inner-sentence
... , vì vậy ...	inner-sentence
... . Đi u này d nđ n ...	inter-sentence
... . Đây là lý do ...	inter-sentence
... , đây là lý do t i sao ...	inner-sentence
Bởi vì ... nên ...	inner-sentence
... là nhờ ...	inner-sentence
Nguyên nhân ... do ...	inner-sentence
Với ... , ...	inner-sentence
Nhờ ... mà ...	inner-sentence
... đ ...	inner-sentence
... với mục đích ...	inner-sentence
... nên ...	inner-sentence
... gây ...	inner-sentence
... Như v y, ...	inter-paragraph
... ảnh hưởng tới ...	inner-sentence
Vì ... nên ...	inner-sentence
Bởi ... , ...	inner-sentence
... . Và đó là lý do ...	inter-sentence
đ ... thì ...	inner-sentence
... cho thấy ...	inner-sentence
... khiến ...	inner-sentence
... bằng cách ...	inner-sentence
... bởi ...	inner-sentence

After identifying argument fragment patterns, a set of 608 articles downloaded from internet using crawler are process with the patterns to generate 2609 fragments. The cue phrases associated with these fragments are presented in Table 2 to show

which cue phrases are frequently used. In order to evaluate the precision of the argument identification method, 250 fragments are randomly selected in 2609 fragments. These 250 fragments are then manually check if they are argument fragments. After checking, there are 195 fragments are argument fragments which yield the precision of 0.78.

Table 2. The list of cue phrases used to extract 2609 fragments and their number of use.

Phrase	Number of use
... đ ...	923
... do ...	328
... nên ...	277
... vì ...	240
... khiến ...	158
... gây ...	163
... bởi ...	114
... cho thấy ...	91
... . Vì thế, ...	69
... nhằm ...	55
... biến thành ...	47
... bằng cách ...	30
... . Kết quả ...	21
... dẫn đến ...	21
... ảnh hưởng đến ...	21
... . Do đó ...	14
... . Vì vậy, ...	13
Đ ... , ...	6
... , vì thế ...	3
... . Nhờ đó, ...	3
... là nhờ ...	3
... làm cho ...	3
... với mục đích ...	3
... . Do vậy, ...	1
... cho nên ...	1
... nguyên nhân chính ...	1

The reasons of the wrong identifying argument fragments are the ambiguity of the cue phrase and the misidentifying inter-paragraph relation. The ambiguity of cue phrase such as, "đ " (in order to) and 'đ " (to put), can be overcome by POS tag process before identifying patterns and extracting argument fragments. The misidentifying inter-paragraph relation is more difficult to overcome. It requires a completely RST structure of the document to identify which paragraphs form a span in RST. However, the number of inter-paragraph argument fragments collected are not very large. Therefore this method can be used to build VAAD for developing a why-QA method.

The experiment result shows that the proposed method can be applied in practice with the higher precision by applying POS tagging task.

5 Conclusions and Future works

In this paper, the research on building VAAD for developing why-QA method is presented. This dataset is important to find out the characteristics of argument of text fragments to answer the why-questions in Vietnamese. In addition, the testing dataset for why-QA method can be generated from this dataset. The testing dataset is also important to evaluate the answering method. Because the arguments are some kinds of RST relations, this paper proposes a method of automatically identifying argument fragments from news posts in the internet using cue phrases. The cue phrases are used in this method because their linguistic functions of discourse are stable. Therefore, the process of four steps which are collecting documents, argument annotating, patterns extracting and argument annotated fragments collecting is proposed to build the dataset.

According to the proposed process, an experiment has been conducted and it shows that the process can be apply to automatically build the practical VAAD for developing why-QA method after POS tagging the documents for extracting patterns and collecting argument fragments.

In future, Vietnamese RST parser should be developed to overcome the misidentifying inter-paragraph causal relation to enrich VAAD.

References

1. Hovy, E.H., Hermjakob, U., Ravichandran, D.: A Question/Answer Typology with Surface Text Patterns. In: 2nd International conference on Human Language Technology Research, pp. 247-251. California (2002)
2. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.: Using syntactic information for improving why-question answering. In: 22nd International Conference on Computational Linguistics, pp. 953--960. Manchester, United Kingdom (2008)
3. Verberne, S.: Developing an approach for why-question answering. In: 11th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pp. 39-46. Trento, Italy (2006)
4. Delmonte, R., Pianta., E.: Answering why-questions in closed domains from a discourse model. In: Conference on Semantics in Text Processing, pp. 103--114. ACL Stroudsburg. PA, USA (2008)
5. Oh, J., Torisawa, K., Hashimoto, C., Sano, M., Saeger, S. D.: Why-question answering using Intra- and Inter-Sentential Causal Relations. In: 51st Annual Meeting of the Association for Computational Linguistics, pp. 1733--1743. ACL Anthology, Sofia, Bulgaria (2013)
6. Higashinaka, R., Isozaki, H.: Corpus-based Question Answering for why-Questions. In: 3rd International Joint Conference of Natural Language Processing, pp. 418-425. Hyderabad, India (2008)
7. Mann, W. C., Thompson, S. A.: Rhetorical structure theory: towards a functional theory of text organization. Text vol. 3, no. 8, pp. 243--281 (1988)
8. Power, R., Scott, D., Bouayad-Agha, N.: Document Structure. Computational Linguistics vol. 29, no. 2, pp. 211-260 (2003)
9. Marcu, D.: The Rhetorical Parsing of Natural Language Texts. In: 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter

of the Association for Computational Linguistics, pp. 96--103. ACL Stroudsburg, PA, USA (1997)
10. Hwee, T. N., Leong, H. T., Lai, J. P. K.: A machine learning approach to answering questions for reading comprehension tests. In: The 2000 Joint SIGDAT conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 124--132. ACL Stroudsburg, PA, USA (2000)
11. Riloff, E., Thelen, M.: A Rule-based Question Answering System for Reading Comprehension Tests. In: The 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems, pp.13--19. ACL Stroudsburg, PA, USA (2000)
12. Saint-Dizier, P.: Processing Natural Language Arguments with the <TextCoop> Platform. Argument & Computation vol. 3, no. 1, pp. 49--82. Taylor & Francis (2012)
13. Zheng, Z.: AnswerBus question answering system. In: The 2nd international conference on Human Language Technology Research, pp. 399--404. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA (2002)
14. Clarke, C., Cormack, G., Kemkes, G., Laszlo, M., Lynam, T., Terra, E., and Tilker, P.: Statistical Selection of Exact Answers (MultiText Experiments for TREC 2002). In: TREC, pp. 823--831. NIST (2002)
15. Brill, E., Dumais, S., Banko, M.: An analysis of the AskMSR question-answering system. In: The ACL-02 conference on Empirical methods in natural language processing, p.257-264. ACL Stroudsburg, PA, USA (2002)
16. Buchholz, S., Daelemans, W.: Shapaqa: Shallow parsing for question answering on the world wide web. In: Euroconference Recent Advances in Natural Language Processing, pp. 47--51, Tzigov Chark, Bulgaria (2001)
17. Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J. J., Marton, G., McFarland, A. J., Temelkuran, B.: Omnibase: Uniform Access to Heterogeneous Data for Question Answering. In: The 6th International Conference on Applications of Natural Language to Information Systems, p.230-234. Springer-Verlag (2002)
18. Kamp, H.: Discourse Representation Theory, In: Gabbay, D., Guenthner, F. (eds.). Handbook of Philosophical Logic, vol. 15, pp. 125-394. Springer, Netherlands (2011)

Proceedings

Edited by Vo Nguyen Quoc Bao, Nguyen Le Hung, and Tran Trung Duy

2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science

NICS 2016

September 14 - 16, 2016
Danang, Vietnam

SCIENCE AND TECHNICS PUBLISHING HOUSE

Towards an Argument-Based Method for Answering Why-question in Vietnamese Language

Chinh Trong Nguyen and Dang Tuan Nguyen
Faculty of Computer Science
University of Information Technology, VNU-HCM
Ho Chi Minh city, Viet Nam

***Abstract*—In this paper, an argument-based method of answering why-questions in Vietnamese is presented. This method is developed in different way from many approaches which use cue phrases of causal relation to find the answers for why-questions. In this method, the arguments is extracted firstly, then the causal part and consequential part of every argument are split in order to index the consequential part. When a why-question is asked, the asking information is extracted and used to search for the reason, then the reason is used to identify the paragraph which can be used to answer the question. For evaluation, an experiment with keyword-based information retrieval and simple argument collecting process is conducted to show the applicability of the method.**

Keywords—argument analysis, argument-based answering, Vietnamese why-question answering.

I. INTRODUCTION

After TREC-8, numerous researches have been conducted to explore many methods of answering the questions in natural language. These researches have focused on how to retrieve a short information from a set of disposable documents for answering each TREC's question. The TREC's questions were usually the factoid questions which asked about person, location, quantity, etc. Therefore, many Question Answering (QA) systems have been built such as Shapaqa [8], AnswerBus [9], MultiText [10], AskMSR [11], or expanded such as START [7]. In 2006, the Ephyra [12] was proposed as a framework for answering the factoid questions in the open-domain. In 2010, Watson [13] was introduced as an impressive QA system which had good experiment results in Jeopardy quiz show. These research results show that the methods of answering factoid questions are basically solved. In order to improve the precision and the confidence of the answers for factoid questions, the semantic analysis and knowledge inference should be applied.

Although there are some impressive results in answering the factoid questions, the methods of answering the non-factoid questions, such as why-question, still have to be researched. The why-questions are not asked frequently (5% of all questions [16]) but their answers are important because they show the reasons for the circumstances or behaviours in questions. These reasons can be used for assessment or improvement of human behaviors in future. Therefore, a method of answering why-questions in Vietnamese language has been conducted. This method, which is a combination of the argument structure and information retrieval, is a new approach of why-question answering method. In order to explain the method, there are four following sections are presented. Section 2 presents the related works in why-question answering. Section 3 introduces the approach of the method and explains why it is selected. Then, section 4 presents some experiments to show the applicability of the method. Finally, some conclusions are presented in section 5.

II. RELATED WORKS

In order to develop the method of answering why-question, some solutions have been studied. These solutions can be divided into two approaches that are Information Retrieval and Information Extraction (IR&IE) approach and Reading Comprehension approach.

In the IR&IE approach, the why-question answering methods are developed from factoid question answering methods. In Verberne's works [2], [3], the answers for a why question can be found in two-step process. In the first step, the candidate paragraphs are categorized into four sub-types of causal relation by the cue words appeared in these paragraphs and then they are selected by matching their type and the question's type. In the second step, the selected paragraphs are compared to the question with some features: syntactic structure, semantic structure and synonym. In [4], Oh et al proposed a why-question answering method which uses the causal relations as the main feature to select the candidate. The causal relations are identified by using cue phrases in Japanese. In addition, the excitation polarities are also identified to improve the accuracy of method. In Japanese why-QA system [5], Higashinaka proposed a method using causal expression and content similarity of the candidates to find the answers. The causal expressions are extracted from the EDR dictionary [5] which is annotated with semantic relations.

In the Reading Comprehension approach, because the questions and their answers are focused on short texts such as short stories, the answering methods are only concentrated on how to examine the answers. The passage retrieval process are not important in this approach. In order to answer the why-question, Riloff proposed a rule-based approach [6] which scores a sentence by its word match with the question and the

978-1-5090-2100-0/16/$31.00 ©2016 IEEE

appearance of words "want", "so" and "because". This approach is similar to the approach of Higashinaka which is based on cue phrases. In a different way, Delmonte proposed a why-question answering method which is based on discourse model of the text [1]. The discourse model is presented in first-order expressions and the answers are found by a reasoning process. This approach provides a different point of view in question answering and promises exciting solutions. However, there is a problem in this approach, that is how to parse the large documents into first-order expression accurately and efficiently.

III. Argument-based method for answering Why-question

According to the above solutions, there is mostly a common important feature for developing the answering method, that is the cue word or the cue phrase. The approach of Delmonte has to cope with an important challenge in parsing large documents into first-order expressions. Parsing large documents into first-order expression is a complex problem in Vietnamese documents. In order to answer why-question in Vietnamese language, a new method which is combination of argument structure identification and information retrieval techniques is proposed in this paper.

A. Utilizing the Argument Structure

In a document, there are some text structures in which a part of text, called "causal part", supplies information to explain the information of another part of text, called "consequential part". These two parts of text form an argument. In many cases, there are phrases, called "cue phrases", used to link two parts of the text structure. In order to answer why-questions, the relevant causal information of the asking problem should be identified. Therefore, many solutions use the cue phrases to identify the texts of causal part and then check the similarity of the identified texts and the question. These solutions can be used to find the answers which contain cue phrases of causal relation, such as *"because"*, *"this is why"*, etc. However, there are many texts which do not contain such the cue phrases but they can be used to answer the why question. The example text in Figure 1 show the reason of disabling the nuclear power plant Fukushima but there is no cue phrases of reason in Vietnamese language.

In Figure 1, Fukushima was disabled due to the release of radioactivity and there is no cue phrases of causal relation in Vietnamese, such as *"bởi vì"*, *"do"*, *"vì thế"*, etc. Although there is no cue phrases, the reader is able to identify the bold paragraph as the reason of disabling the nuclear power plant because he may know that the leak of radioactivity is harmful for the environment. Therefore, there might be an argument in the reader's thought that the plant was disabled because the radioactivity leak is harmful.

According to the above example, the answers for why-questions are intuitively identified by the relevant argument. Therefore, the proposed method is focused on the idea of finding arguments which are relevant to the question. The structure of argument contains two parts: the causal part and the consequential part. After finding the arguments, the causal part of the most relevant argument is extracted and used to search for the answer. Assuming an argument, as shown in Figure 2, might be collected, how to answer the question: *"Tại sao đóng cửa nhà máy điện hạt nhân Fukushima?"* (*"Why disable the nuclear power plant Fukushima?"*)

> "Công bố của ông Noda thể hiện lời cam kết của chính phủ là dừng việc làm mát các thanh nhiên liệu bên trong các lò phản ứng hạt nhân quá nóng vào cuối năm nay.
>
> Nhà máy điện hạt nhân Fukushima Daiichi đã bị phá hỏng hệ thống làm mát sau trận động đất và sóng thần hồi tháng 3 gây ra.
>
> Những người điều hành nhà máy điện Tokyo Electric Power đã nỗ lực phun nước làm mát các thanh nhiên liệu ở bên trong các lò phản ứng hạt nhân.
>
> **Sự cố tan chảy của các thanh nhiên liệu và vụ nổ ở hệ thống làm mát bên trong lò phản ứng hạt nhân, khiến rò rỉ một lượng lớn phóng xạ hạt nhân vào môi trường. Đây được coi là tai nạn hạt nhân tồi tệ nhất sau thảm họa Chernobyl (Liên bang Nga) năm 1986.**
>
> Thảm họa hạt nhân tồi tệ nhất trong lịch sử Nhật Bản đã làm dấy lên mối lo ngại nhiễm phóng xạ cho những người tiêu dùng ở phía đông Nhật Bản khi ăn những thức ăn như cá và các sản phẩm lương thực khách từ khu vực này."
>
> (Source: khoahoc.tv)

Fig. 1. A text contains the reason (bold paragraph) of disabling the nuclear power plant Fukushima and does not contain any Vietnamese cue phrases of reason.

In the argument in Figure 2, the causal part is the phrase (1) *"tính chất nguy hiểm tiềm ẩn của loại năng lượng này"* (*"the potential dangerous of this type of energy"*) and the consequential part is the phrase (2) *"theo tôi được biết một số nước phương tây đang bắt đầu đóng cửa các nhà máy điện hạt nhân"* (*"As I know, the Western nations have been disabling the nuclear power plants"*). Therefore, when identifying the reason of disabling Fukushima plant, because the phrase (2) is relevant to the asking information of the question, the phrase (1) is retrieved to find the candidate text. Then, the bold paragraph in Figure 1 are the most relevant to the phrase (1) therefore it is selected as an answer candidate.

> "Theo tôi được biết một số nước phương tây đang bắt đầu đóng cửa các nhà máy điện hạt nhân **vì** tính chất nguy hiểm tiềm ẩn của loại năng lượng này."
>
> (Source: vnexpress.net)

Fig. 2. An argument about the reason of disabling nuclear power plants with the cue word *"vì" ("because")* which is the bold one.

The method is proposed because of two reasons. Firstly, why-questions require an inference process to identify a chain of events which results the asking information. In order to infer the expected goal, the documents can be parsed into Discourse Representation Structure (DRS) [14] and then the first-order logic reasoning can be applied to find the goal. However, the

process of parsing large documents into first-order expressions, then choosing rules and managing temporary values while reasoning with the huge set of first-order expressions is a complex process. Therefore, instead of inference, the method uses existing argument which are results of inference process done by human. Secondly, because of redundancy of documents, there are many different statements for one problem. The restated texts might be used to extract arguments which are relevant to the asking information. Therefore, the arguments will be collected to build up a knowledge for answering the why-questions.

B. Answering Method for Vietnamese Why-question

According to idea of utilizing the argument structure, the why-question answering method for Vietnamese language is proposed. This method contains three process: argument collecting, argument indexing and answer finding.

In the argument collecting process, every collected document is analyzed to extract the arguments. An argument is identified by the discourse relations between phrases in a sentence, called inner-sentential relations, or between sentences in a paragraph, called inter-sentential relations. The discourse relations are identified by using cue phrases as in [15] because the cue phrases have the stable function in discourse. Then, each argument is split into causal part and consequential part according to the cue phrase it contains. The argument collecting has been described in other paper.

After collecting, the arguments are indexed with consequential part in argument indexing process. This index is used to find the argument which the consequential part is relevant to the information of the why-question.

After indexing, the answer finding process contains four steps as follow:

- Step 1 - Question analysis: the why-question is analyzed to get the asking information.
- Step 2 - Query formulation: a query is generated from the asking information to search against the index created in argument indexing process.
- Step 3 - Reason identification: if there is any arguments found, the reason A_1 of asking information is the combination of causal part of the most relevant arguments. The reason A_1 is called level 1 reason. In order to expand to the further reason, called level 2 reason, the reason A_1 can be used to find more arguments and then the causal part of these arguments are combined to the reason A_2. This expansion can be repeated. However, it is recommended to limit expanding to level 2.
- Step 4 - Candidate identification: the reason identified in step 3 is used to generate a query to find the appropriate paragraphs by a passage retrieval. The most relevant paragraphs can be used to answer the question.

To illustrate the method, the example in [1] is used. In this example, the text is written about the maple to explain why the tree is called *"sugar"* maple. In the text, the cue phrases of reason is the phrase *"This is why"* as shown in Figure 3.

> "Maple syrup comes from sugar maple trees. At one time, maple syrup was used to make sugar. This is why the tree is called a "sugar" maple tree."
>
> (Source: [1])

Fig. 3. The example of a text explaining the reason of the name "sugar" maple. The text contains the cue phrase of reason "This is why". This example is in [1].

In order to answer the question *"Why the tree is call sugar maple tree?"* [1], the following processes are performed:

- Argument Indexing: the consequential part of the above argument is indexed as keyword based document retrieval.
- Answer finding: firstly, the question *"Why the tree is called sugar maple tree?"* is analyzed to identify the asking information as *"the tree is called sugar maple tree"*. Then, this information can be used as the query to find the argument with keyword based document retrieval. Because the similarity of the query and the consequential part, the reason, which is the sentence A1 *"At one time, sugar syrup is used to make sugar"*, is return. Finally, the sentence A1 is used to find the text to answer the question and the paragraph in Figure 3.

Assuming there are a paragraph written about sugar beet without any cue phrases of reason as shown in Figure 4 and a question *"why the plant is called sugar beet?"*, how to find the answer?

> "An unrefined sugary syrup can be produced directly from sugar beet. This thick, dark syrup is produced by cooking shredded sugar beet for several hours, then pressing the resulting mash and concentrating the juice produced until it has the consistency similar to that of honey. No other ingredients are used. In Germany, particularly the Rhineland area, this sugar beet syrup (called Zuckerrüben-Sirup or Zapp in German) is used as a spread for sandwiches, as well as for sweetening sauces, cakes and desserts."
>
> (Source: wikipedia.org)

Fig. 4. A paragraph written about the sugar beet without any cue phrases of reason..

In order to answer the question *"Why the plant is called sugar beet?"*, the asking information *"the plant is called sugar beet"* is extracted. Then, it is used as a query to search the reason in the index above. Because the query is similar to *"the tree is called sugar maple tree"*, the retrieved reason A_1 is *"at one time, sugar syrup is used to make sugar"*. The A_1 is then used to search the paragraph which can be used to answer the question. The paragraph in Figure 4 can be retrieved because its first sentence is similar to A_1. Therefore, the answers can be extracted without any cue phrases in the documents.

IV. EXPERIMENT

In order to evaluate the proposed method, the experiment is set up as follow:

- The crawler is to collect web page text contents. There are two collections of the web content A and B containing 466 texts and 807 texts respectively.
- The argument collector uses cue phrases of reason in Vietnamese to extract the arguments contained in the two document collections. This process is presented in other paper.
- The search tool is developed from Lucene [17] to find the reason for asking information.

There are two tests conducted with the document collection A and B and the set of why-question. The results are shown in Table 1.

TABLE I. THE RESULTS OF THE TEST WITH DOCUMENT COLLECTION A AND B

No.	Question (in Vietnamese)	Result (A)	Result (B)
1	Tại sao CNTT phải là công cụ mới để tổ chức lại hệ thống giáo dục?	0	0
2	2. Tại sao phải giúp trẻ thoải mái khi học Toán?	0	1
3	Tại sao Suzuki tiếp tục đóng góp và chia sẻ đến với cộng đồng qua chương trình "Suzuki chào đón tân sinh viên 2012"?	0	1
4	Tại sao Bà Rịa - Vũng tàu thực hiện chương trình sữa học đường	1	1
5	Tại sao cho trẻ dùng điện thoại di động?	0	0
6	Tại sao nên học ở Đại học Quốc tế Sài Gòn?	0	0
7	Tại sao phải xây dựng mô hình đại học sáng tạo?	0	1
8	Tại sao mong muốn lớn nhất của cô là mái ấm nhỏ hạnh phúc?	0	0
9	Tại sao phải tôn vinh cá nhân hoạt động thiện nguyện	0	0
10	Tại sao Viettel duy trì hoạt động khuyến mãi cho sinh viên?	0	0
11	Tại sao các nhân vật phải nham hiểm và đầy toan tính?	0	0
12	Tại sao số lượng cá biển giảm?	1	1
13	Tại sao cây nắp ấm bắt động vật?	1	1
14	Tại sao cực quang xuất hiện ở Alaska?	1	1
15	Tại sao phải đóng cửa nhà máy điện hạt nhân Fukushima?	1	1
16	Tại sao rau an toàn và vệ sinh?	1	1
17	Tại sao số lượng tê giác giảm dần?	1	1
18	Tại sao số lượng voi giảm	0	0
19	Tại sao phải điều khiển trái cây chín	0	0
20	Tại sao phải thả voọc mông trắng về tự nhiên?	1	1
Total		8	11

In this experiment, there is only one answer which can be returned to each question. Every question asks about certain information in a document. This document is provided to answer finding process in the same way of reading comprehension. The answer is manually evaluated. An answer is correct if it contains a sentence that can be directly answer the question because a returned answer is a paragraph. According to the result in Table 1 the precision of the test with document collection A is 0.4 (8 correct answers per 20 why-questions) while the precision in the test with document collection B is 0.55 (11 correct answers per 20 why-questions). The tested results can be explained as follow:

- The arguments collected are not enough to answer the questions. Because the arguments are collected from web page contents of various domains, there are not enough arguments in a certain domain to answer the questions. Therefore, when more arguments are collected in document collection B, the number of correct answer increases by 3. In addition, it is easier to find the a correct answer for a question asking about general problem, such as question 12, 13, 14 and 15, and it is more difficult to find the answer for the question asking about private problem, such as question 8 and 11.
- The accuracy of argument collecting of process is quite low at 0.78. The misidentifying arguments cause the reasons are misidentified therefore the answers are not correct.
- The precision of reason retrieval is quite low because the pure keyword retrieval is used in this experiment.

Although the accuracy of the answers are low, the precision of 0.55 promises the better results if there are some improvements in argument collecting and in semantic retrieval for finding correct causal parts in the larger web document collection.

V. CONCLUSION

In this paper, the research on argument-based method for answering why-question in Vietnamese is presented. This method is developed from the new approach of why-question answering which is combination of the argument structure identification and information retrieval to find the answer. In this method, the arguments are collected to build a knowledge for finding the reasons of specific problems. The knowledge building process extracts the argument from a document collection by using cue phrases of causal relations, then splits the causal part and the consequential part of every argument in order to index these arguments by their consequential parts. This is called knowledge because it contains the arguments which are inference results done by human. By using the inference results, the answer of a why-question can be found by information retrieval.

Although the precision of test results, which is 0.55, is quite low, it promises the better results in future when some

improvements in argument collecting and semantic retrieval are applied with the larger document collection.

REFERENCES

[1] R. Delmonte, E. Pianta, "Answering why-questions in closed domains from a discourse model, " Conference on Semantics in Text Processing. ACL Stroudsburg, pp. 103-114, 2008.

[2] S. Verberne, L. Boves, N. Oostdijk, P. Coppen, "Using syntactic information for improving why-question answering, " 22nd International Conference on Computational Linguistics. UK, pp. 953-960, 2008.

[3] S. Verberne, "Developing an approach for why-question answering, " 11th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop. Italy, pp. 39-46, 2006.

[4] J. Oh, K. Torisawa, C. Hashimoto, M. Sano, S. D. Saeger, "Why-question answering using Intra- and Inter-Sentential Causal Relations, " 51st Annual Meeting of the Association for Computational Linguistics. Bulgaria, pp. 1733-1743, 2013.

[5] R. Higashinaka, H. Isozaki, "Corpus-based Question Answering for why-Questions, " 3rd International Joint Conference of Natural Language Processing. India, pp. 418-425, 2008.

[6] E. Riloff, M. Thelen, "A Rule-based Question Answering System for Reading Comprehension Tests, " The 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems. USA, pp.13-19. 2000.

[7] B. Katz, G. Marlon, G. Borchardt, A. Brownell, S. Felshin, D. Loreto, J. Rosenberg, B. Lu, F. Mora, S. Stiller, O. Uzuner, A. Wilcox, "External Knowledge Sources for Question Answering, " Proceedings of the 14th Annual Text REtrieval Conference. Gaithersburg, MD, 2005.

[8] S. Buchholz, W. Daelemans, "Shapaqa: Shallow parsing for question answering on the world wide web, " Euroconference Recent Advances in Natural Language Processing. Bulgaria, pp. 47--51, 2001.

[9] Z. Zheng, "AnswerBus question answering system, " 2nd International conference on Human Language Technology Research. USA, pp. 399-404, 2002.

[10] C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, P. Tilker, "Statistical Selection of Exact Answers (MultiText Experiments for TREC 2002), " TREC. NIST, pp. 823—831, 2002.

[11] E. Brill, S. Dumais, M. Banko, "An analysis of the AskMSR question-answering system, " ACL-02 conference on Empirical methods in natural language processing. USA, p.257-264, 2002.

[12] N. Schlaefer, P. Gieselmann, T. Schaaf, A. Waibel, "A pattern learning approach to question answering within the Ephyra framework, " Proceedings of the Ninth International Conference on Text, Speech and Dialog. Springer-Verlag Berlin, Heidelberg, pp. 687-694, 2006.

[13] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, C. Welty, "Building Watson: An Overview of the DeepQA Project, " AI Mag., vol. 31, no. 3, pp. 59-79, 2010.

[14] H. Kamp, Discourse Representation Theory, Netherlands: Springer, 2011.

[15] D. Marcu, "The Rhetorical Parsing of Natural Language Texts, " 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics. USA, pp. 96-103, 1997.

[16] E. H. Hovy, U. Hermjakob, D. Ravichandran, "A Question/Answer Typology with Surface Text Patterns, " 2nd International conference on Human Language Technology Research. USA, pp. 247-251, 2002.

[17] Apache Lucene, Apache Lucene core, [ONLINE] Available at https://lucene.apache.org/core [Access 20 October 2015], 2015.

THE EIGHTH INTERNATIONAL SYMPOSIUM ON INFORMATION AND COMMUNICATION TECHNOLOGY

ISBN: 978-1-4503-5328-1

SoICT 2017

Nha Trang, December 7-8, 2017

Towards Building Vietnamese Discourse Treebank*

Chinh Trong Nguyen
Faculty of Computer Science
University of Information Technology
VNU-HCM
Vietnam
chinhnt@uit.edu.vn

Dang Tuan Nguyen
Faculty of Computer Science
University of Information Technology,
VNU-HCM
Vietnam
dangnt@uit.edu.vn

ABSTRACT

Discourse analysis is an important natural language processing task. There are many discourse parsers in many languages, such as English and Chinese, constructing discourse trees from text documents for further semantic analysis. However, there is no official release of Vietnamese discourse treebank for research in Vietnamese discourse parser. Therefore, this paper presents our preliminary result in building Vietnamese discourse treebank, some problems when building discourse treebank and proposes a discourse annotation framework for it. In order to show the feasibility of developing discourse parsers for Vietnamese documents, two experiments in discourse relation classification and in discourse nucleus classification are conducted using the discourse annotated documents.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; *Language resources* • **Information Systems** → Clustering and classification

KEYWORDS

Vietnamese discourse treebank, rhetorical structure theory, discourse relation classification, discourse nucleus classification.

ACM Reference Format:

Chinh Trong Nguyen, Dang Tuan Nguyen. 2017. Towards building Vietnamese discourse treebank. In *SoICT '17: Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam. ACM, New York, NY, USA,* 7 pages. *https://doi.org/10.1145/3155133.3155200*

SoICT '17, December 7–8, 2017, Nha Trang City, Viet Nam

ACM ISBN 978-1-4503-5328-1/17/12...$15.00
https://doi.org/10.1145/3155133.3155200

1 INTRODUCTION

Discourse structure of a document is important to analyze the meaning which the writer intends to show in the document. In discourse structure, every clause or every simple sentence is not separate but it has a relationship with others to lead the reader through the entire document so that he can understand the writer's intention. Therefore, many semantic related tasks, such as summarization and why-question answering, may need the discourse structure of a document to improve their result.

In text summarization, the problem to be solved is to identify a text fragment that has the same meaning to the origin text and it must be shorter than the origin. Assuming that there are a text T_1 about sea butterflies in Vietnamese for summarizing as following

> T_1: *"Không giống những động vật thân mềm khác, bướm biển có cơ chế di chuyển phức tạp. Để phân tích chuyển động của bướm biển, các nhà khoa học dùng 4 máy quay tốc độ cao và laser hồng ngoại ghi lại cách chúng bơi trong bể. Họ còn thả vào nước những hạt nhỏ xíu lấp lánh để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua."* [source: vnexpress.net]
>
> (Unlike other case-body species, sea butterflies have a complex movement behavior. In order to analyze the movement of sea butterflies, the scientists have used 4 high speed infrared laser cameras recording the way they swim in the tank. They have also dropped many tiny twinkle pieces to study the movement of the water flow after sea butterflies swam through)

The summary of the text T_1 may be "*phân tích chuyển động của bướm biển và nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua*" (to analyze the movement of sea butterflies and to study the movement of the water flow after sea butterflies swam through). The question is how to identify this summary by using discourse structure.

In order to summarize text T_1 by using discourse structure, the text will be parsed by using a discourse parser to produce its discourse structure first. Then, the summary will be identified by selecting the most important texts on the discourse structure. The discourse structure of the text T_1 about sea butterflies in RST[6] framework is shown in Fig. 1.

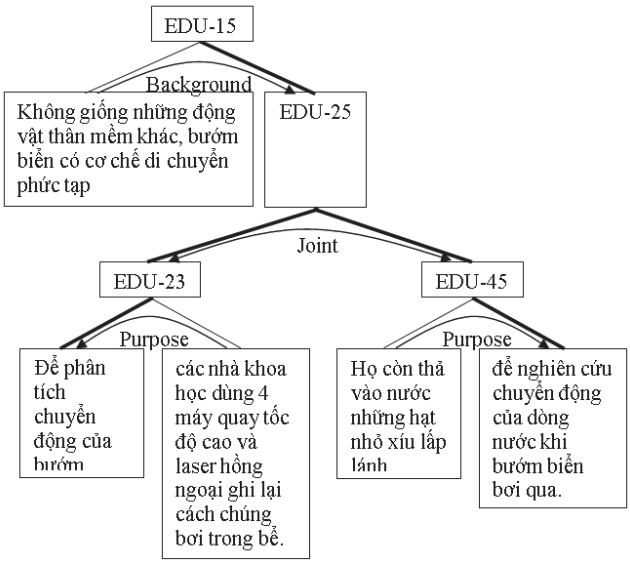


Figure 1: Discourse structure of the text about sea butterflies.

In Fig. 1, there are five sentences and clauses connected to form three mediate units (EDU-23, EDU-45 and EDU-25) and the original text (EDU-15). The sentences and clauses are 1) "*Không giống những động vật thân mềm khác, bướm biển có cơ chế di chuyển phức tạp*" (Unlike other case-body species, sea butterflies have a complex movement behavior), 2) "*Để phân tích chuyển động của bướm biển*" (In order to analyze the movement of sea butterflies), 3) "*các nhà khoa học dùng 4 máy quay tốc độ cao và laser hồng ngoại ghi lại cách chúng bơi trong bể*" (the scientists have used 4 high speed infrared laser cameras recording the way they swim in the tank), 4) "*Họ còn thả vào nước những hạt nhỏ xíu lấp lánh*" (They have also dropped many tiny twinkle pieces), and 5) "*để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua*" (to study the movement of the water flow after sea butterflies swam through). Each sentence, clause or mediate unit connects to others by a discourse relation (Purpose, Joint or Background). Each relation is presented by one-head arrows or two-head arrows that the head of arrow indicates which text is salient in the relations. The salient text is called nucleus. When traveling on the discourse structure, the EDU-25 is salient to form the origin text (EDU-15), then EDU-23 and EDU-45 are salient to form EDU-25, then the clause *" Để phân tích chuyển động của bướm biển"* (In order to analyze the movement of sea butterflies) is salient to form EDU-23 and the clause "*để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua"* (to study the movement of the water flow after sea butterflies swam through) is salient to form EDU-45. Therefore, the important text of the origin is identified by combining salient sentences and clauses that is "*Để phân tích chuyển động của bướm biển và để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua*" (In order to analyze the movement of sea butterflies and to study the movement of the water flow after sea butterflies swam through). This text is very close to the expecting summary.

In why-question answering, the problem is how to identify the reason of an event or a behavior which is asked in a why-question. This problem can be solved by combining information retrieval and information extraction methods using cue phrases of causal relations in answer scorer as in Verberne's research work [7]. However, there is another way to answer why-questions by using discourse structure. Assuming that there is a question "*tại sao họ còn thả những hạt nhỏ xíu lấp lánh?*" (why have they dropped many tiny twinkle pieces?) to the text T_1, the answer will be easily identified by using discourse structure in Fig. 1. Because the main content of the question which is "*họ còn thả những hạt nhỏ xíu lấp lánh* " (they have dropped many tiny twinkle pieces) connects to the clause "*để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua*" (to study the movement of the water flow after sea butterflies swam through) by Purpose relation. In this relation, the latter clause is salient. Therefore, the answer of the question may be "*để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua*" (to study the movement of the water flow after sea butterflies swam through).

Text summarization and why-question answering are just two of many applications using discourse structure of document to improve their results in natural language processing. Therefore, many discourse parsing methods have been studied. These methods can be divided into two types which are rule based parsing, such as Marcu's method [15], and machine learning based parsing, such as research work of Feng [1-2], Hernault [3], Lin [4] and Ghosh [5]. Marcu's method mainly uses cue phrases (discourse markers) and decision rules to identify the discourse relation in English. This rule based approach requires high cost when defining rules and the rules have to be modified whenever new features needed to be added. The methods of Hernault, Feng and Ghosh identifies the discourse relation in English by using SVM or CRF classifiers. These classifiers are easy to be modified or changed when adding or removing features for separating discourse relations. Therefore, the discourse parsers may have many advantages if they are built with machine learning approach. However, a good discourse parser of this type needs large and good discourse annotated corpora for training. Thus, building discourse treebank is an important work.

In Vietnamese, there is no official release of discourse annotated corpora for building Vietnamese discourse parsers. Thus, this paper presents the preliminary study in building Vietnamese discourse treebank with some problems of discourse unit segmentation, discourse annotating representation and format, and some experiments in discourse relation classification for Vietnamese document by using the annotated result when applying the proposed framework in building Vietnamese discourse treebank for further research.

This paper contains five sections. Section 1 introduces the necessary of building Vietnamese discourse treebank for natural language processing researches. Section 2 briefly presents some results of building treebanks in other languages to address the problems in building discourse treebank. Section 3 proposes a solution of building Vietnamese discourse treebank. Some preliminary evaluations on discourse relation identification are

presented in section 4 to show the feasibility of a Vietnamese discourse treebank built as the proposed solution. Then, section 5 presents some conclusions and future works.

2 RELATED WORKS

Many discourse treebanks have been built in English [9-10], Chinese [11-12], Turkish [13] and Arabic [14] because of the wide range application of discourse annotated corpora in computational linguistics [8]. These corpora can be used in language generation, discourse parsing, summarization, argumentation and machine translation. In order to build these corpora, the discourse annotation framework, the annotation scheme and the annotation file format are very important for annotators.

2.1 Discourse Annotation Framework

The above treebanks have been built upon the ground of Rhetorical Structure Theory (RST) [6]. According to RST, a document can be represented as a tree, called discourse tree, in which the leaves are Elementary Discourse Units (EDU), the internal nodes are contiguous text spans of the document and the links between tree nodes are rhetorical relations (or discourse relations).

An EDU is a smallest discourse unit which can have a discourse relation with another adjacent discourse unit (EDU or text span) to form a larger discourse unit. The EDU may be differently identified in different frameworks however it is usually a clause. In Fig. 1, there are three sentences in which the first sentence is an EDU, the next two sentences contain two EDUs each.

A text span is a discourse unit consisting two or more adjacent and non-overlapping EDUs or smaller text spans combining together with a discourse relation. A text span can be the whole document or a group of contiguous text spans in the document. In Fig. 1, EDU-15, EDU-25, EDU-23 and EDU-45 are text spans built from EDUs and smaller text spans.

A rhetorical relation indicates the meaning of the combination of discourse units. A rhetorical relation is chosen according to the intention of the writer. A rhetorical relation can be mono-nuclear or multi-nuclear. A mono-nuclear relation consists discourse units in which one of them is more salient than the others. In a multi-nuclear relation, all discourse units are salient. A salient unit means that it provides important information to the discourse unit directly containing it, thus it may be used as the summary. Salient unit and non-salient unit are called nucleus and satellite respectively. For example, in Fig. 1 the discourse unit EDU-23 contains two EDUs linking together by a Purpose relation. Purpose relation is mono-nuclear thus the left most discourse unit of EDU-23 is the nucleus and the other is satellite. An example of multi-nuclear relation is the Joint relation in EDU-25 in Fig. 1. In EDU-25, all two discourse units are nuclei.

Rhetorical relations can be differently defined according to the point of view of the linguists. In the introduction to RST, Mann[6] proposed 24 relations to analyze the discourse structure. In Carlson's work on building discourse-tagged corpus [9], he used 78 relations in which 53 relations are mono-nuclear and 25 relations are multi-nuclear. These relations are divided into 16 classes. In building Penn Discourse Treebank (PDTB), Miltsakaki[10] used a hierarchy of discourse sense tags (discourse relations) which consists of three levels. The highest level called "class level" which contains 4 classes, the lower level called "type level" which contains 16 types and the lowest level called "subtype level" which contains 22 subtypes. In this hierarchy, there are two types which do not contains any subtypes. There are, at most, 24 discourse relations can be used to tag in PDTB. In constructing Chinese discourse treebank, Zhou [11-12] adapted from PDTB and used the same discourse relations of PDTB. As did Chinese discourse treebank, Turkish Discourse Bank [13] also adapted from PDTB.

2.2 Discourse Annotation Scheme

The discourse treebanks mentioned above mostly adapted from PDTB which has been annotated through a two phases process. These phases are EDU segmentation and relation (sense tag) labeling.

In EDU segmentation phase, every document is split into clauses and sentences which are EDUs. According to Carlson's annotation scheme [9], a clause which is a subject, an object or a complement of the main verb in a sentence are not an EDU. In some cases, a relative clause or a nominal post-modifier, which is an embedded discourse unit, is not necessary to be broken up into a new EDU. In this phase, the boundary of EDUs can be identified by punctuation marks, connective words or phrases, or syntactic clues.

In relation labeling phase, each pair of adjacent discourse units is checked if there is a discourse relation between discourse units and which of the discourse units is nucleus. As in RST, some discourse relations, such as Sequence relation, may hold between two or more discourse units. These multi-nucleus relations are quite difficult to annotate discourse relations of documents and then to learn parameters of classifiers because the discourse relations are annotated differently. Therefore, the annotating framework of PDTB requires that all discourse relations always hold between two discourse units. For the multi-nucleus discourse relations, which hold on more than two units, are broken up into many relations which hold between only two units. For a sample text T_2 contains Sequence relation as follows:

T_2: *"First, place the soy sauce, olive oil, lemon juice, Worcestershire sauce, garlic powder, basil, parsley, and pepper in a blender. Then, add hot pepper sauce and garlic, if desired. After that, blend on high speed for 30 seconds until thoroughly mixed."*

T_2 is modified by adding discourse markers from the following text so that the discourse relation in T_2 is explicit

"Place the soy sauce, olive oil, lemon juice, Worcestershire sauce, garlic powder, basil, parsley, and pepper in a blender. Add hot pepper sauce and garlic, if desired. Blend on high speed for 30 seconds until thoroughly mixed." [Source: http://allrecipes.com]

The RST style discourse tree of the text T_2 is shown as Fig. 2 in which the EDU-13 is the entire text T_2 and three sentences are EDUs linked together by a sequence relation. In order to annotate this text as in PDTB, its discourse tree has to be transformed as in Fig. 3.

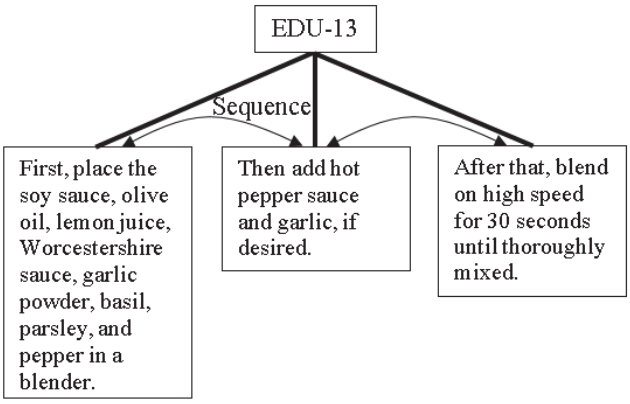


Figure 2: The discourse tree of the text T_2 in RST.

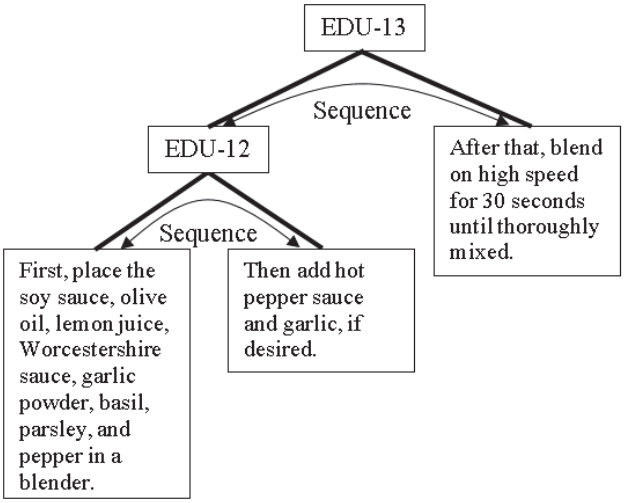


Figure 3: The discourse tree of the text T_2 as in PDTB.

A discourse relation between two discourse units is identified by two aspects which are syntactic connectives and semantic inference. In English, syntactic connectives are coordinating conjunctions, subordinating conjunctions, sentential conjunctions and adverbial conjunctions. The discourse relations identified by syntactic connectives are called explicit discourse relations. When there are no syntactic connectives, the discourse relations have to be identified by the semantic of the discourse units and these relations are called implicit discourse relations. Assuming there is a text T_3 as follow

T_3: *"It's colder and colder therefore Tom has to wear a scarf"*

In T_3, the connective *"therefore"* has split the text into two EDUs (bounded by square brackets) and the two EDUs are connected by Result relation. This relation is explicit because it was identified by the connective "therefore". Assuming the meaning of the text T_3 can be written as the following T_4.

T_4: *"It's colder and colder. Tom has to wear a scarf"*

In T_4, the two EDUs are split by a period and the discourse relation between them is not explicitly identified. If the semantic of these EDUs are considered, it is clear that the behavior *"wear a scarf"* is the result of the cold weather. Therefore, the result relation may be identified implicitly.

After identifying the relation between two discourse units, the salient units of the relation have to be identified as in RST. However, PDTB has used arguments of the relation, which are arg_1 and arg_2, instead of nucleus and satellites [10]. The arg_2 and arg_1 are not nucleus and satellite respectively. In PDTB, arg_2 is the discourse unit containing the syntactic connective and arg_1 is the other. In addition, the nucleus of the relation is identified by the sense of the relation (the sense of the connective in case of explicit relations). If there is no connective between discourse units, the semantic of these units are considered for argument identification. For example, in the text T_3 the connective word *"therefore"* syntactically belongs to the second clause thus the second clause is the arg_2 of the result relation and the first clause is the arg_1. The second clause is also nucleus because of the sense of connective word *"therefore"*.

The arg_1 and arg_2 are important to be syntactically identified in relation labeling phase when building PDTB. However, these arguments are meaningless in Chinese as Zhuo reported [12]. Therefore, the nucleus and satellite units are really important instead of arguments because they are identified by considering the semantic of the discourse relations.

2.3 Annotation File Format

The PDTB file contains many text blocks. Each block represents a discourse relation with five sections which are relation type, sup_1, arg_1, sup_2 and arg_2. Each section has two important fields, the SpanList and GornAddressList which are used to identify the origin text of the relation's parts, and many other fields. With this file format, the annotators have to read the discourse tree by using a discourse annotation tool.

3 BUILDING VIETNAMESE DISCOURSE TREEBANK

Vietnamese discourse treebank (VDTB) is really needed in order to automatically analyze the discourse structure of text document for many NLP tasks, especially for non-factoid question answering and text summarization, in Vietnamese. Hence, Vietnamese discourse treebank has been building similarly to PDTB has. However, some aspects of VDTB are different from PDTB for easily processing.

3.1 Discourse Annotation Framework

Vietnamese discourse treebank has been building according to RST framework because of the simplicity and fully capturing the semantic and textual features of document. That means each document has to be broken up into EDUs, which are adjacent

and non-overlap text spans, and then each discourse unit will be constructed from exact two EDUs or larger discourse units and an appropriate discourse relation. In this framework, the discourse relations used in annotating process are 24 discourse relations proposed by Mann [4]. These relations can be converted into PDTB relations easily.

In discourse relation labeling, the arg_1 and arg_2 are not used in VDTB as in PTDB because the arg_1 or arg_2 are not indications of the nucleus without knowing the semantic of the connective words or phrases between two discourse units. In VDTB, each relation has two versions of representation which are used to indicate that the nucleus is the first or the second discourse unit of the relation. For example, there are discourse relations of cause as following texts T_5 and T_6

T_5: *"She will not arrive because she has a lot of things to do"*

T_6: *"She has a lot of things to do. That is why she will not arrive"*

T_5 and T_6 may be annotated in VDTB as following

T'_5: *[She will not arrive] CAUSE_SN [because she has a lot of things to do]*

T'_6: *[She has a lot of things to do] CAUSE_NS [That is why she will not arrive]*

This type of annotation shows which discourse unit of the relation is nucleus without knowing the sense of the connective word or phrase by using the notation NS or SN. If a relation label ends with NS or SN, the nucleus of the relation will be the first or second discourse unit respectively. If the text T_4 and T_5 are annotated in PDTB, the result may be as follow:

T''_5: *[She will not arrive]^{arg1} <u>because</u> [she has a lot of things to do]^{arg2}(CONTINGENCY:Cause:reason)*

T''_6: *[She has a lot of things to do.]^{arg1} <u>That is why</u> [she will not arrive] ^{arg2}(CONTINGENCY:Cause:reason)*

T''_5 and T''_6 show that there is no way to identify the nucleus of a discourse relation directly using only annotated result.

3.2 Discourse Annotation Scheme

The annotation scheme of VDTB, which is adapted from of PDTB mentioned above, has two phases which are EDU segmentation and relation labeling.

In EDU segmentation phase, relative clauses and nominal post-modifier clauses are not broken up into EDUs because these EDUs are only treated as embedded discourse units that they cannot be used in any of 24 discourse relations proposed by Mann. This is a difference from EDU segmentation of PDTB.

In relation labeling, the discourse relations in VDTB are not divided into explicit and implicit as in PDTB. When assigning a label to a relation which holds between two adjacent and non-overlapped discourse units, the meaning of these units, instead of the connective words or phrases, are used to identify the appropriate discourse relation and which unit is the nucleus because there are many discourse relations hold between many pairs of discourse unit without any connectives. Then, the name of the appropriate relation with a suitable suffix (NS or SN) is used to label the relation. However, the connective words or phrases, if existing, are also taken note into a connective list for further processing in machine learning based classifiers of discourse relation or discourse parsers.

After identifying the EDU segmentation and relation labeling phases, the discourse annotation scheme is proposed as a bottom up process. In this process, the discourse relations are divided into three types that are inner-sentential relation, inter-sentential relation and inter-paragraph relation. An inner-sentential relation holds between discourse units of the same sentence; an inter-sentential relation holds between discourse units of which the text spans are one or more sentences in the same paragraph; and an inter-paragraph relation holds between discourse units of which the text spans are one or more paragraphs. Three classes of discourse relation are proposed because the observed data during annotating discourse structure of text news shows that the discourse relations usually hold at each level of inner-sentential, inter-sentential and inter-paragraph. This means that if there is a discourse relation holds between a clause of a sentence and another sentence, this relation can hold between the sentence consisting the clause and the relative sentence.

The process of annotating the discourse structure of a document is a five-step process as following

1. Reading entire document to capture the meaning of it and the intention of the writer.
2. Breaking up the document into EDU.
3. Labeling inner-sentential discourse relations.
4. Labeling inter-sentential discourse relations.
5. Labeling inter-paragraph discourse relations.

3.3 Annotation File Format

In VDTB, each document is annotated in a text file in which the number of paragraphs is reserved in order to easily separate three levels of discourse relation as mentioned above. The conventions used in annotating are:

1. Square brackets [] are used to indicate the boundary of a discourse unit.
2. Brackets {} are used to indicate the boundary of a discourse relation in which the relation name following by notation of nucleus position is placed first and two discourse units are placed next in the same order as in original text.
3. Each punctuation mark between two discourse units belongs to the first discourse unit.
4. Each connective word or phrase between two discourse units belongs to the second discourse unit.

For example, the annotated result of the sample text T_1 in VDTB is T'_1 as following:

T'_1: ***[{BACKGROUND_SN** *[Không giống những động vật thân mềm khác, bướm biển có cơ chế di chuyển phức tạp.]* ***[{JOINT [{PURPOSE_NS** *[Để phân tích chuyển động của bướm biển,] [các nhà khoa học dùng 4 máy quay tốc độ cao và laser hồng ngoại ghi lại cách chúng bơi trong bể.]}]* ***[{PURPOSE_SN** *[Họ còn thả vào nước những hạt nhỏ xíu lấp lánh] [để nghiên cứu chuyển động của dòng nước khi bướm biển bơi qua.]}]}]}]*

Documents annotated by using this format are easily converted into appropriate format for training EDU segmenters or displayed in tree style like syntax tree for both phrase structure and dependency type. In addition, it is easy to extract inter-paragraph discourse relations for further research on discourse analysis on inter-paragraph level.

4 PRELIMINARY RESULTS

Before constructing an official VDTB it is important to make some experiments to show the feasibility of VDTB because constructing VDTB is an expensive job. At this time, the documents being annotated are chosen from Vietnamese scientific news from the internet because scientific news is usually small and focuses on a certain subject that it is simpler to manually analyze the discourse structure than other types of document, such as essay or thesis.

4.1 Discourse Annotation Results

At this time, there are 921 discourse relations have been annotated in VDTB. These relations are divided into three levels which are inner-sentential, inter-sentential and inter-paragraph. The details of annotated discourse relation are shown in Table 1 in which relations with asterisk are multi-nucleus relations and each relation has frequencies by relation levels and position of nucleus.

Table 1: Frequency of Discourse Relation by Relation Level and Position of Nucleus

Relation	Inner-sen.		Inter-sen.		Inter-para.	
	NS	SN	NS	SN	NS	SN
Background	0	1	1	24	2	17
Circumstance	32	139	18	9	1	0
Concession	3	36	0	13	0	4
Condition	2	10	0	1	0	0
Contrast*	5		5		0	
Elaboration	0	0	3	0	10	0
Evaluation	1	0	2	2	4	0
Evidence	4	0	0	0	11	0
Interpretation	12	1	150	1	127	0
Joint*	12		37		39	
Non-volitional cause	4	13	1	4	0	0
Non-volitional result	7	37	0	10	0	0
Purpose	14	8	0	2	0	0
Restatement	1	0	1	0	1	0
Sequence*	5		13		4	
Solution hood	0	0	0	1	0	0
Summary	0	0	0	1	31	3
Volitional cause	1	1	0	0	0	0
Volitional result	0	3	0	0	0	0
Total	**352**		**315**		**254**	

In VDTB, there 352 relations at inner-sentential level, 315 relations at inter-sentential level and 254 relations at inter-paragraph level. The total number of each kind of discourse relation are very different. Some discourse relations which have been used frequently in the chosen scientific news are Circumstance, Interpretation, Joint, Non-volitional result, Concession, Purpose and Non-volitional cause. Among the relation kinds, the Interpretation, Background and Joint relation mostly did not have any connective word or phrase. Therefore, the meaning of two discourse units must be used to identify them. Some relation kinds which usually have connective word or phrase are Non-volitional cause, Non-volitional result, Purpose, Volitional cause and Volitional result. However, it is difficult to differentiate Non-volitional cause from Volitional cause or Non-volitional result from Volitional result only by connective word or phrase. Therefore, connective words or phrases are just supporting features when identifying discourse relations.

4.2 Discourse Relation Classification Results

Discourse relation classification is an important task of discourse parser. The discourse parsers described in [2,3] are two-phase processes. The first phase is EDU segmentation based on a sequential labeling method such as CRF. Then, in the second phase each pair of consecutive discourse units has been check if there is a discourse relation hold between its discourse units by using a classifier. Therefore, two experiments in classification of discourse relations in VDTB have been conducted.

In all experiments, SVM and J48 method are used with the data of 921 discourse relations. Then, 4-fold cross validation is used for evaluation. Weka[1] is used in these experiments after translating each discourse relation into feature vector. There are four types of features that are connective, connective position in each discourse unit, the number of common words of two discourse units and the similarity of text structure of two discourse units. In this case, the text structure is simply the order of POS chain of the text. This feature is used because the Sequence relation may use similar structure in each discourse unit.

The first experiment is full discourse relation classification in which the exact relation with the position of nucleus is required for each classification result. In this experiment, there are only 7 discourse relation kinds are chosen for evaluation because there are few number of other discourse relation kinds in VDTB that they cannot be used in experiment. The experiment results are shown in Table 2.

Table 2 shows the precision (P), recall (R) and f-measure (F) of classifying each kind of discourse relation by each classification method. In Table 2, the precisions are very low in Background, Joint and Non-volitional result relation because Background and Joint relations do not have any marker as mentioned above and Non-volitional result relations are just different from Volitional result relations in meaning. In other relation kinds, the precisions are from 0.5 to 0.69 because the relation of these kinds usually have markers and they can be recognized by using these markers. Therefore, semantic analysis should be applied in discourse relation classification in order to improve the result.

[1] http://www.cs.waikato.ac.nz/ml/weka

Table 2: The results of full discourse relation classification

Relation	SVM			J48		
	P	R	F	P	R	F
Background (NS)	.0	.0	.0	.06	.05	.05
Circumstance (NS)	.65	.43	.52	.49	.57	.53
Circumstance (SN)	.5	.72	.59	.56	.66	.61
Interpretation (NS)	.43	.93	.59	.53	.58	.55
Joint	.0	.0	.0	.15	.12	.13
Non-volitional result(SN)	.0	.0	.0	.32	.26	.28
Summary (NS)	.69	.65	.67	.62	.74	.68
Average	**.32**	**.39**	**.34**	**.39**	**.43**	**.4**

The second experiment is discourse nucleus classification. In this experiment, there are only three classes which are NS (nucleus is the first discourse unit), SN (nucleus is the second discourse unit) and NN (both are nuclei). This experiment was conducted to show if VDTB may be useful in text summarization because the importance is that the salient text spans are identified exactly although the discourse relation may be wrongly identified. The results of discourse nucleus classification are shown in Table 3.

Table 3: The results of discourse nucleus classification

Nucleus position	SVM			J48		
	P	R	F	P	R	F
The first unit (NS)	.63	.88	.73	.63	.75	.68
The second unit (SN)	.74	.61	.67	.66	.57	.61
Both of units (NN)	.0	.0	.0	.21	.13	.16
Average	**.46**	**.5**	**.48**	**.5**	**.49**	**.49**

The average precisions shown in Table 3 are low because the amount of multi-nuclear relations (Joint, Sequence, Contrast) is small in VDTB so that the classifiers do not learn the parameters of these relations yet. However, the precisions in classifying mono-nuclear relations are pretty high (0.63 – 0.74) show that the VDTB may be useful in practical use.

5 CONCLUSIONS AND FUTURE WORKS

Discourse analysis is one of many important tasks in natural language processing. Its results, the discourse parses, are very useful in text summarization and question-answering because the semantic of the whole text document has to be analyzed in discourse instead of separate sentences. In order to parse a discourse structure of a document, the efficient way is building machine learning based discourse parsers which require large and good discourse annotated corpora, called discourse treebanks. There are many discourse treebanks in English, Chinese, Turkish, Arabic and more. However, there is no official release of Vietnamese discourse treebank (to our knowledge) which is very useful for research in Vietnamese NLP. Therefore, a discourse annotation framework for Vietnamese discourse treebank (VDTB) has been proposed.

At this time, the VDTB contains 352, 315 and 254 annotated discourse relations respectively at inner-sentential, inter-sentential and inter-paragraph level with 19 discourse relation kinds (label) without nucleus position. These annotated relations have been used in two experiments of discourse relation classification to show the feasibility of VDTB. In two experiments, there are four types of feature, which are connective, position of connective, the number of common words in two discourse units and the similarity of text structure, has been used.

In full discourse relation classification, the results are low in which the average precisions are 0.32 and 0.39 by using SVM and J48 method respectively. However, in classifying other relation kinds, in which connective words or phrases are used, the precisions are pretty good (0.65 for Circumstance NS and 0.69 for Summary NS).

In nucleus classification, the results are also low in which the average precision are 0.46 and 0.5 by using SVM and J48 method respectively. However, the results are pretty high (0.74 in SN and 0.63 in NS by using SVM method). The reason of low average precision is the number of multi-nuclear relations is small in VDTB.

The above results show that VDTB may be practical to be built larger in future. When a large and good VDTB exists, the discourse parsers with semantic features should be studied to analyze the discourse structures of Vietnamese text documents for further NLP tasks.

REFERENCES

[1] V. W. Feng, G. A. Hirst. 2014. Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing. In *ACL* 1 (2014), 511–521.

[2] W. Feng, G. Hirst. 2012. Text-level Discourse Parsing with Rich Linguistic Features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL '12, Jeju Island, Korea (2012), 60–68.

[3] H. Hernault, H. Prendinger and M. Ishizuka. 2010. HILDA: A discourse parser using support vector machine classification. *Dialogue & Discourse* 1, 3 (2010), 1–33. DOI: http://dx.doi.org/10.5087/dad.2010.003

[4] Z. Lin, H.T. Ng, and M.Y. Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering* 20, 2 (2014), 151–184.

[5] Ghosh, S., Johansson, R. and Tonelli, S., 2011. Shallow discourse parsing with conditional random fields. In *Proceedings of the 5th International Joint Conference on Natural Language Processing* (IJCNLP 2011), Chiang Mai, Thailand, 1071–1079

[6] W. C. Mann, S. A. Thompson. 1988. Rhetorical structure theory: towards a functional theory of text organization. *Text* 3, 8 (1988), 243–281

[7] S. Verberne. 2009. *In Search of Why: Developing a system for answering why-questions*. Ph.D. Dissertation. Radboud University, Nijmegen, Germany.

[8] M. Taboada, W. C. Mann. 2006. Applications of rhetorical structure theory. *Discourse studies* 8, 4 (2006), 567–588.

[9] L. Carlson, D. Marcu, M. E. Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. Current and new directions in discourse and dialogue. Springer, Netherlands, 85–112.

[10] E. Miltsakaki, L. Robaldo, A. Lee, and A. Joshi. 2008. Sense annotation in the penn discourse treebank. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, Berlin, Heidelberg, 275–286.

[11] Y. Zhou, N. Xue. 2012. PDTB-style discourse annotation of Chinese text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. ACL 1 (2012), 69–77.

[12] Y. Zhou, N. Xue. 2015. The chinese discourse treebank: a chinese corpus annotated with discourse relations. *Language Resources and Evaluation* 49, 2 (2015), 397–431.

[13] D. Zeyrek, I. Demirsahin, A. Sevdik-Callı, R. Çakıcı. 2013. Turkish Discourse Bank: Porting a discourse annotation style to a morphologically rich language. *Dialog and Discourse* 4, 2 (2013), 174–184.

[14] A. Al-Saif, K. Markert. 2010. The Leeds Arabic Discourse Treebank: Annotating Discourse Connectives for Arabic. In *LREC* (2010).

[15] D. Marcu. 1998. The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Ph.D. Dissertation, University of Toronto, Canada.

Tran Khanh Dang
Josef Küng
Tai M. Chung
Makoto Takizawa (Eds.)

Communications in Computer and Information Science 1500

Future Data and Security Engineering

Big Data, Security and Privacy, Smart City and Industry 4.0 Applications

8th International Conference, FDSE 2021
Virtual Event, November 24–26, 2021
Proceedings

Springer

FDSE 2021

Building a Vietnamese Dataset for Natural Language Inference Models

Chinh Trong Nguyen[1] and Dang Tuan Nguyen[2](✉)

[1] University of Information Technology, VNU-HCM, Ho Chi Minh City, Vietnam
chinhnt@uit.edu.vn
[2] Saigon University, Ho Chi Minh City, Vietnam
dangnt@sgu.edu.vn

Abstract. Natural language inference models are important resources for many natural language understanding applications. These models are possibly built by training or fine-tuning using deep neural network architectures for state-of-the-art results. This means high-quality annotated datasets are important for building state-of-the-art models. Therefore, we propose a method of building Vietnamese dataset for training Vietnamese inference models which work on native Vietnamese texts. Our method aims at two issues: removing cue marks and ensuring the writing-style of Vietnamese texts. If a dataset contains cue marks, the trained models will identify the relation between a premise and a hypothesis without semantic computation. For evaluation, we fine-tuned a BERT model on our dataset and compared it to a BERT model which was fine-tuned on XNLI dataset. The model which was fine-tuned on our dataset has the accuracy of 86.05% while the other has the accuracy of 64.04% when testing on our Vietnamese test set. This means our method is possibly used for building a high-quality Vietnamese natural language inference dataset.

Keywords: Natural language inference · Textual entailment · NLI dataset · Transfer learning

1 Introduction

Natural language inference (NLI) research aims at identifying whether a text ***p***, called the premise, implies a text ***h***, called the hypothesis, in natural language. NLI is an important problem in natural language understanding (NLU). It is possibly applied in question answering [1–3] and summarization systems[4, 5]. NLI was early introduced as RTE [6] (Recognizing Textual Entailment). The early researches in RTE were divided in two different approaches [6] similarity-based and proof-based. In similarity-based approach, the premise and the hypothesis are parsed into representation structures, such as syntactic dependency parses, then a similarity is computed on these representations. In general, the high similarity of the premise-hypothesis pair means there is an entailment relation. However, there are many cases that the similarity of the premise-hypothesis pair is high but there is no entailment relation. The similarity is possibly defined as a

T. K. Dang et al. (Eds.): FDSE 2021, CCIS 1500, pp. 185–199, 2021.
https://doi.org/10.1007/978-981-16-8062-5_12

handcraft heuristic function, or an edit-distance based measure. In proof-based approach, the premise and the hypothesis are translated into formal logic then the entailment relation is identified by a proving process. This approach has an obstacle of translating a sentence into formal logic which is a complex problem.

Recently, NLI problem has been studied on classification-based approach thus deep neural networks are effective for solving this problem. The release of BERT architecture [7] showed many impressive results of improving benchmarks in many NLP tasks including NLI. When using BERT architecture, we will save many efforts in creating lexicon semantic resources, parsing sentences into appropriate representation, and defining similarity measures or proving schemes. The only one problem when using BERT architecture is the high-quality training dataset for NLI. Therefore, many RTE or NLI datasets have been released for years. In 2014, SICK [8] was released with 10k English sentence pairs for RTE evaluation. SNLI [9] has the similar format of SICK with 570k pairs of text span in English. In SNLI dataset, the premises and the hypotheses may be sentences or groups of sentences. The training and testing results of many models on SNLI dataset was higher than on SICK dataset. Similarly, MultiNLI [10] with 433k English sentence pairs was created by annotating on multi-genre documents for increasing the difficulty of the dataset. For cross-lingual NLI evaluation, XNLI [11] was created by annotating different English documents from SNLI and MultiNLI.

For building Vietnamese NLI dataset, we may use machine translator for translating the above datasets into Vietnamese. Some Vietnamese NLI (RTE) models was created by training or fine-tuning on Vietnamese translated versions of English NLI dataset for experiments. The Vietnamese translated version of RTE-3 was used for evaluation of similarity-based RTE in Vietnamese [12]. When evaluating PhoBERT in NLI task [13], the Vietnamese translated version of MultiNLI was used for fine-tuning. Although we can use machine translator for automatically building Vietnamese NLI dataset, we should build our Vietnamese NLI datasets for two reasons. The first reason is that some existing NLI datasets contain cue marks which was used for entailment relation identification without considering the premises [14]. The second reason is that the translated texts may not ensure the Vietnamese writing style or may return weird sentences.

In this paper, we would like to propose our method of building a Vietnamese NLI dataset which is annotated from Vietnamese news for ensuring writing style and contains more "*contradiction*" samples for removing cue marks. When proposing our method, we would like to reduce the annotation cost by using entailment sentence pairs existing in news webpages. We present this paper in five sections. Section 1 introduces the demand of building Vietnamese NLI dataset for building Vietnamese NLI models. Section 2 presents our proposed method of building Vietnamese NLI dataset. Section 3 presents the process of building Vietnamese NLI dataset and some experiments. Section 4 presents some experiments on our dataset in Vietnamese NLI. Then, some conclusions and our future works are presented in Sect. 5.

2 The Constructing Method

Our approach in building Vietnamese NLI dataset is generating samples from existing entailment pairs. These entailment pairs will be crawled from Vietnamese news websites for saving annotation cost, ensuring writing style and multi-genre.

2.1 NLI Sample Generation

The first requirement about our NLI dataset is that it does not contain cue marks. If a dataset contains these marks, the model trained on this dataset will identify "*contradiction*" and "*entailment*" relations without considering the premises or hypotheses [14]. Therefore, we will generate samples in which the premise and the hypothesis have many common words while their relation varies. We used some logic implication rules for this generation task. Given A and B are propositions, we will have the relations of eight premise-hypothesis types as shown in Table 1.

We used premise-hypothesis types 1 to 4 for removing the cues marks. When training a model, the model will learn from samples of types 1 to 4 the ability of recognizing the same sentences and contradiction sentences. We also used types 5 and 6 for training the ability of recognizing the summarization and paraphrase cases. Type 6 is added in the attempt of removing special marks which can occur when creating type 5 samples. We also added types 7 and 8 for recognizing the contradiction in paraphrase and summarization cases in which the proposition B is the paraphrase or the summary of the proposition A, respectively. Types 7 and 8 are valid only if B is the paraphrase or the summary of A.

Table 1. The relations of premise-hypothesis types used for building supplement dataset.

Type	Condition	P	H	Relation
1		A	A	entailment
2		¬A	¬A	entailment
3		A	¬A	contradiction
4		¬A	A	contradiction
5	A⇒B	A	B	entailment
6	A⇒B	¬B	¬A	entailment
7	A⇒B	A	¬B	contradiction*
8	A⇒B	¬A	B	contradiction*

In general, the types 7 and 8 cannot be applied in cases where the proposition A implies the proposition B by using presuppositions. For example, assuming A is the proposition "*we are hungry*", B is the proposition "*we will have lunch*" and A⇒B is the valid proposition "*if we are hungry then we will have lunch*" because we have two presuppositions that we should eat when we are hungry and we eat when we have lunch. We see that ¬B, which is the proposition "*we will not have lunch*", is not the contradiction of the proposition A.

2.2 Entailment Pair Collection

Entailment pairs exist in text documents, but it is difficult to extract them from the text documents. Therefore, after considering many news posts on many Vietnamese news

websites such as, VnExpress[1], we found that the title is usually the paraphrase or the summary of the introductory sentence in a news post. We can divide the news posts into four types. In type 1, the title is the paraphrase of the introductory sentence in the news post. In the example shown in Fig. 1, the title "*Nhiều tài xế dừng xe đậy nắp cống suốt 10 ngày*" (in English: "*many drivers was stopping to close the drain cover in 10 days*") is a paraphrase of the introductory sentence "*Nhiều tài xế dừng ôtô giữa ngã tư để đậy lại miệng cống hở do chiếc nắp cong vênh và câu chuyện diễn ra suốt 10 ngày ở Volgograd*" (in English: "*Many drivers was stopping the cars at the crossroad to close the slightly opened drain cover because the drain cover was bent*").

Fig. 1. An example of type-1 news post from vnexpress.net website

In type 2, the title is the summary of the introductory sentence in the news post. In the example shown in Fig. 2, the title "*Gạo chữa nhiều bệnh*" (in English: "*rice used for curing many diseases*") is the summary of the introductory sentence "Gạo nếp và gạo tẻ đều có vị thơm ngon, mềm dẻo, vừa cung cấp dinh dưỡng, vừa chữa nhiều bệnh như nôn mửa, rối loạn tiêu hóa, sốt cao" (in English: "*Glutinous rice and plain rice, which are delicious and soft when cooked, provide nutrition and are used for curing many diseases such as vomiting, digestive disorders, high fever*").

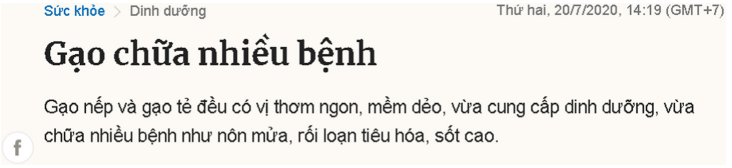


Fig. 2. An example of type-2 news post from vnexpress.net website

In type 3, the title is possibly inferred from the introductory sentence in the news post. Some pre-suppositions are possibly used in this inference. In the example shown in Fig. 3, the title "*Xuất khẩu rau quả tăng mạnh*" (in English: "*Vegetable export increases significantly*") can be inferred from the introductory sentence "*Bốn tháng đầu năm nay, giá trị xuất khẩu rau quả đạt 1,35 tỷ USD, tăng 9,5% so với cùng kỳ năm ngoái.*" (In English: "*in the first four months this year, vegetable export reaches 1.35 billion USD, increases 9.5% in comparison with the same period in last year*"). In this inference, we have used a pre-supposition which defines that increasing 9.5% means increasing significantly in export.

[1] https://vnexpress.net.

Fig. 3. An example of type-3 news post from vnexpress.net website

In type 4, the title is a question which cannot have an entailment relation to the introductory sentence in the news post. In the example shown in Fig. 4, the title, which is a question "*Vì sao giá dầu lao dốc chỉ trong 6 tuần?*" (In English: "*why does the oil price dramatically decreases in 6 weeks only*"), cannot have an entailment relation with the introductory sentence "*Chỉ mới cách đây hơn một tháng, giới buôn dầu còn lo ngại thiếu cung có thể đẩy dầu thô lên 100 USD một thùng.*" (In English: "*just more than one month ago, oil traders still worried that the insufficient supply could increase the oil price by 100 USD per barrel*").

Fig. 4. An example of type-4 news post from vnexpress.net website

We collected only title-introductory sentence pairs of type 1 and type 2 to make entailment pair collection because the pairs of type 3 and 4 cannot be applied 8 relation types when generating NLI samples. The type of a sentence pair is identified manually for high quality. In every pair in our collection, its title is the hypothesis, and its introductory sentence is the premise.

3 Building Vietnamese NLI Dataset

We built our NLI dataset with a three-step process. In the first step, we extracted title-introductory pairs from Vietnamese news websites. In the second step, we manually selected entailment pair and made the contradiction sentences from titles and introductory sentences for high quality. In the third step, we generate NLI samples from entailment pairs automatically and their contradiction sentences by applying 8 relation types shown in **Table 1.**

3.1 Contradiction Creation Guidelines

We made the contraction of a sentence manually for high-quality result. We proposed three types of making the contradiction. These are simple ways to make the contradiction of a sentence using syntactic transformation and lexicon semantic. In the type 1, a given

sentence will be transformed from affirmative to negative or vice versa by adding or removing the negative adverb. If the given sentence is an affirmative sentence, we will add a negative adverb to modifier the main verb of the sentence. If the given sentence is a negative sentence, we will remove the negative adverb which is modifying the main verb of the sentence. The negative adverbs used in our work are "*không"*, "*chưa"* and "*chẳng*" (in English: they mean "*not*" or "*not…yet"*). We used one of these adverbs according to the sentence for ensuring the Vietnamese writing-style. We have four cases of making contradiction with this type.

Case 1 of type 1, making contradiction from an affirmative sentence containing one verb. We will add one negative adverb to modify the verb. For example, making the contradiction of the sentence "*Đài Loan bầu lãnh đạo"* (in English: "*Taiwan voted for a Leader*"), we will add negative adverb "*không*" ("*not*") to modify the main verb "*bầu"* ("*voted*") for making the contradiction "*Đài Loan không bầu lãnh đạo"* (in English: "*Taiwan did not vote for a Leader"*).

Case 2 of type 1, making contradiction from an affirmative sentence containing a main verb and other verbs. We will add one negative adverb to modify the main verb only. For example, making the contradiction of the sentence "*Báo Mỹ đánh giá Việt Nam chống Covid-19 tốt nhất thế giới"* (in English: "*US news reported that Vietnam was the World's best nation in Covid-19 prevention"*), we will only add negative adverb "*không"* to modify the main verb "*đánh giá"* ("*reported"*) for making the contradiction "*Báo Mỹ không đánh giá Việt Nam chống Covid-19 tốt nhất thế giới*" (in English: "*US news did not report that Vietnam was the World's best nation in Covid-19 prevention"*).

Case 3 of type 1, making contradiction from an affirmative sentence containing two or more main verbs. We will add negative adverbs to modify all main verbs. For example, making the contradiction of the sentence "*Bão Irma mang theo mưa lớn và gió mạnh đổ bộ Cuba cuối tuần trước, biến thủ đô Havana như một 'bể bơi khổng lồ*"' (in English: "*Storm Irma brought heavy rain and winds to Cuba last week, making the Capital Havana a 'giant swimming pool"'*), we will add two negative adverbs "*không"* to modify two main verbs "*mang"* and "*biến"* for making the contradiction "*Bão Irma không mang theo mưa lớn và gió mạnh đổ bộ Cuba cuối tuần trước, không biến thủ đô Havana như một" "bể bơi khổng lồ"* (in English: "*Storm Irma did not bring heavy rain and winds to Cuba last week, not making the Capital Havana a 'giant swimming pool"'*).

Case 4 of type 1, making contradiction from a negative sentence containing negative adverbs. We will remove all negative adverbs in the sentence. In our data, we did not see any sentence of this case; however, we put this case in our guidelines for further use.

In the type 2, a given sentence or phrase will be transformed using the structure "*không có …*" (in English: "*there is/are no"*) or "*không … nào …*" (in English: "*no …*"). We have two cases of making contradiction with this type.

Case 1 of type 2, making contradiction from an affirmative sentence by using structure "*không có …*". We use this case when the given sentence has a quantity adjective or a cardinal number modifying the subject of the sentence and it is non-native if we add a negative adverb to modifying the main verb of the sentence. The quantity adjective or cardinal number will be replaced by the phrase "*không có"*. For example, making the contradiction of the sentence "*120 người Việt nhiễm nCoV ở châu Phi sắp về nước"* (in

English: "*120 Vietnamese nCoV-infested people in Africa are going to return home"*), we will replace "120" by "*không có*" because if we add negative adverb "*không*" to modify the main verb "về" ("*return*"), the sentence "*120 người Việt nhiễm nCoV ở châu Phi sắp không về nước*" (in English: "*120 Vietnamese nCoV-infested people in Africa are not going to return home*") sounds non-native. Therefore, the contradiction should be "*không có người Việt nhiễm nCoV ở châu Phi sắp về nước*" (in English: "*no Vietnamese nCoV-infested people in Africa is going to return home*"). Case 1 of type 2 will be used when we are given a phrase instead of a sentence. For example, making the contradiction of the phrase "*trường đào tạo quản gia cho giới siêu giàu Trung Quốc*" (in English: "*the butler training school for Chinese super-rich class*"), we will add the phrase "*không có*" at the beginning of the phrase to make the contradiction "*không có trường đào tạo quản gia cho giới siêu giàu Trung Quốc*" (in English: "*there is no butler training school for Chinese super-rich class*").

Case 2 of type 2, making contradiction from an affirmative sentence by using structure "*không …nào …*". We will use this structure when we have case 1 of type 2 but the generated result of that case is not native. For example, making the contradiction of the sentence "*gần ba triệu ngôi nhà tại Mỹ mất điện vì bão Irma*" (in English: "*nearly three million houses in U.S. were without power because of Irma storm*"), if we replace "*gần ba triệu*" (in English: "*nearly three million*") by "*không có*", we will have a non-native sentence "*không có ngôi nhà tại Mỹ mất điện vì bão Irma*" therefore we should use the structure "*không … nào …*" to make the contradiction "*không ngôi nhà nào tại Mỹ mất điện vì bão Irma*" (in English: "*There are no houses in U.S. were without power because of Irma storm*").

In type 3, a contradiction sentence is generated using lexicon semantic. A word of the given sentence will be replaced by its antonym. This way will make the contradiction of the given sentence. Although we can use all cases of type 1 and type 2 for making the contradiction, we still recommend this type because the samples generated with this type may help the fine-tuned models to learn more about antonymy. We have two cases of making contradiction with this type.

Case 1 of type 3, making contradiction from a sentence by replacing the main verb of the sentence with its antonym. For example, making the contradiction of the sentence "*Mỹ thêm gần 18.000 ca nCoV một ngày*" (in English: "*the number of nCoV cases in U.S. increases about 18.000 in one day*"), we can replace the main verb "*thêm*" ("*increase*") by its antonym "*giảm*" ("*decrease*") to make the contradiction "*Mỹ giảm gần 18.000 ca nCoV một ngày*" (in English: "*the number of nCoV cases in U.S. decreases about 18.000 in one day*").

Case 2 of type 3, making contradiction from a given sentence by replacing an adverb or a phrase modifying the main verb by the antonym or the contradiction of that adverb or that phrase, respectively. We use this case when we need to make the samples containing antonymy, but the main verb does not have any antonyms because there are many verbs which do not have their antonym. For example, making the contradiction of the sentence "*Mỹ viện trợ nhỏ giọt chống Covid-19*" (in English: "*the U.S. aided a little in Covid-19 prevention*"), we cannot replace the main verb "*viện trợ*" ("*aid*") with its antonym because it does not have an antonym. Therefore, we will replace "*nhỏ giọt*" ("*a little*") by "*ào ạt*" ("*a lot*") to make the contradiction "*Mỹ viện trợ ào ạt chống Covid-19*" (in

English: "*the U.S. aided a lot in Covid-19 prevention*"). In this example, "*nhỏ giọt*" and "*ào ạt*" have the opposite meanings; and the phrases "*nhỏ giọt*" and "*ào ạt*" have the adverb role in the sentence when modifying the main verb "*viện trợ*".

3.2 Building Steps

We built our Vietnamese NLI dataset follow the three-step process which is a semi-automatic process shown in **Fig. 5**.

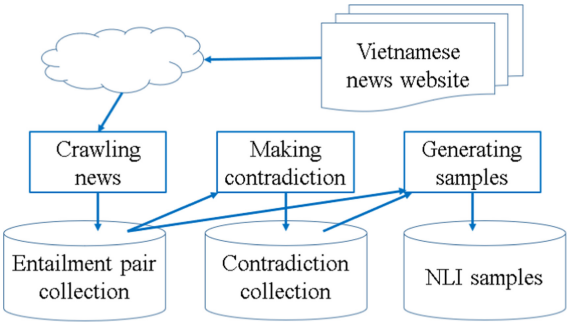


Fig. 5. Our three-step process of building Vietnamese NLI dataset

In the first step – crawling news, we used a crawler to fetch unique webpages from sections of international news, business, life, science, and education in website *vnexpress.net*. Then we extracted their titles and introductory sentences by a website-specific pattern defined with regular expression. The results are sentence pairs stored in an entailment pair collection with unique numbers. These pairs are not always type 1 or 2 therefore the entailment pairs will be manually selected right before making contradiction sentences.

In the second step – making contradiction, we firstly manually identified if each pair of the collection was type 1 or 2 for entailment pair selection. When an entailment pair was selected, we made the contradiction sentences for the title and the introductory sentence using the contradiction creation guidelines. In the entailment pairs, the introductory sentences are the premises, and the titles are the hypotheses. As the results, we have a collection of pairs of sentences ¬A and ¬B stored in contradiction collection in which each sentence pair ¬A and ¬B has a condition A⇒B. In this step, we have two people making contradiction sentences. These people are society science bachelors. Because the guidelines of making contradiction sentence are simple, there are no disagreements in the annotation results.

In the third step – generating samples, we used a computer program implemented from Algorithm 1 for combining the premises, hypotheses stored in entailment pair collection and their contradiction sentences stored in contradiction collection by their unique numbers. The combination rules follow Table 1 in generating NLI samples. For generating "*neutral*" samples, the computer program combined sentences from different premise-hypothesis pairs. In Algorithm 1, the function *getContradict()* return the contradiction sentence stored in contradiction collection. The three functions *ent()*, *neu()*, and

con() are used for creating entailment, neutral and contradiction sample from a premise and a hypothesis, respectively.

Algorithm 1 Generating NLI samples.

```
Input: E, a list of premise-hypothesis pairs.
Output: SD, the NLI sample data with SNLI format.

SD←∅
PL←∅  //premise list
HL←∅  //hypothesis list
cPL←∅ //premise contradiction list
nHL←∅ //hypothesis contradiction list
for i←1 to |E|
   prem ← E[i].premise
   hyp ← E[i].hypothesis
   nprem ← genContradict(prem)
   nhyp ← genContradict(hyp)
   if nprem = NULL and nhyp = NULL then
      continue
   end if
   PL←PL + {prem}
   HL←HL+{hyp}
   cPL←nPL+{nprem}
   cHL←nHL+{nhyp}
end for
PL ← PL+{PL[1]}, HL ← HL+{HL[1]}
cPL ← nPL+{nPL[1]}, cHL ← nHL+{nHL[1]}
for i←1 to |PL|-1
   SD ← SD+ent(PL[i],PL[i])+ent(HL[i],HL[i])
          +ent(PL[i],HL[i])+neu(PL[i],PL[i+1])
          +neu(PL[i+1],PL[i])+neu(HL[i],HL[i+1])
          +neu(HL[i+1],HL[i])+neu(PL[i],HL[i+1])
          +neu(PL[i+1],HL[i])+neu(HL[i],PL[i+1])
          +neu(HL[i+1],PL[i])
   if cHL[i] != NULL then
      SD ← SD+con(HL[i],cHL[i])+con(cHL[i],HL[i])
             +ent(cHL[i],cHL[i])
      if cHL[i+1] != NULL then
```

```
26            SD ← SD+neu(PL[i],cHL[i])+neu(cHL[i],cHL[i+1])
                  +neu(cHL[i+1],PL[i])+neu(cHL[i+1],cHL[i])
27        end if
28        SD ← SD+neu(PL[i+1],cHL[i])+neu(cHL[i],PL[i+1])
29    end if
30    if cPL[i] != NULL then
31        SD ← SD+con(PL[i],cPL[i])+con(cPL[i],PL[i])
32        SD ← SD+ent(cPL[i],cPL[i])
33        if cPL[i+1] != NULL then
34            SD ← SD+neu(HL[i],cPL[i+1])+neu(cPL[i],cPL[i+1])
                  +neu(cPL[i+1],PL[i])+neu(cPL[i+1],HL[i])
35        end if
36            SD ← SD +neu(HL[i+1],cPL[i])+neu(cPL[i],HL[i+1])
37    end if
38    if cPL[i]!=NULL && cHL[i]!=NULL then
39        SD ← SD+ent(cHL[i+1],cPL[i])
40        if cHL[i+1] != NULL then
41            SD ← SD+neu(cPL[i],cHL[i+1])
                  +neu(cHL[i+1],cPL[i])
42        end if
43        if cPL[i+1] != NULL then
44            SD ← SD+neu(cHL[i],cPL[i+1])+neu(cPL[i+1],cHL[i])
45        end if
46    end if
47 end for
48 return SD
```

3.3 Building Results

In our present NLI dataset, called VnNewsNLI, the rates of making contradiction sentences by applying type 1, type 2 and type 3 are 61.74%, 17.67% and 20.58%, respectively. The rates of entailment, neutral and contradiction samples in our VnNewsNLI dataset are shown in Table 2. In Table 2, the rates of sample types are approximate. Although the rate of neutral samples (30.70%) is lower than of others in development set, the differences in number between these samples are not much therefore the development set is still balanced.

The statistics of the VnNewsNLI dataset by syllable are shown in Table 3. We used syllable as text length unit in Table 3 because there are many multi-lingual pretrained model which were trained on unsegmented Vietnamese text datasets. According to Table 3, the premises and hypotheses are often short (9–14 syllables) and quite long (> 26

syllables) sentences therefore this dataset may provide the characteristic of short and long sentences. There is a difference between the VnNewsNLI dataset and the SNLI dataset that the premises and hypotheses are almost sentences in the VnNewsNLI dataset while they are almost groups of sentences in the SNLI dataset.

Table 2. The statistics of NLI samples in VnNewsNLI dataset

Criterion	Development set		Test set	
	n	%	n	%
Entailment	947	34.74%	4,140	33.42%
Contradiction	942	34.56%	4,128	33.33%
Neutral	837	30.70%	4,118	33.25%
Total	2,726	100.00%	12,386	100.00%

Table 3. The statistics of NLI samples by syllable in VnNewsNLI dataset. (ent. – entailment, neu. – neutral, con. – contradiction).

Length in syllable	Development set			Test set		
	ent	neu	con	ent	neu	con
Premises, ≤ 8	55	54	37	267	266	188
Premises, 9–14	334	332	227	1589	1575	1060
Premises, 15–20	86	85	54	217	214	134
Premises, 20–26	48	35	60	163	155	212
Premises, > 26	424	331	564	1904	1908	2534
All premises	**947**	**837**	**942**	**4140**	**4118**	**4128**
Hypotheses, ≤ 8	62	54	75	297	266	376
Hypotheses, 9–14	346	332	453	1615	1575	2126
Hypotheses, 15–20	70	85	102	167	214	250
Hypotheses, 20–26	45	36	30	155	155	106
Hypotheses, > 26	424	330	282	1906	1908	1270
All hypotheses	**947**	**837**	**942**	**4140**	**4118**	**4128**

4 Experiments

We did some experiments on our VnNewsNLI dataset and on Vietnamese XNLI dataset [11] then compared their results to find if our dataset is useful when building a Vietnamese NLI model. XNLI dataset was manually annotated from English texts then the annotated

results were translated into different languages using machine translators. Therefore, Vietnamese XNLI dataset is a Vietnamese translated NLI dataset. For experiments, we used BERT architecture for training Vietnamese NLI models as shown in Fig. 6.

According to the BERT architecture in Fig. 6, a premise and a hypothesis of a sample will be concatenated into an input. This input has the following order: the "*[CLS]*" token, then all premise's tokens, then the "*[SEP]*" token, then all hypothesis' tokens, and the "*[SEP]*" token at the end. Each input token will be converted to a tuple of word embedding, segment embedding and position embedding. These embeddings will go through BERT architecture to generate a context vector for each input token and a context vector for the whole input. The context vector of the whole input is returned at the "*[CLS]*" position. This vector will be used for identifying the relation between the premise and the hypothesis by a classifier. This classifier is a feed forward neural network fully connected to the context vector of the input. It will be trained in fine-tuning steps. We chose BERT architecture for experiment because it can compute the context vector with syntactic and semantic features of the input [15–17].

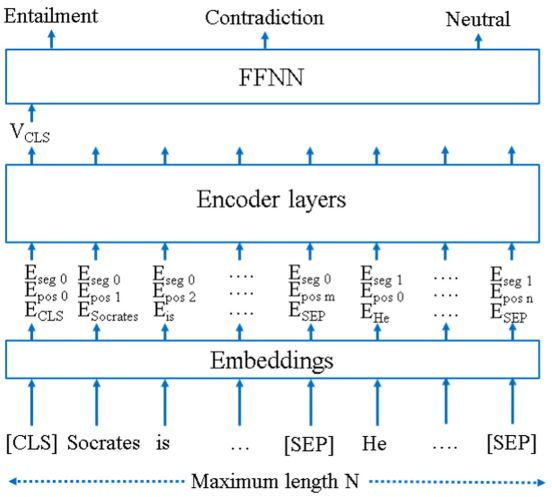


Fig. 6. The illustration of NLI BERT architecture[7]

4.1 Experiment Settings

We built two Vietnamese NLI models using BERT architecture as shown in **Fig. 6**. The first model, viXNLI, was fine-tuned from PhoBERT pretrained-model [13] on Vietnamese version of XNLI development set with word segmentation. The second model, viNLI, was fine-tuned from PhoBERT pretrain-model on our VnNewsNLI development set with word segmentation. We used a small Vietnamese development set of XNLI and an equally small development set of VnNewsNLI for showing the efficiency when using PhoBERT pre-trained model. We used Huggingface python library[18] for implementing

the BERT architecture and fairseq python library[19] for tokenizing Vietnamese words into sub-words. We also used VnCoreNLP [20] for word segmentation.

We fine-tuned these models in 2 to 8 epochs with learning rate of 3.10^{-5}, batch size of 16 and input maximum length of 200 because the $\text{PhoBERT}_{\text{base}}$ pretrained model has the limit input length of 258 and the lengths of the premises and hypotheses are rarely greater than 100 syllables. Other parameters were left with default settings. We chose the best models from checkpoints for testing.

4.2 Experiment Results

The experiment results are shown in Table 4. In Table 4, the accuracy of viNLI model (40.30%) is lower than of viXNLI model (68.64%). In our VnNewsNLI dataset, each premise or hypothesis is a sentence. In XNLI dataset, each premise or hypothesis is translated from English and is a group of sentences. Our viNLI model was fine-tuned on our VnNewsNLI dataset therefore it may not capture the semantic of multi-sentential premise-hypothesis pairs in XNLI test set effectively. In contrast, viXNLI was fine-tuned on XNLI dataset therefore it may capture the semantic of premise-hypothesis pairs effectively in both XNLI's samples and VnNewsNLI's samples. This is the reason why viXNLI's accuracy on XNLI (68.64%) approximates to viXNLI's accuracy on VnNewsNLI (64.04%) while there are big gaps between the viNLI's accuracies on XNLI (40.30%) and on VnNewsNLI (86.05%) and between the viXNLI's accuracy (64.04%) and viNLI's accuracy (86.05%) on the same VnNewsNLI test set.

Table 4. The accuracy of viXNLI and viNLI models on test datasets

Dataset	viXNLI (%)	viNLI (%)
XNLI test set	68.64	40.30
VnNewsNLI test set	64.04	86.05

The accuracy of viNLI model (86.05%) is higher than the accuracy of viXNLI model (64.04%) on VnNewNLI test set. This means our development set is more appropriate for fine-tuning a Vietnamese NLI model than the Vietnamese XNLI's development set. It also means our proposed method is possibly used for building Vietnamese NLI dataset with an attention in adding many multi-sentential.

In our experiment, we fine-tuned viXNLI and viNLI models on two small development sets with about 2,500 samples and test them on two larger test sets with about 5,000 samples and 12,000 samples. The results shows that BERT pre-train models are possibly fine-tuned on small datasets to build effective models as described in [7].

5 Conclusion and Future Works

In this paper, we proposed a method of building a Vietnamese NLI dataset for fine-tuning and testing Vietnamese NLI models. This method is aimed at two issues. The first issue

is the cue marks which are used by the trained model for identifying the relation between a premise and a hypothesis without considering the premise. We addressed this issue by generating samples using eight types of premise-hypothesis pair. The second issue is the Vietnamese writing style of samples. We addressed this issue by generating samples from titles and introductory sentences of Vietnamese news webpages. We used title-introductory pairs of appropriate webpages for reducing annotation cost. These samples were generated by applying a semi-automatic process. For evaluating our method, we built our VnNewsNLI dataset by extracting the title and the introductory sentence of many webpages in a Vietnamese news website VnExpress and applied our building process. When building our VnNewsNLI, we had two people manually annotated each sentence for generating contraction sentences.

We evaluated our proposed method by comparing the results of a NLI model, viXNLI, fine-tuned on Vietnamese XNLI dataset and of a NLI model, viNLI, fine-tuned on our VnNewsNLI dataset. We used the same deep neural network architecture BERT for building these NLI models. The results showed that viNLI model had a higher accuracy (86.05% vs. 64.04%) on our VnNewsNLI test set while it had a lower accuracy (40.30% vs. 68.64%) on Vietnamese XNLI test set when comparing to viXNLI. The VnNewsNLI's accuracy of 86.05% showed a promise of building high-quality Vietnamese NLI dataset from Vietnamese documents for ensuring writing-style.

Currently, our VnNewsNLI dataset contains a quite small number of samples with about 15,000 samples. In future, we will apply our proposed process for building a large and high-quality multi-genre Vietnamese NLI dataset.

References

1. Punyakanok, V., Roth, D., Yih, W.-T.: Natural language inference via dependency tree mapping: an application to question answering. Comput. Linguist. **6**, 10 (2004)
2. Lan, W., Xu, W.: Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In: International Conference on Computational Linguistics, pp. 3890–3902. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018)
3. Minbyul, J., et al.: Transferability of natural language inference to biomedical question answering. In: Conference and Labs of the Evaluation Forum, Thessaloniki, Greece (2020)
4. Falke, T., Ribeiro, L.F.R., Utama, P.A., Dagan, I., Gurevych, I.: Ranking generated summaries by correctness: an interesting but challenging application for natural language inference. In: Annual Meeting of the Association for Computational Linguistics, pp. 2214–2220. Association for Computational Linguistics, Florence, Italy (2019)
5. Pasunuru, R., Guo, H., Bansal, M.: Towards improving abstractive summarization via entailment generation. In: Workshop on New Frontiers in Summarization, pp. 27–32. Association for Computational Linguistics, Copenhagen, Denmark (2017)
6. Dagan, I., Roth, D., Sammons, M., Zanzotto, F.M.: Recognizing Textual Entailment: Models and Applications. Morgan & Claypool Publishers, San Rafael (2013)
7. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics (2019)

8. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A SICK cure for the evaluation of compositional distributional semantic models. In: International Conference on Language Resources and Evaluation, pp. 216–223. European Language Resources Association, Reykjavik, Iceland (2014)
9. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Conference on Empirical Methods in Natural Language Processing, pp. 632–642. Association for Computational Linguistics, Lisbon (2015)
10. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1112–1122. Association for Computational Linguistics, New Orleans (2017)
11. Conneau, A., et al.: XNLI: evaluating cross-lingual sentence representations. In: Conference on Empirical Methods in Natural Language Processing, pp. 2475–2485. Association for Computational Linguistics, Brussels (2018)
12. Nguyen, M.-T., Ha, Q.-T., Nguyen, T.-D., Nguyen, T.-T., Nguyen, L.-M.: Recognizing textual entailment in vietnamese text: an experimental study. In: International Conference on Knowledge and Systems Engineering, pp. 108–113. IEEE, Ho Chi Minh City (2015)
13. Nguyen, D.Q., Nguyen, A.T.: PhoBERT: pre-trained language models for Vietnamese. In: Conference on Empirical Methods in Natural Language, pp. 1037–1042 (2020)
14. Jiang, N., de Marneffe, M.-C.: Evaluating BERT for natural language inference: a case study on the CommitmentBank. In: Conference on Empirical Methods in Natural Language Processing, pp. 6086–6091. Association for Computational Linguistics, Hong Kong (2019)
15. Tenney, I., Das, D., Pavlick, E.: BERT rediscovers the classical NLP pipeline. Annual Meeting of the Association for Computational Linguistics, pp. 4593–4601. Association for Computational Linguistics, Florence (2019)
16. Rogers, A., Kovaleva, O., Rumshisky, A.: A primer in bertology: what we know about how bert works. Trans. Assoc. Comput. Linguist. **8**, 842–866 (2020)
17. Peters, M.E., Neumann, M., Zettlemoyer, L., Yih, W.-T.: Dissecting contextual word embeddings: architecture and representation. In: Conference on Empirical Methods in Natural Language Processing, pp. 1499–1509. Association for Computational Linguistics, Brussels (2018)
18. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics (2020)
19. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. In: Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 48–53. Association for Computational Linguistics, Minneapolis (2019)
20. Vu, T., Nguyen, D.Q., Nguyen, D.Q., Dras, M., Johnson, M.: VnCoreNLP: a Vietnamese natural language processing toolkit. In: Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 56–60. Association for Computational Linguistics, New Orleans (2018)

Proceedings

2021 15th International Conference on Advanced Computing and Applications (ACOMP) | 978-1-6654-0639-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACOMP53746.2021.00001

2021 15th International Conference on Advanced Computing and Applications

ACOMP 2021

A Vietnamese Answer Extraction Model Based on PhoBERT

Chinh Trong Nguyen
Faculty of Computer Science
University of Information Technology, VNU-HCM
Ho Chi Minh City, Vietnam
chinhnt@uit.edu.vn

Dang Tuan Nguyen
Faculty of Information Technology
Saigon University
Ho Chi Minh City, Vietnam
dangnt@sgu.edu.vn

***Abstract*—** PhoBERT pre-trained models have shown its out-performance in many natural language processing tasks. Fine-tuning PhoBERT models is possibly the efficient way to build Vietnamese deep models for answer extraction. For building a Vietnamese answer extraction model using PhoBERT pre-trained model, we need a large SQuAD style annotated dataset. However, there are existing English annotated datasets for answer extraction task and multilingual BERT models which are possibly fine-tuned on English dataset and used in other languages. Therefore, we would find a pre-trained model and a way of fine-tuning this pre-trained model for Vietnamese answer extraction task with a low cost of building Vietnamese annotated dataset. We have conducted the experiments with multilingual BERT pre-trained model and PhoBERT pre-trained model to show the performance of these pre-trained models. In the experiments, we have used Vietnamese translated version of SQuAD dataset and Vietnamese manually annotated dataset to show whether the Vietnamese translated dataset is useful in building an answer extraction model. Our experiment results showed that a PhoBERT pre-trained model is a good choice for building a Vietnamese answer extraction model.

Keywords*—answer extraction, BERT, deep learning, transfer learning.*

I. INTRODUCTION

Answer extraction [2, 3] is an important task in question answering systems. This task aims at extracting a short text ***ans*** in a text content ***T*** with a condition that ***ans*** is the most appropriate answer to a question ***ques***. In the early research in answer extraction, an answer was extracted with two steps. In the first step, the class ***C*** of the question ***ques*** was identified by a question classifier. Then, the named entities of class ***C*** were extracted from the text content ***T*** with an information extractor and the best named entity was chosen to make the answer. Recently, we can use deep learning models for extracting answer candidates from a text context without using the classifier and information extractor. When introducing BERT architecture [1], a fine-tuned model from BERT pre-trained model showed the impressive result on question answering dataset SQuAD v1.1[4]. In the context-aware answer extraction model [3] which has higher result than BERT model in span-F_1 and span-EM scores, a BERT model is also used for generating contextual vector representations to predict an answer-span. This means that deep learning models using BERT architecture is a reasonable approach for answer extraction.

PhoBERT pre-trained models [5] are Vietnamese neural language models using BERT architecture. These pre-trained models, $\text{PhoBERT}_{\text{base}}$ and $\text{PhoBERT}_{\text{large}}$, showed the SOTA results on many Vietnamese NLP tasks [5] however the results of Vietnamese question answering task using PhoBERT pre-trained models were not shown. Therefore, PhoBERT models may be appropriate to build a Vietnamese answer extraction model by fine-tuning them on Vietnamese manually annotated dataset. However, there are existing English datasets for the question answering task that we can translate them into Vietnamese, and there are multilingual BERT pre-trained models which can be used as cross-lingual models for answer extraction in Vietnamese. The cross-lingual models are possibly trained or fine-tuned in a certain language and used for prediction in the other language.

In this paper, we would like to conduct some experiments to show if PhoBERT pre-trained models are good choices for building answer extraction models for Vietnamese question answering systems and whether the Vietnamese translated dataset is useful for building those models. In our experiments, we would like to show the influence of the segment embeddings of BERT models in answer extraction, and the efficiency of $\text{PhoBERT}_{\text{base}}$ pre-trained model in comparison to multilingual $\text{BERT}_{\text{base}}$ pre-trained model. This paper presents our work in five sections. Section 1 introduces our questions about using PhoBERT in building an answer extractor. Section 2 presents some background information about using a BERT pre-trained model in answer extraction. Section 3 presents our approaches in building Vietnamese answer extraction models. Section 4 presents the experiments and the datasets used in the experiments for showing the efficiency of $\text{PhoBERT}_{\text{base}}$ pre-trained model and the benefit of Vietnamese translated dataset. Finally, some conclusions and future works are presented in section 5.

II. BACKGROUNDS

A. BERT architecture

Bidirectional Encoder Representations from Transformers (BERT) [1] is deep neural network containing M encoder layers ($\boldsymbol{M}$ = 12 in $\text{BERT}_{\text{base}}$ and $\boldsymbol{M}$=24 in $\text{BERT}_{\text{large}}$ settings). Each encoder is the encoder part of a transformer [6]. The input of BERT is a tuple $<\boldsymbol{E_w}, \boldsymbol{E_p}, \boldsymbol{E_s}>$ in which $\boldsymbol{E_w}$ is a list of word embeddings, $\boldsymbol{E_p}$ is a list of position embeddings, and $\boldsymbol{E_s}$ is a list of segment embeddings. Position embeddings and segment embeddings are used to encode the position of each token of an input text. These embeddings are pass through encoder layers to generate the context vector $\boldsymbol{V_{word}}$ of each input token. A BERT model usually uses two segment embeddings for encoding the first and the second text spans in the input. If the segment separation of the input is needed, all tokens of the first text span will be assigned the segment number 0 and all tokens of the second text span will be assigned the segment number 1. The BERT model also uses two special token to mark the boundary of the text spans. Token *[CLS]* indicates the beginning of the input text and token *[SEP]* indicates the end of a text span. The

2688-0202/21/$31.00 ©2021 IEEE
DOI 10.1109/ACOMP53746.2021.00022

position and segment embeddings will be trained jointly with other parameters of BERT model. Each BERT model has a maximum length of input tokens ***N***. These input tokens are words, numbers, punctuations, and sub-words. Sub-words are parts of word. They are not always morphemes. They are used for reducing the vocabulary size. These tokens will be converted into embedding tuples when they go through embeddings layers. **Figure 1** illustrates the BERT architecture.

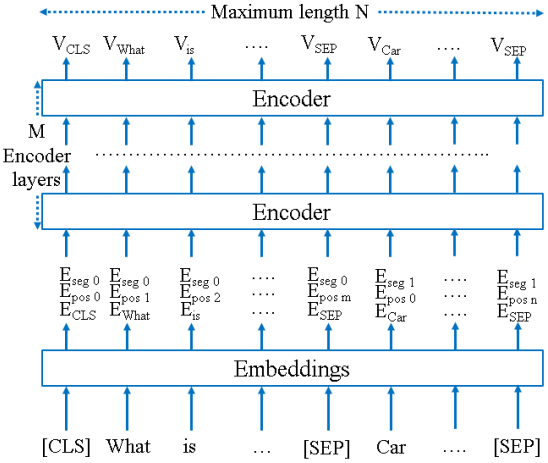


Figure 1 *The illustration of BERT architecture[1]*

According to the BERT architecture in **Figure 1**, the context vectors of words in a text span are calculated in three steps. In the first step, the text span must be tokenized into tokens. Each token is assigned a segment number and an attention value. The attention value of a token indicates if this token contributes to the context (value 1) or does not (value 0). In the second step, the text span is converted to three lists of embeddings by looking up the tokens' word embeddings, position embeddings and segment embeddings. In the third step, these embeddings are passed through encoder layers to generate the token's context embeddings.

BERT architecture showed that it can compute the context vector of each input word with syntactic and semantic information [7][8][9]. In natural language, the word usage and the word position in a sentence show the word meaning and syntactic function, respectively. In a BERT model, the word embeddings represent the distributional semantic of input tokens. They are estimated in a context prediction task [10]. As the result, if two tokens can be used in the similar contexts, their word embeddings are similar. Before passing through encoder layers, the position embeddings and the segment embeddings will be added to word embeddings. The position and segment embeddings will be trained jointly with other parameters of a BERT model in masked prediction and next sentence prediction tasks [1] therefore they can encode the effect of word position to the semantic of the input text. In other words, position and segment embeddings represent the syntactic information of words. Each encoder layer uses entire its input embeddings to generate the context vector of each token. The higher encoder layer will generate the deeper semantic and syntactic features therefore a context vector generated from a BERT model can encode the semantic and syntactic of a token in a text.

In addition, The researches in training methods [11, 12] have improved BERT models in model size, training time and SOTA results in NLP tasks. These mean deep learning with BERT architecture is a reasonable approach for answer extraction which can be solved with text classification techniques.

PhoBERT [5] pre-trained models, $\text{PhoBERT}_{\text{base}}$ and $\text{PhoBERT}_{\text{large}}$, are Vietnamese language models using BERT architecture. They were trained on a very large Vietnamese text corpus. The text corpus was applied Vietnamese word segmentation in preprocess step thus these models are better for Vietnamese NLP tasks than multilingual BERT[1]. PhoBERT models have two settings which may affect the performance. The first one is that the input length is 258 tokens. This input length is about a half of the input length of BERT. This means the context for finding answer span will be narrowed and the answer identification will possibly reduce in the answer extraction task. The second one is that the segment embedding size is 1. This means we cannot separate the question segment from the context segment using segment embeddings. The way to separate the question segment from the context segment is to place them in a pair of tokens **<s>** and **</s>** when using PhoBERT models.

B. Answer extraction using a BERT pre-trained model

Answer extraction is an application of name entity recognition methods. Name entity recognition is aimed at identifying the text spans of some types from a text content. These types may be person name, organization name, number, etc. Answer extraction method using a BERT model is also to identify the text spans which are possibly the answers from a text content for a given question. In this method, a BERT pre-trained model is used for generating context vectors of all words from the question and the text content. Then, two classifiers will be used for identifying the start positions and the end positions of the answer spans. These two classifiers are feed-forward neural networks (FFNN) fully connected to each context vector. The neural network architecture for answer extraction is illustrated in **Figure 2** in which Start FFNN and End FFNN are classifiers which identify start positions and end positions of the answer spans.

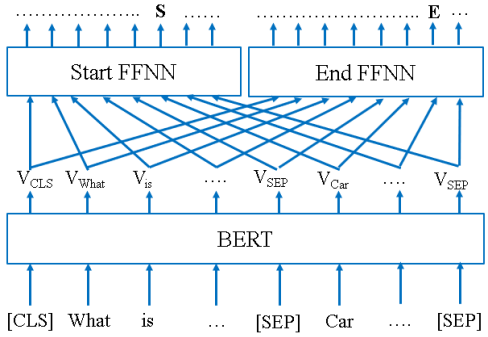


Figure 2 *The deep neural network architecture for answer extraction[1]*

The input of this architecture contains a list of tokens composed of the *[CLS]* token at the beginning, all question tokens with preserved order in the next, then the *[SEP]* token at the end of question, all text tokens with preserved order in the next, and the *[SEP]* token at the end of the text content. The question segment and the text content segment are also separated by assigning the segment value 0 to all elements

from [CLS] position to the first [SEP] position and the segment value 1 to the rest of the input. The **Figure 1** shows the details of an input with two text spans.

Start FFNN and End FFNN classifiers calculate the scores of start positions S_{word} and end positions E_{words} for each context vector V_{word}, respectively. The expected start position S is the position with the maximum value of all S_{word} values. Similarly, the expected end position E is the position with the maximum value of all E_{word} values. These conditions are used for calculating the errors when fine-tuning the answer extraction model.

Although the results of answer extraction model are the start position S and the end position E of the answer span, these positions are not always valid. They are possibly in question segment, they are not acceptable because the end position E is lower than the start position S, or they are not reasonable because the length of the answer span is too long. Therefore, the text span with start position S_p and end position E_p is selected with the following conditions in many possible text spans.

- S_p and E_p are not in question segment and are not the positions of *[CLS]* or *[SEP]* tokens.
- The length of the span is not greater than a predefined number LEN.
- Given R_S is the score of the position S_p from Start FFNN classifier, R_E is the score of the position E_p from End FFNN classifier. The overall score R of the span [S_p, E_p] is the sum of R_S and R_E. The span with the maximum overall score R will be selected. In practice, the overall score R may differ from this formula.

C. Answer extraction evaluation

The SQuAD question answering tasks [4, 13] use two measures F_1 and EM (exact match) for answer extraction evaluation. These measures calculation needs a test set and an answer prediction set. The sample size of these two sets is N. The ith sample of the test set contains a question q_i, a context t_i and an answer a_i. For each ith sample, a testing model returns its predicting answer w_i in the answer prediction set.

Assuming a_i=$a_{i1}a_{i2}…a_{im}$ and w_i= $w_{i1}w_{i2}…w_{in}$, i∈[1,N] where a_{ij} and w_{ik} are words in the answer a_i and the predicting answer w_i respectively, then the exact match EM_i, precision P_i, recall R_i and f$_1$-measure F_{1i} of the prediction answer w_i are calculated with formulae (1), (2), (3), and (4), respectively.

$$EM_i = \begin{cases} 0 \; if \; a_i \neq w_i \\ 1 \; if \; a_i = w_i \end{cases} \quad (1)$$

$$P_i = \frac{|\{a_{ij}\} \cap \{w_{ik}\}|}{|\{w_{ik}\}|} \quad (2)$$

$$R_i = \frac{|\{a_{ij}\} \cap \{w_{ik}\}|}{|\{a_{ij}\}|} \quad (3)$$

$$F_{1i} = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (4)$$

Then, the F$_1$-measure F_1 and exact match EM scores of the model is calculated with formulae (5) and (6), respectively.

$$F_1 = \frac{1}{N} \sum_N F_{1i} \quad (5)$$

$$EM = \frac{1}{N} \sum_N EM_i \quad (6)$$

III. Our approaches

We have three approaches in building Vietnamese answer extraction models. The first approach is to fine-tune PhoBERT pre-trained models, which are mono-lingual models, on Vietnamese annotated dataset. This approach has an advantage of using language-specific pre-trained model if we have the manually annotated dataset for that language. The second approach is to fine-tune a multilingual BERT pre-trained models on Vietnamese annotated dataset. This approach is obviously not good as the first approach however it might be reasonable if there is only translated dataset. The third approach is to fine-tune a multilingual BERT pre-trained models on English annotated dataset and use them for extracting answers in Vietnamese. This approach is motivated from the cross-lingual language understanding.

At present, we have only a small Vietnamese annotated dataset for question answering task therefore we have to choose one of the above approaches. For selecting a reasonable approach, we will conduct three experiments. The first experiment is to show the influence of segment embeddings of BERT models in answer extraction. Although PhoBERT pre-trained models are possibly fine-tuned for question answering task, they are built on BERT architecture, and they have only one segment embedding meaning the segment embeddings does not affect the answer extraction results. Therefore, we would like to show the influence of segment embeddings in BERT models. If the segment embeddings do not affect the answer extraction results in BERT models, we have more proofs in using PhoBERT models for answer extraction.

The second experiment is to compare the performance of PhoBERT model and multilingual BERT models in Vietnamese answer extraction when using Vietnamese translated dataset for fine-tuning. The best model in this experiment might show the reasonable way to build a Vietnamese answer extraction when we have only small dataset.

The third experiment is to compare the performance of a PhoBERT model fine-tuned on large Vietnamese translated dataset and a PhoBERT model fine-tuned on a small Vietnamese manually annotated dataset. This result will show whether the translated dataset is useful in building a Vietnamese answer extraction model.

IV. Experiments

Our experiments have been conducted to answer three questions. The first question is "*does the number of segment embeddings affect the answer extraction results of BERT fine-tuned models?*". For answering this question, we fine-tuned a BERT model with segment value 0 assigned to all input tokens as shown in **Figure 3**, and compared its results with the results of a BERT model fine-tuned with segment value 0 assigned to question tokens and segment value 1 assigned to text content tokens as shown in **Figure 1**. If the F_1 and EM scores of these two models are approximate, the influence of segment embeddings is not important in BERT models therefore that the PhoBERT models use only one

segment embedding does not affect to answer extraction results much.

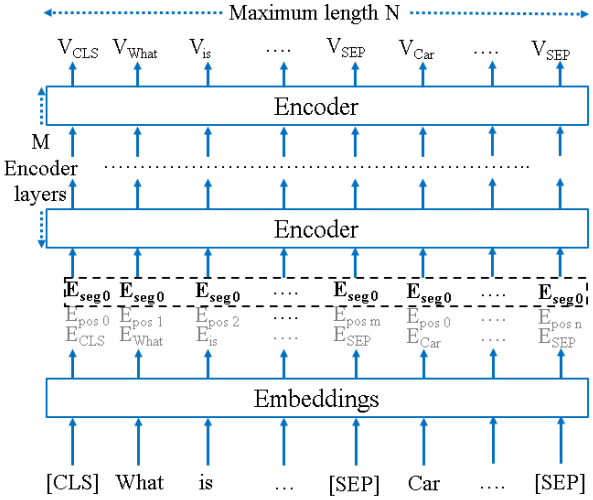


Figure 3 *The input without segment separation for the BERT model*

The second question is "*is PhoBERT pre-trained model better than multilingual BERT model when using Vietnamese translated dataset in fine-tuning answer extraction models for Vietnamese?*". For answering this question, we have fine-tuned a $\text{PhoBERT}_{\text{base}}$ pre-trained model [5] and a multilingual $\text{BERT}_{\text{base}}$ pre-trained model [1] on a same Vietnamese translated training set. We have also fine-tuned a multilingual $\text{BERT}_{\text{base}}$ model on English dataset SQuAD. Then we have compared their three results to choose the best approach for building Vietnamese answer extraction model when we did not have a large Vietnamese annotated dataset.

The third question is "*are Vietnamese translated versions of existing question answering datasets useful for fine-tuning an answer extraction model using BERT pre-trained model for Vietnamese?*". For answering this question, we fine-tuned a $\text{PhoBERT}_{\text{base}}$ pre-trained model on native Vietnamese training set and compared it with the $\text{PhoBERT}_{\text{based}}$ model fine-tuned on Vietnamese translated dataset.

A. Datasets

We used SQuAD v1.1 [4], MLQA [14], and XQuAD [15] datasets in our experiments. SQuAD is the Stanford question answering dataset which contains about 100k questions with the text contents and their answers in English. MLQA is a multilingual question answering dataset which is used for evaluating the cross-lingual question answering task. MLQA has a test set containing about 11k questions with answers in English and a test set containing about 5k questions with answers in Vietnamese. XQuAD is also a multilingual question answering dataset which has test sets of 1190 questions with answers in English and in Vietnamese.

In the influence of segment embeddings experiments, we used the training set and the development set of SQuAD v1.1 dataset [4]. We have fine-tuned $\text{BERT}_{\text{base}}$ models on SQuAD v1.1 training set and evaluated these models on SQuAD v1.1 development set (called eSQA), MLQA English test set [14] (called eMLQA), and XQuAD English test set [15] (called eXQA).

In the experiments of choosing the best model from $\text{PhoBERT}_{\text{base}}$ and multilingual $\text{BERT}_{\text{base}}$ pre-trained models when fine-tuning on Vietnamese translated dataset, we used the Vietnamese translation of SQuAD v1.1 training set (called vSQA), and a Vietnamese SQuAD-style training set (called UITt) for fine-tuning. For evaluation, we test our models on MLQA Vietnamese test set [14] (called vMLQA), XQuAD Vietnamese test set [15] (called vXQA), and a Vietnamese SQuAD-style test set (called UITs). The UITt and UITs was made by students at University of Information Technology. The statistics of training sets and test sets in Vietnamese are shown in TABLE 1.

TABLE 1 THE STATISTICS OF VIETNAMESE TRAINING SETS AND TEST SETS (THE UNIT OF LENGTH IS VIETNAMESE WORD)

Criteria	vSQA	vMLQA	vXQA	UITt	UITs
#Question	74,532	5,495	1,190	1,225	243
Max. question length	54	32	34	95	27
Min. question length	2	3	3	2	2
Avg. question length	11	9	12	10	8
Max. answer length	141	63	26	120	37
Min. answer length	1	1	1	1	1
Avg. answer length	3	4	3	9	5
Max. context length	785	1,978	597	327	325
Min. context length	18	8	34	25	66
Avg. context length	143	171	147	113	141

Our vSQA was translated from SQuAD v1.1 by using a machine translator. Many samples have been removed because their translated answers did not appear exactly in the translated text contents when translating SQuAD v1.1. In vSQA, each question has only one answer which is translated from the first answer in the answer set of the corresponding question in SQuAD v1.1. We have also picked 100 translated contexts randomly and checked whether the quality of machine translation is acceptable. These translated contexts have been checked by an English-Vietnamese translator. We have used a five-grade scale to indicate the quality of each translated context. A translated text content has grade 1 in quality if its meaning is very different from the original text content's meaning. We cannot use it for fine-tuning. For example, the English original text content and its Vietnamese translation are followings.

The original text content is "*Adult contemporary tends to have lush , soothing and highly polished qualities where*

emphasis on melody and harmonies is accentuated . It is usually melodic enough to get a listener 's attention , and is inoffensive and pleasurable enough to work well as background music . Like most of pop music , its songs tend to be written in a basic format employing a verse–chorus structure ."

The Vietnamese machine translation: "*Bản_nhạc đương_đại dành cho người_lớn có xu_hướng có chất_lượng tươi_tốt , nhẹ_nhàng và được đánh bóng cao khi nhấn_mạnh vào giai_điệu và hoà_âm . Nó thường đủ du_dương để thu_hút sự chú_ý của người nghe , và đủ inoffensive và thú_vị để hoạt_động tốt như nhạc nền . Giống như hầu_hết nhạc pop , các bài hát của nó có xu_hướng được viết ở định_dạng cơ_bản sử_dụng cấu_trúc câu-chorus*". This translation contains many untranslated words and many wrongly translated words, for example the translated result of "*highly polished*" was "*được đánh bóng cao*".

A translated text content has grade 2 in quality if it may express the related meaning of the original text content but there are wrong translated words. This translated content will be a noisy sample therefore we should not use it. For example, the English original text content and its Vietnamese translation are followings.

The original text content is "*The same divine agencies who caused disease or harm also had the power to avert it , and so might be placated in advance . Divine consideration might be sought to avoid the inconvenient delays of a journey , or encounters with banditry , piracy and shipwreck , with due gratitude to be rendered on safe arrival or return . In times of great crisis , the Senate could decree collective public rites , in which Rome 's citizens , including women and children , moved in procession from one temple to the next , supplicating the gods .*"

The Vietnamese machine translation: "*Cũng chính những cơ_quan thần_thánh , những người đã gây ra bệnh_tật hoặc tổn_hại cũng có sức_mạnh để ngăn_chặn nó , và vì_vậy có_thể được xoa_dịu trước . Divine có_thể được xem_xét để tránh sự chậm_trễ bất_tiện của cuộc hành_trình hoặc khi gặp phải băng cướp , cướp_biển và đắm tàu , với lòng biết_ơn đến hạn phải được thực_hiện khi đến hoặc trở về an_toàn . Trong những thời_kỳ khủng_hoảng lớn , Thượng_viện có_thể ban_hành nghi_thức công_cộng tập_thể , trong đó công_dân của Rome , bao_gồm cả phụ_nữ và trẻ_em , di_chuyển trong đám rước từ ngôi đền này sang ngôi đền khác , cầu_khẩn các vị thần .*" This translation mentions the God's power in averting harms as presented in original text content however there are many wrong translated words, for example "*những cơ_quan thần_thánh*" which should be "*những quyền_lực thần_thánh*", so that we cannot use this translation.

A translated text content has grade 3 in quality if it can show the meaning of the original text content but there are some errors in word translation. For example, the English original text content and its Vietnamese translation are followings.

The original text content is "*The foundation explains on its website that its trustees divided the organization into two entities : the Bill & Melinda Gates Foundation (foundation) and the Bill & Melinda Gates Foundation Trust (trust) . The foundation section, based in Seattle , US , " focuses on improving health and alleviating extreme poverty , " and its trustees are Bill and Melinda Gates and Warren Buffett . The trust section manages " the investment assets and transfer proceeds to the foundation as necessary to achieve the foundation 's charitable goals " —it holds the assets of Bill and Melinda Gates , who are the sole trustees , and receives contributions from Buffett .*"

The Vietnamese machine translation is "*Quỹ giải_thích trên trang_web của mình rằng những người được uỷ_thác đã chia tổ_chức thành hai tổ_chức : Quỹ_Bill & Melinda_Gates (quỹ) và Quỹ_Bill Melinda_Gates_Foundation_Trust (uỷ_thác) . Tổ_chức có trụ_sở tại Seattle , Hoa_Kỳ , " tập_trung vào việc cải_thiện sức_khoẻ và giảm nghèo cùng_cực , " và những người được uỷ_thác là Bill và Melinda_Gates và Warren_Buffett . Phần uỷ_thác quản_lý " tài_sản đầu_tư và số tiền chuyển đến quỹ khi cần_thiết để đạt được các mục_tiêu từ_thiện của quỹ " — nó nắm giữ tài_sản của Bill và Melinda_Gates , những người được uỷ_thác duy_nhất và nhận đóng_góp từ Buffett .*" There are two errors in word translation. The first error is that the phrase "*two entities*" was translated as "*hai tổ_chức*" while the correct translation should be "*hai đối_tượng nhận*". The second error is that the phrase "*The foundation section*" was translated as "*tổ_chức*" while the correct translation should be "*phần quỹ*". Although the translation contains some errors, its meaning is similar to the original text content's meaning. We might use this translation with cautions because it may be a noisy sample.

A translated text content has grade 4 in quality if it and the original text content have the same meaning. The translation has no errors, but the writing-style is not Vietnamese native. For example, the English original text content and its Vietnamese translation are followings.

The original text content is " *A desire to be closer to the urban scene has also attracted some young professionals to reside in inner ring suburbs such as Grosse Pointe and Royal Oak , Detroit . Detroit 's proximity to Windsor , Ontario , provides for views and nightlife , along with Ontario 's minimum drinking age of 19 . A 2011 study by Walk Score recognized Detroit for its above average walkability among large U.S. cities . About two-thirds of suburban residents occasionally dine and attend cultural events or take in professional games in the city of Detroit .*"

The Vietnamese machine translation is " *Mong_muốn gần_gũi hơn với khung_cảnh đô_thị cũng đã thu_hút một_số chuyên_gia trẻ đến cư_trú ở các vùng ngoại_ô vòng trong như Grosse_Pointe và Royal_Oak , Detroit . Sự gần_gũi của Detroit với Windsor , Ontario , mang đến cho bạn tầm nhìn và cuộc_sống về đêm , cùng với độ tuổi uống rượu tối_thiểu của Ontario là 19 . Một nghiên_cứu năm 2011 của Walk_Score đã công_nhận Detroit về khả_năng đi bộ trên mức trung_bình giữa các thành_phố lớn của Hoa_Kỳ . Khoảng hai_phần_ba cư_dân ngoại_ô thỉnh_thoảng dùng_bữa và tham_dự các sự_kiện văn_hoá hoặc tham_gia các trò_chơi chuyên_nghiệp ở thành_phố Detroit .*" This translation contains two phrases which a native writer did not use. The first phrase is "*mang đến cho bạn*" which should be "*mang đến*" or "*cho bạn*". The second phrase is "*giữa các thành_phố lớn*" which should be "*khi so với các thành phố lớn*" or "*trong số các thành phố lớn*". However, this translation is a good and it should be a sample in training set.

A translated text content has grade 5 if it ensures the meaning of the original text content and the Vietnamese

writing-style. This translation will be a good sample in training set.

From the checking results, 2% translation are in grade 1, 3% translation are in grade 2, 22% translation are in grade 3, 14% translation are in grade 4, and 59% translation are in grade 5. With these results, we might fine-tune a Vietnamese answer extraction model on the Vietnamese translated training set of SQuAD v1.1 dataset however the fine-tuned model may not very good because the rate noisy samples in the dataset is about 27%.

B. Experiment settings

We used Huggingface python library [16] for implementing the architecture shown in **Figure 2**. For answering the first question, we have fine-tuned two $BERT_{base}$ models on SQuAD v1.1 training set. The first model, named QI-seg, has been fine-tuned with the input separated using two segment embeddings. The second model, named QI-noseg, has been fine-tuned with input using only one segment embeddings.

For answering the second question, we have fine-tuned three models. The first model, named QII-PhoBERT, is a $PhoBERT_{base}$ pre-trained model fine-tuned on vSQuAD dataset. The second model, named QII-mBERT, is a multilingual $BERT_{base}$ pre-trained model fine-tuned on vSQA dataset. The third model, named QII-mXBERT, is a multilingual $BERT_{base}$ pre-trained model fine-tuned on SQuAD v1.1 training set and then fine-tuned on Vietnamese annotated training set UITt. The QII-mXBERT was a cross-lingual model thus we have fine-tuned it with a small Vietnamese training set to improve its performance.

For answering the third question, we have fine-tuned a $PhoBERT_{base}$ pre-trained model, named QIII-PhoBERT, on Vietnamese UITt training set. The QIII-PhoBERT model will be compare to QII-PhoBERT to show whether the translated dataset is useful to build Vietnamese answer extraction model.

In our experiments, $BERT_{base}$ and multilingual $BERT_{base}$ pre-trained models have been fine-tuned with maximum input length of 384 tokens, and $PhoBERT_{base}$ models have been fine-tuned with maximum input length of 240 tokens. We used learning rate at 3.10^{-5}, the number of fine-tuning epochs from 2 to 14, and batch size at 16 when fine-tuning $BERT_{base}$ models and at 12 when fine-tuning $PhoBERT_{base}$ models. We have used word segmentation tool from VnCoreNLP [17] in the preprocessing step when fine-tuning $PhoBERT_{base}$ models. We have chosen the maximum length of answer span at 30 tokens for all models.

C. The results

We have chosen models from the best checkpoints when fine-tuning all models for testing. The test results of QI-seg and QI-noseg models on SQuAD v1.1 development set are shown in TABLE *2*.

TABLE 2 THE TEST RESULTS OF MODELS WITH AND WITHOUT SEGMENT SEPARATION IN INPUT TOKENS

Dataset	**QI-seg**		**QI-noseg**	
	F_1(%)	EM (%)	F_1(%)	EM (%)
eSQA	86.22	77.78	85.84	78.03
eXQA	79.66	67.23	80.54	69.33
eMLQA	73.60	60.28	73.55	60.18

According to TABLE *2*, our models QI-seg and QI-noseg have reached the regular performance for question answering task on SQuAD v1.1 dataset using $BERT_{base}$ pre-trained model. These two models have been fine-tuned in 2 epochs. In this test, we did not focus on archiving SOTA results, but we focus on the differences in performance of the two models.

In TABLE *2*, the F_1 and EM scores of QI-seg model are slightly different from F_1 and EM scores of QI-noseg model on all three test sets. On eSQA test set, F_1 score of QI-seg (86.22%) is higher than of QI-noseg (85.84%) while EM score of QI-seg (77.78%) is lower than of QI-noseg (78.03%). On eXQA test set, F_1 and EM scores of QI-seg (79.66% and 67.23%) are lower than of QI-noseg (80.54% and 69.33%). However, on eMLQA test set, F_1 and EM scores of QI-seg (73.60% and 60.28%) are higher than of QI-noseg (73.55% and 60.18%). These results have showed that there are very little differences between QI-seg and QI-noseg models. Therefore, we can confirm that the segment embeddings in BERT models have very small influences on the answer extraction results. Thus, the answer of our first question is "*the number of segment embeddings does not affect the answer extraction results of BERT fine-tuned models much*." This means that PhoBERT models have only one segment embedding will have a small affect on the answer extraction results.

The test results of QII-PhoBERT, QII-mBERT and QII-mXBERT models on vMLQA and vXQA are shown in TABLE *3*. The F_1 and EM scores of QII-PhoBERT model on vMLQA test set (57.89% and 40.11%) and on vXQA test set (71.26% and 50.67%) are higher than those of QII-mBERT and QII-mXBERT models on the same test sets. These results show that the model fine-tuned from PhoBERT pre-trained model is better than both multilingual model and cross-lingual model when using Vietnamese translated dataset in fine-tuning Vietnamese answer extraction models.

TABLE 3 THE TEST RESULTS OF THE MODELS FINE-TUNED FROM PHOBERT AND FROM MULTILINGUAL BERT PRE-TRAINED MODELS

Dataset	QII-PhoBERT		QII-mBERT		QII-mXBERT	
	F_1 (%)	EM (%)	F_1 (%)	EM (%)	F_1 (%)	EM (%)
vMLQA	**57.89**	**40.11**	53.58	35.56	57.16	38.23
vXQA	**71.26**	**50.67**	65.20	45.71	68.37	48.57

In TABLE *3*, the QII-PhoBERT model outperformed QII-mBERT and QII-mXBERT models in all Vietnamese test sets. Because QII-PhoBERT and QII-mBERT models are fine-tuned on the same training set, the differences must come from pre-trained models. Although the cross-lingual model QII-mXBERT has been fine-tuned on small Vietnamese annotated dataset UITt and had a better performance than QII-mBERT, it was not good as QII-PhoBERT model. Therefore, we can confirm that PhoBERT pre-trained models are better multilingual BERT pre-trained models in building Vietnamese answer extraction models and this is also the answer of the second question.

TABLE *4* shows the test results of QII-PhoBERT model and QIII-PhoBERT model. The results show that the QII-PhoBERT model has the higher F_1 and EM scores than QIII-PhoBERT's F_1 and EM scores. However, the training set used for fine-tuning QII-PhoBERT model is much larger than the training set used for fine-tuning QIII-PhoBERT

model. In addition, the test results of QI-noseg model on eXQA and eMLQA test sets shown in TABLE *2* are much higher than the results of QII-PhoBERT model on vXQA and vMLQA test sets shown in TABLE *3*. Although these results are not comparable, they indicate that our machine translation training set cannot replace the native annotated one because we had about 73% good translation and the rest of translation may be the noisy samples. Therefore, we cannot confirm that Vietnamese translated datasets can be used as a replacement of native Vietnamese annotated datasets for fine-tuning Vietnamese answer extraction models. Because F_1 score (61.39%) and EM score (41.48%) of QII-PhoBERT model nearly doubled the scores of QIII-PhoBERT model, we can take a note that Vietnamese machine translation datasets might be useful in building Vietnamese answer extraction models when we have only a small Vietnamese annotated dataset.

TABLE 4 THE TEST RESULTS OF MODELS FINE-TUNED ON VIETNAMESE AND VIETNAMESE TRANSLATED TRAINING SET

Model	**F_1**(%)	**EM** (%)
QIII-PhoBERT	38.00	19.21
QII-PhoBERT	61.39	41.48

V. CONCLUSSIONS AND FUTURE WORKS

In this paper, we presented our work in building an answer extraction model using $PhoBERT_{base}$ pre-trained model when we have only small Vietnamese annotated dataset. There are three approaches in building Vietnamese answer extraction models in this case. These approaches are fine-tuning $PhoBERT_{base}$ pre-trained models on Vietnamese translated dataset, fine-tuning multilingual $BERT_{base}$ pre-trained models on Vietnamese translated dataset and fine-tuning multilingual $BERT_{base}$ pre-trained models on English annotated dataset then fine-tuning it on a small Vietnamese annotated dataset. Before choosing a reasonable approach, we have conducted an experiment to show the influence of segment embeddings to $BERT_{base}$ answer extraction models because $PhoBERT_{base}$ pre-trained model has only one segment embedding. Then, we have fine-tuned $PhoBERT_{base}$ and multilingual $BERT_{base}$ models on a Vietnamese translated dataset and evaluated them on existing cross-lingual test sets and on a small Vietnamese test set. The test results of these models show that the segment embeddings have very small influences on answer extraction results and fine-tuning $PhoBERT_{base}$ pre-trained model is a good choice for building Vietnamese answer extraction models even when fine-tuning on Vietnamese translated dataset.

We also conducted an experiment to check if the Vietnamese translated version of existing question answering datasets is useful for fine-tuning PhoBERT pre-trained model. Although our datasets were not large enough to confirm the effect of the Vietnamese translated dataset, the results on MLQA (F_1=57.89%, EM=40.11%) and on XQuAD (F_1=71.26%, EM=50.67%) show that the Vietnamese translated training sets might be used for fine-tuning Vietnamese answer extraction models when we have only a small Vietnamese annotated dataset.

In future, we will build a large Vietnamese SQuAD-style annotated dataset for building a Vietnamese question-answering system using PhoBERT pre-trained model because this pre-trained model has shown its efficiency in our experiments.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171--4186.

[2] S. Abney, M. Collins, and A. Singhal, "Answer extraction," in Applied Natural Language Processing Conference, Seattle, Washington, USA, 2000, pp. 296--301.

[3] Y. Seonwoo, J.-H. Kim, J.-W. Ha, and A. Oh, "Context-Aware Answer Extraction in Question Answering," in Conference on Empirical Methods in Natural Language Processing, Online, 2020, pp. 2418--2428.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in Empirical Methods in Natural Language Processing, Austin, Texas, 2016, pp. 2383--2392.

[5] D. Q. Nguyen, and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in Conference on Empirical Methods in Natural Language, 2020, pp. 1037--1042.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in Neural Information Processing Systems, Long Beach, CA, USA, 2017, pp. 5998--6008.

[7] I. Tenney, D. Das, and E. Pavlick, "BERT Rediscovers the Classical NLP Pipeline," in Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 4593--4601.

[8] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," *Transactions of the Association for Computational Linguistics,* vol. 8, pp. 842--866, 2020.

[9] M. E. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih, "Dissecting Contextual Word Embeddings: Architecture and Representation," in Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018, pp. 1499--1509.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in International Conference on Learning Representations, Arizona, USA, 2013.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," in Workshop on Energy Efficient Machine Learning and Cognitive Computing, Vancouver BC, Canada, 2019.

[12] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in International Conference on Learning Representations, 2020.

[13] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," in Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 784--789.

[14] P. Lewis, B. Oğuz, R. Rinott, S. Riedel, and H. Schwenk, "MLQA: Evaluating cross-lingual extractive question answering," in Annual Meeting of the Association for Computational Linguistics, Online, 2020, pp. 7315--7330.

[15] M. Artetxe, S. Ruder, and D. Yogatama, "On the cross-lingual transferability of monolingual representations," in Annual Meeting of the Association for Computational Linguistics, Online, 2020, pp. 4623--4637.

[16] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, and

others, “Transformers: State-of-the-Art Natural Language Processing,” in Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38--45.

[17] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, “VnCoreNLP: A Vietnamese Natural Language Processing Toolkit,” in Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, New Orleans, Louisiana, 2018, pp. 56--60.

Elementary discourse unit segmentation for Vietnamese texts

Chinh Trong Nguyen

University of Information Technology,
VNU-HCM,
Ho Chi Minh City, Vietnam
Email: chinhnt@uit.edu.vn

Dang Tuan Nguyen*

Saigon University,
Ho Chi Minh City, Vietnam
Email: dangnt@sgu.edu.vn
*Corresponding author

Abstract: Elementary discourse unit (EDU) segmentation is an important problem in discourse analysis of text. In Vietnam, we do not have any tool or model official published to solve this problem yet. Therefore, we would like to propose a solution for this problem. Our approach is to apply a sequential labelling method for identifying the beginning of each EDU in a sentence. For sequential labelling method, we use a deep neural network architecture containing a BERT for generating word feature vectors as transfer learning approach and a feed forward neural network for identifying the tag of every word. For building the model, we have automatically built an EDU segmentation dataset from a Vietnamese constituent treebank NIIVTB and used this dataset to fine-tune PhoBERT pretrained model. The results show that our EDU segmentation model has span-based F1 score of 0.8, which is sufficient to be used in practical tasks.

Keywords: EDU segmentation; sequential labelling; BERT; transfer learning.

Reference to this paper should be made as follows: Nguyen, C.T. and Nguyen, D.T. (2022) 'Elementary discourse unit segmentation for Vietnamese texts', *Int. J. Intelligent Information and Database Systems*, Vol. 15, No. 3, pp.249–266.

Biographical notes: Chinh Trong Nguyen is currently a PhD student at University of Information Technology, Vietnam National University – Ho Chi Minh City, Vietnam. His research interests are text retrieval, question-answering system and computational linguistics.

Dang Tuan Nguyen is an Associate Professor in Information Technology at Saigon University, Ho Chi Minh City, Vietnam. His research interests focus on natural language processing, artificial intelligence and computational linguistics.

1 Introduction

Rhetorical structure theory (RST) has been applied in many natural language processing tasks; such as, text classification using discourse tree structure (Chernyavskiy and Ilvovsky, 2020), conversational agents (Cervone, 2020), text summarisation (Marcu, 1998) and argument evaluation (Taboada and Mann, 2006). RST is a framework for representing the text in the hierarchical structure in which each discourse unit may have many discourse relations to the others (Mann and Thompson, 1988). Therefore, the most important task in RST is parsing a text into RST tree.

RST parsers have been studied for years. These RST parsers are possibly divided into two types: rule-based (Marcu, 1997, 1998; Polanyi et al., 2004; Subba and Di Eugenio, 2009) and machine learning-based (Feng and Hirst, 2012; Joty et al., 2015; Li et al., 2016; Liu and Lapata, 2017; Yu et al., 2018). However, these parsers analyse the RST structure of a text with the same two-stage process. The first stage is elementary discourse unit (EDU) (Marcu, 1998) segmentation in which a text is divided into clauses. This means each sentence is not obviously an EDU therefore the sentence detection is not possibly applied for EDU segmentation. The second stage is discourse relation labelling in which two consecutive discourse units are identified if they are possibly combined into a new discourse unit with a certain discourse relation.

EDU segmentation is not only important for RST parsing but also useful for identifying the answers of 'Why' questions (Azmi and Alshenaifi, 2016; Verberne et al., 2010) because there are many answers of 'why' question appeared in inter-sentential causal relations. For example, in SQuAD dataset (Rajpurkar et al., 2016), the question 'Why is Priestley usually given credit for being first to discover oxygen?' has the answer 'published his findings first' which is extracted from a single sentence "Because he published his findings first, Priestley is usually given priority in the discovery." If the sentence is not segmented into clauses, the whole sentence will be chosen as the answer. Then, the precision and F-measure of the answering method will decrease.

Although many researches in RST parsing have been conducted in many languages, the number of researches in Vietnamese RST parsing is smaller. Therefore, we would like to apply the RST framework in analysing the discourse structure of Vietnamese texts. The first problem in Vietnamese RST parsing is EDU segmentation which is the purpose of this article. In this paper, we would like to present our research in applying PhoBERT (Nguyen and Nguyen, 2020), a Vietnamese neural network language model using BERT architecture (Devlin et al., 2019), in EDU segmentation. We have chosen this language model because every word vector of a text span is computed all together with attention mechanism (Vaswani et al., 2017) in this language model. This means every word vector contains the context information with long range dependencies which improves the semantic representation of the word vector. The out-performed results of BERT in many NLP tasks have approved the effective of the computed word vectors. For automatically segmenting a Vietnamese text into EDUs, we have created an EDU annotated dataset then fine-tuned PhoBERT model to build an EDU segmentation model. For evaluation, we have tested our model and compared the results to the results of Maximum Entropy models, multilingual BERT fine-tuned model and LSTM+CRF model.

This article presents our Vietnamese EDU segmentation in five sections. Section 1 introduces our studying problem. Section 2 presents background information about EDU segmentation and how to identify an EDU in Vietnamese. Section 3 describes our approach and proposes Vietnamese EDU segmentation method. Then, the experiment and

evaluation on our method are presented in Section 4. Finally, some conclusions are given in Section 5.

2 Backgrounds

2.1 EDU segmentation

EDU is the discourse unit which is not possibly divided into smaller discourse units. In RST (Mann and Thompson, 1988), EDUs are clauses excluding clausal subjects, complements and restrictive relative clauses. The EDUs of a text are identified by an EDU segmenter. There are two approaches to build EDU segmenters: rule-based and machine learning-based.

In rule-based approach, the EDU segmenters (Le Thanh et al., 2004; Marcu, 1998) use a set of rules defined on syntactic parses of sentences and discourse cue phrases. These segmenters have F-score around 86%.

In machine learning-based approach, the EDU segmenters are built by applying the sequential labelling method. The segmenter of HILDA parser (Hernault et al., 2010) applies SVM classifier on syntactic tree and cue phrases for identifying the boundaries of EDU with F-score of 95% on 38 test documents of RST-DT dataset (Carlson et al., 2003). ToNy segmenter (Muller et al., 2019) uses multilingual BERT (Devlin et al., 2019) combining to bi-LSTM (Graves and Schmidhuber, 2005) as classifier. ToNy segmenter is trained on dependency annotated RST-DT dataset. GumDrop segmenter (Yu et al., 2019) uses NCRF++ model (Yang and Zhang, 2018) trained on POS tag and dependency annotated RST-DT dataset. The ToNy and GumDrop's F-scores are 96.04% and 95.7%, respectively.

2.2 EDU segmentation evaluation

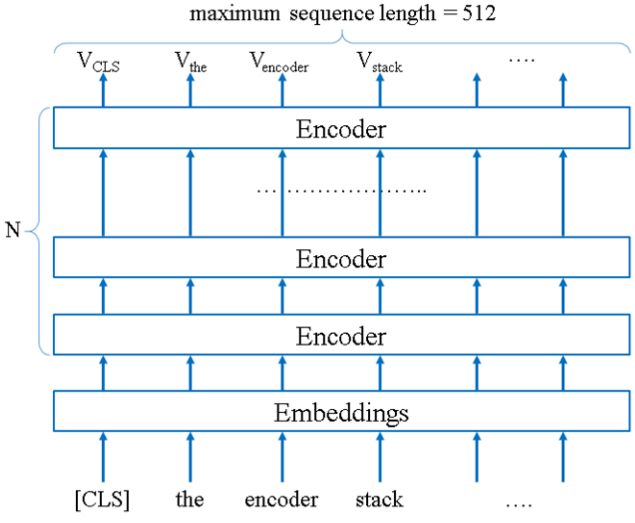
There are two metrics for evaluating the EDU segmentation. The first metric is based on the EDU boundary marks (Soricut and Marcu, 2003). These boundary marks are labelled in a sentence to split it into EDUs. The precision, recall and F-score are calculated from matching boundary marks of the prediction results and of the gold results. For example, given the EDU boundary predicting result 'they/_said/B that/_you/_could/_win/_if/B you/_wanted/_' and given the gold result 'they/_said/_that/B you/_could/_win/_if/B you/_wanted/_', there are two boundary marks in prediction result at position 2 and 7 and two boundary marks in gold result at position 3 and 7. Then, the true positive TP = 1, the false negative FN = 1 and the false positive FP = 1 yield the precision P = 0.5, the recall R = 0.5 and F-score = 0.5.

The second metric is based on EDU spans (Zeldes et al., 2019). In the previous example, the prediction result has three spans (1, 1), (2, 6) and (7, 9), the gold result has three spans (1, 2), (3, 6) and (7, 9). Then the true positive TP = 1, the false negative FN = 2 and the false positive FP = 2 yield the precision P = 0.33, the recall R = 0.33 and F-score = 0.33.

2.3 BERT pre-trained model

BERT (Devlin et al., 2019) is a deep neural network architecture based on the encoder architecture (Vaswani et al., 2017). The models pre-trained with BERT architecture are used in many NLP downstream tasks by fine-tuning on specific training data. The experiments in fine-tuning BERT model for many NLP tasks (Devlin et al., 2019), including sequential labelling, have shown that this approach has improved the performance of these tasks significantly.

Figure 1 The illustration of BERT architecture (see online version for colours)



In sequential labelling task, all BERT's outputs except the first and the last ones, as shown in Figure 1, are feature vectors $\{V_{word}\}$. Each V_{word} is corresponding to a specific word or sub-word of a text sequence (a sentence or even a paragraph). In BERT model, each V_{word} is estimated with the context of the whole sequence. This means the feature vector will be different if the context of the word changes. This is the good reason to choose BERT model for generating feature vector for each word in a sentence, we will use these feature vectors to identify the boundary of EDU because the boundary is not always based on the markers but the clause structures in the sentence.

For EDU boundary detection, we need a classifier, which is a feed-forward neural network (FFNN), to label each feature vector. For EDU segmentation, there are two labels (Muller et al., 2019) 'Begin-Seg = Yes' and '_' indicating the beginning of an EDU and the inside of an EDU respectively. This classifier will be trained jointly with BERT pretrained-model on EDU segmentation dataset.

3 Our approach

For EDU segmentation, we do not use the parse tree generated from constituent parser for identifying the EDU as in HILDA (Hernault et al., 2010) because we did not have an official published constituent parser for Vietnamese with sufficient accuracy yet and the

high quality Vietnamese constituent treebank have included only 15,535 sentences (Nguyen et al., 2018) while the Vietnamese dependency parsing has LAS score of 78.77 (Vu et al., 2018) which should be improved. Thus, we have chosen BERT architecture and fine-tune a BERT pretrained model with an EDU annotated dataset to predict the EDU boundaries in each sentence because BERT model can capture the long-range dependencies from both directions.

For Vietnamese EDU segmentation, we will detect the beginning each EDU as in EDU segmentation task in DISRPT (Zeldes et al., 2019). When we have the boundary of each EDU, we can also identify the explicit connectives which are the spans consisting of prepositions or conjunctions at the beginning and the end of each EDU. Then, we can remove the explicit connectives of each EDU to make shorter answer candidates for reason-type questions in question answering systems.

3.1 Defining EDU

We follow the EDU definition of RST (Mann and Thompson, 1988). Given $S = P_1P_2..P_n$ in which S is an independent clause and P_i, i = 1..n are direct sub-constituents of S. S is an EDU if and only if P_i is not a clause except clausal subject, complement or restrictive relative clause.

3.2 Annotating EDU

In our approach, EDU will be annotated at sentence level. Each EDU has a beginning and an end word (or punctuation) labelled with '¬BC' and 'EC', respectively. The phrases between two consecutive EDUs are connectives. For example, the sentence 'nói_tóm_lại là giá chưa duyệt thì không được bán' (in English: 'in brief, the price is not approved yet so do not sell') has two EDU 'giá chưa duyệt ' and ' không được bán ' (in English: ' the price is not approved yet' and 'do not sell'), then it will be annotated as in Figure 2.

Figure 2 An example of EDU annotation format

```
1 Ông/BC Đoàn_Nguyên_Đức/Nr :/PU '/PU Sự_việc/Nn không_chỉ/Cp
  gây/Vv phương_hại/Nn cho/Cs Công_Phượng/EC '/PU ./PU
2 Sau/BC những/Nq ngày/Nu làm/Vv dấy/Vv lên/R sự/Ncs
  quan_tâm/Vv của/Cs người/Nn hâm_mộ/Vv cả/Nw nước/Nn ,/PU
  vụ/Nc tuổi_tác/Nn của/Cs tuyển_thủ/Nn U._19/Nr Việt_Nam/Nr
  Công_Phượng/Nr đã/R đến/Vv lúc/Nt khép/Vv lại/EC ./PU
3 Gọi/BC điện/Nn cho/Cs chúng_tôi/EC ông_bầu/BC
  Đoàn_Nguyên_Đức/Nr bực_tức/Vv nói/Vv :/PU "/PU Tôi/Pp không/R
  hiểu/Vv người_ta/Pp dựng/Vv nên/R chuyện/Nn vì/Cs mục_đích/Nn
  gì/EC ./PU
4 Rõ_ràng/BC sự_việc/Nn không_chỉ/Cp gây/Vv phương_hại/Nn
  cho/Cs Công_Phượng/Nr ,/PU Học_viện/Nn HAGL/Nr mà/Cp còn/R
  gây/Vv nghi_ngờ/Nn về/Cs sự/Ncs minh_bạch/Aa của/Cs các/Na
```

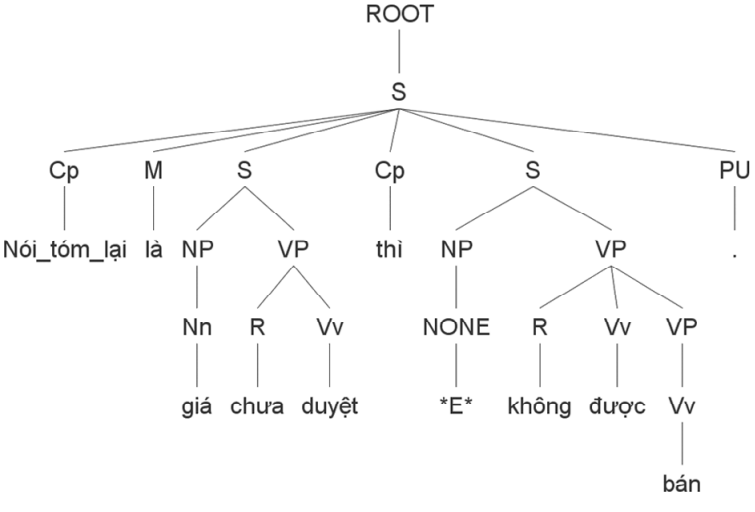
By using this annotating format, we can identify the two EDUs and two connectives 'nói_tóm_lại là' and 'thì' (in English: 'in brief' and 'then')

3.3 Building EDU segmentation dataset

Currently, we do not have any EDU segmentation dataset in Vietnamese. Therefore, we have built one from NIIVTB (Nguyen et al., 2018), which is a Vietnamese constituent treebank, by exploiting the patterns of independent clauses in parse trees. There is an important difference between Vietnamese and English in restrictive clause. In Vietnamese, restrictive clauses and independent clauses are not different. For example, the Vietnamese sentence 'tôi thấy Hoa đang qua đường' and the strictly translation into English 'I saw Hoa crossing the street' have different structures. In Vietnamese sentence, although 'Hoa đang qua đường' (in English: 'Hoa crossing the street') has the function as a restrictive clause, it is also a complete sentence. However, in English sentence, 'Hoa crossing the street' cannot stand alone as a sentence. Therefore, if a text span has an independent clause structure, it is possibly not an EDU in Vietnamese.

For identifying EDU in Vietnamese, we have considered 500 randomly selected sentences in NIIVTB to find the constituent structure patterns of EDU. In NIIVTB, constituents which have clause structure are labelled with 'S', 'SPL' and 'SQ' meaning sentence, special sentence and question respectively (Nguyen et al., 2018). Therefore, we have only considered structures including 'S', 'SPL' or 'SQ' for finding EDU patterns. The first pattern is that an EDU is a constituent labelled 'S' which is directly a part of a constituent labelled 'S' as shown in Figure 3. The two constituents 'giá chưa duyệt' (in English: 'the price is not approved yet') and 'không được bán' (in English: 'do not sell') have label 'S' and they are direct sub-constituents of the label 'S' constituent which is the whole sentence. Therefore, they are identified as EDUs.

Figure 3 The parse tree of the sentence 'nói_tóm_lại là giá chưa duyệt thì không được bán' (in English: 'in brief, the price is not approved jet so do not sell')



The second pattern is that an EDU is a constituent labelled 'SQ' which is directly a part of a constituent labelled 'S' as shown in Figure 4. There is an interrogative sentence labeled 'SQ' 'không biết cả ba cha_con anh có vượt qua nổi?' (in English: 'do not know that are they and their father able to overcome?') which is a part of the whole sentence therefore it is an EDU.

Figure 4 The parse tree of the sentence ‘Giờ_đây, con đường tiếp_tục đến trường để thực_hiện khát_vọng cháy_bỏng cho ngày_mai vẫn còn lắm gian_nan, không biết cả ba cha_con anh có vượt qua nổi?’ (in English: ‘Now, the way to keep learning for accomplishing the burning desire of the future is still very difficult, do not know that are they and their father able to overcome?’)

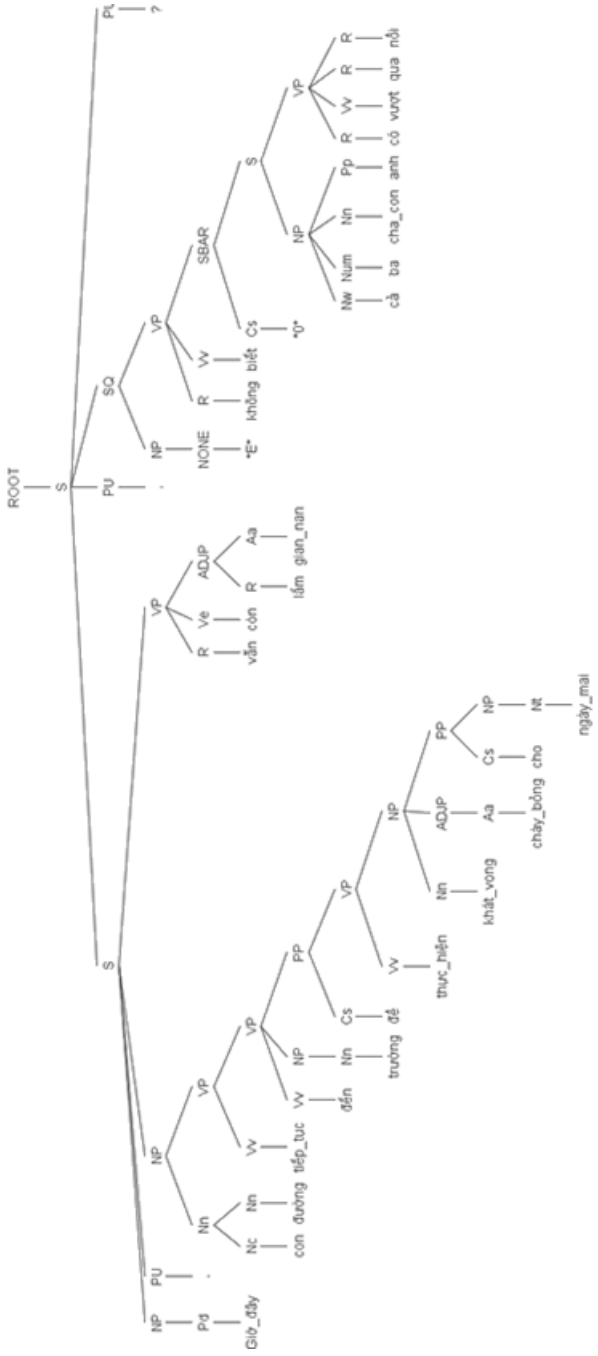


Figure 5 The parse tree of the sentence ‘Tôi già rồi nên kém, chứ mấy thằng ở khu dưới có ngày bắt được 30–50 con’ (in English: ‘I am old already, so I am weak, but the downtown guys sometimes caught 30–50 ones a day’)

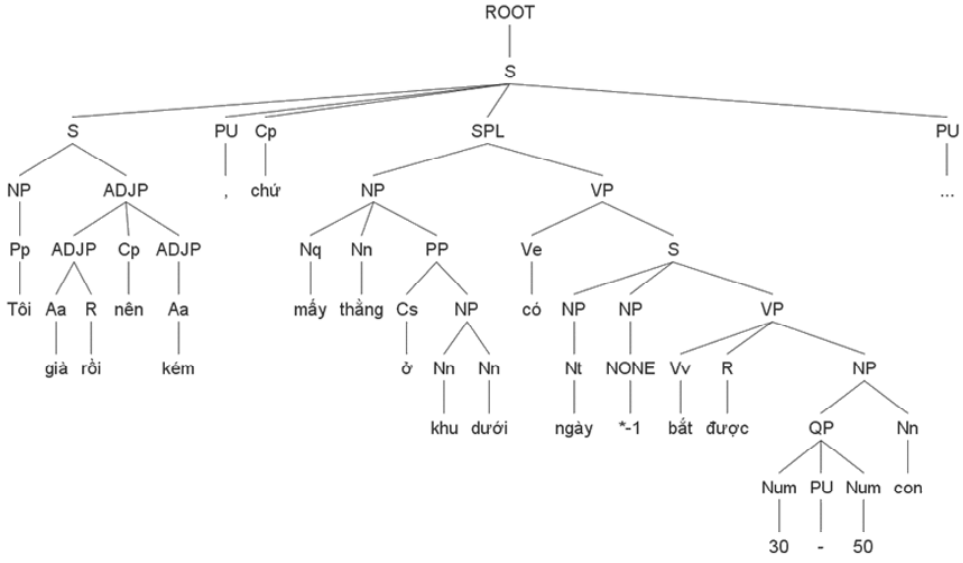
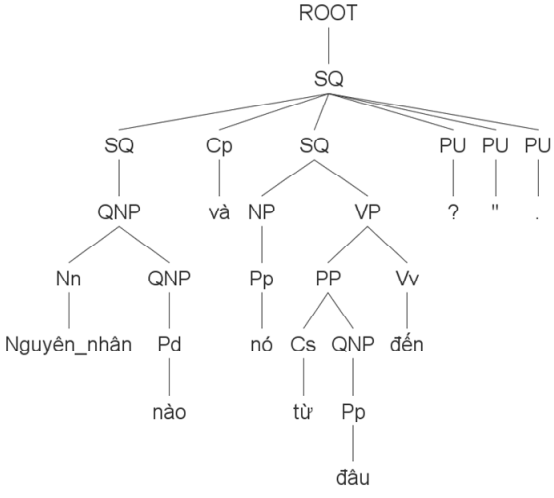


Figure 6 The parse tree of the sentence ‘Nguyên_nhân nào và nó từ đâu?’ (in English: ‘What is the reason and where does it arise?’)



The third pattern is that an EDU is a constituent labelled ‘SPL’ which is directly a part of a constituent labelled ‘S’ as shown in Figure 5. The constituent ‘mấy thằng ở khu dưới có ngày bắt được 30–50 con’ (in English, it means ‘the downtown guys sometimes caught 30 – 50 ones a day’ is a special sentence because there are a verb (‘có’) and a noun (‘ngày’) between the subject and the main verb (‘bắt’). The word-by-word translation of ‘có ngày’ is ‘there are days’ but it means ‘sometimes’ in this context. The ‘SPL’

constituent in Figure 5 is an EDU because it is a complete sentence and is not possibly split into other EDUs.

Figure 7 The parse tree of the sentence ‘‘Bắt kiểu này ngày bắt được bao nhiêu con?’, tôi hỏi’ (in English: ‘‘How many ones you caught a day by using this way ‘, I asked’)

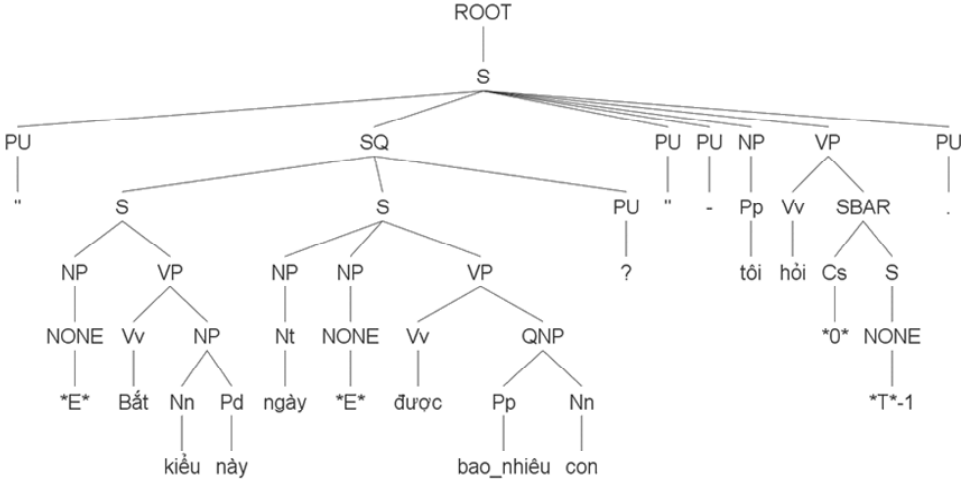
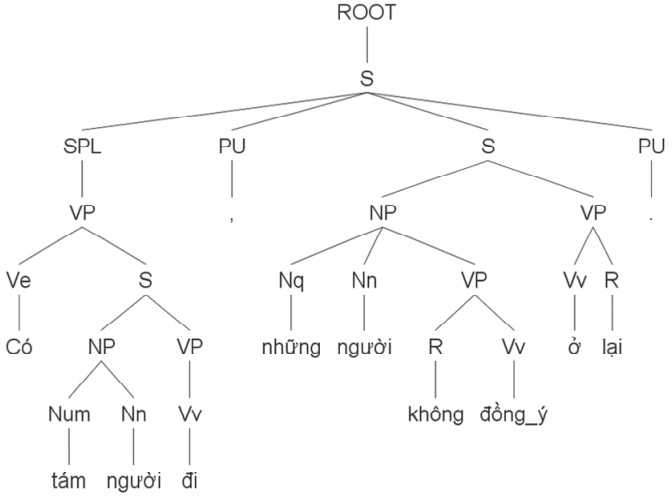


Figure 8 The parse tree of the sentence ‘Có tám người đi, những người không đồng_ý ở lại’ (in English: ‘There are eight people moved, people who was not agreed stay back’) in which a ‘S’ constituent is a restrictive relative clause.



The fourth pattern is that an EDU is a constituent labelled ‘SQ’ which is directly a part of a constituent labelled ‘SQ’ as shown in Figure 6. The whole sentence is a compound sentence in which two interrogative sentences ‘Nguyên_nhân nào’ (in English: ‘what is the reason’) and ‘nó từ đâu’ (in English: ‘where does it arise’) combine therefore each of these interrogative sentences is an EDU.

The fifth pattern is that an EDU is a constituent labelled ‘S’ which is directly a part of a constituent labelled ‘SQ’ as shown in Figure 7. The ‘SQ’ constituent is an interrogative sentence composed of two simple sentence ‘Bắt kiểu này’ (in English, it means ‘using this way ‘) and ‘ngày bắt được bao_nhiêu con’ (in English: ‘How many ones you caught a day’) therefore the two simple sentences are EDUs.

We have not considered constituents as EDUs only by their label of ‘S’, ‘SPL’ or ‘SQ’ because they are possibly restrictive relative clauses in many cases. Figure 8 illustrates an example where a ‘S’ constituent is a restrictive relative clause. The sentence in Figure 8 has the ‘S’ constituent ‘tám người đi’ (in English: ‘eight people moved’) which is both a simple sentence and a restrictive relative clause therefore it is not possibly considered as an EDU. There is a similar case in Figure 5 where the clause ‘ngày bắt được 30–50 con’ (in English, it means ‘caught 30–50 ones a day’) cannot be separated from the previous constituent ‘mấy thằng ở khu dưới’ (in English, it means ‘the downtown guys’).

From the above EDU recognition patterns, we have generalised them to a set of rule and proposed an algorithm for building EDU segmentation dataset in Algorithm 1. In Algorithm 1, flatten is a function breaking a parse tree in words with left-to-right order and every word also has a POS tag and node level. There is also Place_hoder constant which is the character ‘*’ used in NIIVTB for presenting missing arguments of verb frames. The flatten is the implementation of the Algorithm 2.

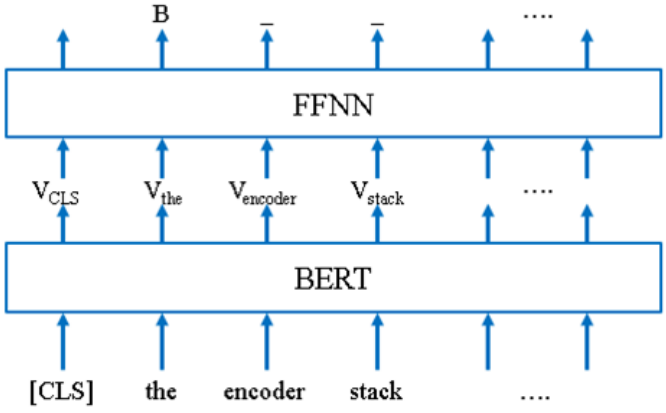
Algorithm 1 Programmatically building EDU segmentation dataset

```
Input: T, a manual parse tree in Vietnamese
Output: Ann, EDU annotated sentence
1   flatten(T, 0, WORDs, POSs, LEVELs)
2   APOSs ← POSs
3   for i = 1 to |WORDs|-1
4     if LEVELs[i-1] != LEVELs[i] then
5       APOSs[i] ← 'BC'
6   for i = |WORDs|-1 downto 1
7     if POSs[i-1] is Conjunction, Preposition
            or Punctuation then
8       APOSs[i] ← POSs[i]
9       APOSs[i-1] ← 'BC'
10  for i = 0 to |WORDs|-2
11    if WORDs[i] is Place_holder then
12      APOSs[i+1] ← 'BC'
13  Ann □ □
14  for i = 0 to |WORDs|-1
15    if WORDs[i] is not Place_holder then
16      Ann ←Ann ∪ {WORDs[i]+'/'+APOSs[i]}
17  return Ann
```

Algorithm 2 Flatten constituent parse tree for EDU detection

```
Input:
    • T, a manual parse tree in Vietnamese
    • d, the level of the current node by EDU span
Output:
    • WORDs, words in parse tree
    • POSs, POS tags of words
    • LEVELs, node level of word by EDU span
1   nodes ← T.subTrees()
2   for i = 0 to |nodes|
3     if nodes[i].isLeaf() then
4       WORDs ← WORDs ∪ {nodes[i].word}
5       POSs ← POSs ∪ {nodes[i].tag}
6       LEVELs ← LEVELs ∪ {d}
7     else
8       if T is Sentence, Question or Special Question then
9         if nodes[i] is Sentence, Question,
            or Special Question then
10          flatten(nodes[i], d+1, WORDs, POSs, LEVELs)
11          Continue
12      flattern(nodes[i], d, WORDs, POSs, LEVELs)
```

Figure 9 EDU boundary classification model (see online version for colours)



3.4 The EDU boundary classification model

Our EDU boundary classification model is based on BERT model. It has two tiers as in Figure 9. The first tier is BERT pretrained model which generates feature vectors for

each word of a sentence. The second tier is a FFNN. The number of inputs of the FFNN is equal to the dimension of feature vectors (768 with BERT base and 1,024 with BERT large). This FFNN will be trained when fine-tuning BERT pretrained model to predict the label of each word by calculating its feature vector. Although this FFNN process feature vectors one by one without using nearby feature vectors, the context information has already been encoded in the processing feature vector through attention layers of encoder blocks in BERT model (Vaswani et al., 2017). Therefore, we can use a simple FFNN for an effective classifier.

4 Experiments and evaluation

We have conducted two experiments to choose the effective tag set used in EDU segmentation dataset and to show the effective of PhoBERT fine-tuned model in EDU segmentation in Vietnamese. In the first experiment, we have trained Maximum Entropy models and our models on two version of EDU segmentation dataset which are tagged with two tags 'O' and 'BC' and with the POS tags of NIIVTB (Nguyen et al., 2018) and tag 'BC'. Then we have tested the models to choose the effective tag set. In the second experiment, we have trained LSTM+CRF model by using NCRF++ (Yang and Zhang, 2018) and fine-tuned multilingual BERT model (Devlin et al., 2019) on the selected EDU segmentation dataset then we have compared the results of these two models to the result of our model.

4.1 Experiment dataset

Our dataset has been built from 9,046 sentences of NIIVTB with the Algorithm 1. Because NIIVTB did not provide the original text, we had to crawl the web pages and extract the main content of these web pages to rebuild the treebank. However, the sentences in these web pages did not match the annotation entirely thus we have just recovered 9,046 parse trees. Our dataset has been divided into train dataset with 8,143 sentences and test dataset with 904 sentences. The statistics of our dataset are shown in Table 1.

Table 1 EDU segmentation dataset

	Train	*Test*	*Total*
#Sentence	8,142	904	9,046
#EDU	12,910	1,412	14,322
#Max. sentence length in word	113	82	_
#Min. sentence length in word	2	3	_
#Ave. sentence length in word	22	22	_

We have annotated our dataset in two versions. In the first version, named EDU-UNI, we have used two labels 'BC' and '¬O' indicating the beginning of a new EDU and the others, respectively. In the second version, named EDU-ALL, we have used all POS tags of NIIVTB and 'BC' tag. We have created them to test if the number of labels affects the performance of EDU segmentation.

4.2 Experiment settings

For implementing the EDU boundary classification model, we have used the RoBERTa from Huggingface library (Wolf et al., 2020) to implement BERT tier and used PhoBERTbase (Nguyen and Nguyen, 2020) as pretrained model for fine-tuning on EDU segmentation datasets. We have also used VnCoreNLP (Vu et al., 2018) for segmenting Vietnamese words. There are two settings for the FFNN tier corresponding to the two EDU segmentation datasets. Therefore, we have two models for EDU segmentation:

1 UNISEG: the FFNN tier has 768 inputs for each word or sub-word's feature vector and 3 outputs for 'BC', '¬O' tags and a '<pad>' tag indicating sub-word. This model has been fine-tuned on EDU-UNI dataset.

2 ALLSEG: the FFNN tier has 768 inputs for each word or sub-word's feature vector and 36 outputs which are all POS tags used in NIIVTB and 'BC' and '<pad>'. This model has been fine-tuned on EDU-ALL dataset.

The two models has been fine-tune with learning parameters described in Devlin et al. (2019). We have fine-tuned UNISEG and ALLSEG in three epochs and four epochs respectively to avoid overfitting. We have also fine-tune them in more epochs, but the performance has decreased.

We have also implemented three Maximum Entropy models with different feature selections. The settings of these models are shown in Table 2. For implementing these models, we have used Apache OpenNLP library (https://opennlp.apache.org) with GIS (Curran and Clark, 2003) and 100 iterations. We have used maximum entropy models for EDU segmentation because they are much simpler than BERT-based models. If the performance of maximum entropy models is slightly lower than of BERT-based models, we can use maximum entropy models for EDU segmentation in practice.

Table 2 EDU boundary classification model settings using Maximum Entropy.

Name	*Feature (w: current word, w – i: previous i word, w + i: next i word)*
ME-1	w – 1, w, w + 1
ME-2	w – 2, w – 1, w, w + 1, w + 2
ME-3	w – 3, w – 2, w – 1, w, w – 1, w – 2, w + 3

Algorithm 3 Predicting EDUs from sequential labelling result

```
Input: POS, a list of labels for each word of a sentence
Output: SPAN, a list of EDU spans of the sentence
1  bm ← 0, SPANS ← ∅
1  for i = 0 to |POS|-1
2    if POS[i] == 'BC' then
3         SPANS ← SPANS ∪ {(bm, i-1)}
4         bm ← 0
5  SPANS ← SPANS ← {(bm, i-1)}
6  return SPANS
```

After training these models, we have used them to predict the positions of 'BC' tag in each test sentences. Then the predicted each EDU span is identified with the Algorithm 3.

For considering the effectiveness of our model, we have implemented a long-short term memory with conditional random field (LSTM+CRF) model and a multilingual BERT fine-tuned model for experiments. For LSTM+CRF model, we have used NCRF++ toolkit (https://github.com/jiesutd/NCRFpp) with the following settings: word embeddings extracted from word embedding layer of PhoBERT, 768 in word embedding size, no character embeddings, 20 iterations. The LSTM+CRF has been trained on the selected EDU segmentation dataset with a minor adjustment in which words are converted into sub-words to use PhoBERT word embedding layer because PhoBERT's word embeddings have been trained on large data. The test results of LSTM+CRF have also been converted into words from sub-words for comparison. For multilingual BERT fine-tuned model, called mBERT, we have used pretrained BERT multilingual base cased model (Devlin et al., 2019) in the same architecture to our model. The mBERT model has been trained on the selected dataset with a minor adjustment in which Vietnamese words are converted to morphemes. The test results of mBERT does not need a word converting post-processor.

4.3 Tag set selection results

We have conducted the EDU boundary classification with the above settings. Table 3 shows the accuracy of the sequential labelling models on EDU-ALL and EDU-UNI datasets. In Table 3, the BERT-based models are outperformed the Maximum Entropy models. The accuracy of 0.93 shows that our fine-tuning results reach the SOTA of Vietnamese POS tagging with accuracy of 0.967 reported (Nguyen and Nguyen, 2020).

Table 3 The accuracy of the sequential labelling models

Model	*Accuracy*	
	EDU-ALL	*EDU-UNI*
ME-2	0.8460	0.8555
ME-3	0.8283	0.7852
ME-4	0.8283	0.7852
UNISEG	_	0.9884
ALLSEG	0.9338	_

Table 4 EDU segmentation results by using maximum entropy models and BERT-based models

Model	*Tag-based F1*		*Span-based F1*	
	EDU-ALL	*EDU-UNI*	*EDU-ALL*	*EDU-UNI*
ME-2	0.3000	0.2325	0.4435	0.1624
ME-3	0.2991	0.1734	0.4037	0.0808
ME-4	0.2991	0.1734	0.4037	0.0808
UNISEG	_	0.7709	_	0.8000
ALLSEG	0.7428	_	0.7905	_

The results of EDU segmentation using these models are shown in Table 4. In Table 4, the EDU segmentation using maximum entropy models are quite low although their results on POS tagging are pretty good (accuracy of 0.846). There are two reasons for these results. Firstly, there are conjunctions, prepositions and punctuations which are tagged with different labels in EDU segmentation datasets because their labels are identified by the structure of the sentences, not just some words in a sliding window with size 7, 5 or 3. Secondly, maximum entropy models have not captured the dependencies between words in a sentence therefore they cannot model the structure of the sentence.

Although our BERT-based models can capture the dependencies between words in a sentence, their span-based F1 scores about 0.8 need to be improved to apply in practical applications. The span-based F1 scores are not very high because our EDU segmentation datasets contain different annotations on a same word or punctuation. These problems have been shown in Section 3.3 in which there are many constituents having syntactic structure of sentence, but these constituents are restrictive relative clauses. Therefore, some beginning EDU marks will be inserted at the beginning of clauses in some cases, but they are not inserted at the beginning of the similarly structured clauses in other cases.

In this experiment, we have found that the EDU annotation with two tags 'O' and 'BC' is more effective than the annotation with the POS tags of NIIVTB and 'BC' tag. Therefore, we have chosen two tags EDU segmentation dataset for training EDU segmentation model.

4.4 Effective model selecting results

In this experiment, we have trained LSTM+CRF and mBERT model on two tags EDU segmentation dataset and compared their results to our results. The experiment results are shown in Table 5.

Table 5 EDU segmentation results by using LSTM+CRF, mBERT and UNISEG models

Model	*Tag-based F1*	*Span-based F1*
LSTM+CRF	0.2323	0.5945
mBERT	0.7174	0.3703
UNISEG	0.7709	0.8000

The results in Table 5 show that UNISEG model has out-performed LSTM+CRF and mBERT models in EDU segmentation with our dataset. The mBERT model has low results because mBERT uses multilingual BERT pretrained model which might not effectively capture the context information when computing word vectors. In Table 5, LSTM+CRF model has strange results that the span-based F1 score is double tag-based F1 score. We have investigated the test set and found that there are 633 sentences which are also EDU. Therefore, the 'O' tag biased prediction of LSTM+CRF model has increased the span-based F1 while tag-based F1 is low.

5 Conclusions and future works

In this paper, we have presented our research in building an EDU segmentation model for Vietnamese text. Our approach is to apply BERT architecture for the sequential labelling problem to propose the architecture for EDU boundary classification model, then to build a model by fine-tuning on EDU segmentation dataset. Because we have not had any published EDU segmentation dataset, we have inspected the parse trees of NIIVTB to find out the syntactic patterns of EDU and proposed an algorithm for converting the manual parse trees into EDU segmentation format used in DISRPT EDU segmentation share task.

For evaluation, we have conducted the EDU segmentation experiments with different model settings by training or fine-tuning the models on two datasets with two-tag annotation and 37-tag annotation to choose the effective tag set, then we have compared our model to LSTM+CRF model and multilingual BERT fine-tuned model to show the effectiveness. The experiment results show that our BERT-based model, using PhoBERT pretrained model, can segment Vietnamese sentences into EDUs with F1 score of 0.8 when using training dataset with two label 'BC' and 'O'. Our model is possibly used in practical tasks however it should be improved for better results.

In future, we need a large and high-quality Vietnamese EDU annotation dataset for improving the EDU segmentation model. Then, we will apply a SOTA sequential labelling architecture to fine-tune an EDU segmentation model on this dataset.

References

Azmi, A.M. and Alshenaifi, N.A. (2016) 'Answering Arabic why-questions: baseline vs. RST-based approach', *ACM Transaction on Information Systems*, Vol. 35, No. 1, p.19.

Carlson, L., Marcu, D. and Okurowski, M.E. (2003) 'Building a discourse-tagged corpus in the framework of rhetorical structure theory', in van Kuppevelt, J. and Smith, R.W. (Eds.): *Current and New Directions in Discourse and Dialogue*, Vol. 22, pp.85–-112, Springer Netherlands, Dordrecht.

Cervone, A. (2020) *Computational Models of Coherence for Open-Domain Dialogue*, PhD dissertation, University of Trento,

Chernyavskiy, A. and Ilvovsky, D. (2020) 'Recursive neural text classification using discourse tree structure for argumentation mining and sentiment analysis tasks', Paper presented at the *International Symposium on Methodologies for Intelligent Systems*.

Curran, J.R. and Clark, S. (2003) 'Investigating GIS and smoothing for maximum entropy taggers', Paper presented at the *10th Conference on European Chapter of the Association for Computational Linguistics*.

Devlin, J., Chang, M-W., Lee, K. and Toutanova, K. (2019) 'BERT: pre-training of deep bidirectional transformers for language understanding', Paper presented at the *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Feng, V.W. and Hirst, G. (2012) 'Text-level discourse parsing with rich linguistic features', Paper presented at the *Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea

Graves, A. and Schmidhuber, J. (2005) 'Framewise phoneme classification with bidirectional LSTM and other neural network architectures', *Neural Networks*, Vol. 18, Nos. 5–6, pp.602–610.

Hernault, H., Prendinger, H., du Verle, D.A. and Ishizuka, M. (2010) 'HILDA: a discourse parser using support vector machine classification', *Dialogue & Discourse*, Vol. 1, No. 3, pp.1–33.

Joty, S., Carenini, G. and Ng, R.T. (2015) 'Codra: a novel discriminative framework for rhetorical analysis', *Computational Linguistics*, Vol. 3, No. 41, pp.385–435.

Le Thanh, H., Abeysinghe, G. and Huyck, C. (2004) 'Automated discourse segmentation by syntactic information and cue phrases', Paper presented at the *International Conference on Artificial Intelligence and Applications*, Innsbruck, Austria.

Li, Q., Li, T. and Chang, B. (2016) 'Discourse parsing with attention-based hierarchical neural networks', Paper presented at the *Conference on Empirical Methods in Natural Language Processing*.

Liu, Y. and Lapata, M. (2017) 'Learning contextually informed representations for linear-time discourse parsing', Paper presented at the *Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark.

Mann, W.C. and Thompson, S.A. (1988) 'Rhetorical structure theory: toward a functional theory of text organization', *Text*, Vol. 8, No. 3, pp.243–281.

Marcu, D. (1997) 'The rhetorical parsing of unrestricted natural language texts', Paper presented at the *European Chapter of the Association for Computational Linguistics*.

Marcu, D. (1998) *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*, PhD, University of Toronto,

Muller, P., Braud, C. and Morey, M. (2019) 'ToNy: contextual embeddings for accurate multilingual discourse segmentation of full documents', Paper presented at the *Workshop on Discourse Relation Parsing and Treebanking*.

Nguyen, D.Q. and Nguyen, A.T. (2020) 'PhoBERT: Pre-trained language models for Vietnamese', Paper presented at the *Conference on Empirical Methods in Natural Language*.

Nguyen, Q.T., Miyao, Y., Le, H.T. and Nguyen, N. T. (2018) 'Ensuring annotation consistency and accuracy for Vietnamese treebank', *Language Resources and Evaluation*, Vol. 52, No. 1, pp.269–315.

Polanyi, L., Culy, C., Van Den Berg, M., Thione, G.L. and Ahn, D. (2004) 'A rule based approach to discourse parsing', Paper presented at the *Workshop on Discourse and Dialogue at HLT-NAACL*.

Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P. (2016) 'SQuAD: 100,000+ questions for machine comprehension of text', Paper presented at the *Empirical Methods in Natural Language Processing*, Austin, Texas.

Soricut, R. and Marcu, D. (2003) 'Sentence level discourse parsing using syntactic and lexical information', Paper presented at the *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Subba, R. and Di Eugenio, B. (2009) 'An effective discourse parser that uses rich linguistic information', Paper presented at the *Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Taboada, M. and Mann, W.C. (2006) 'Applications of rhetorical structure theory', *Discourse Studies*, Vol. 8, No. 4, pp.567–588.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N. and Polosukhin, I. (2017) 'Attention is all you need', Paper presented at the *Neural Information Processing Systems*, Long Beach, CA, USA.

Verberne, S., Boves, L., Oostdijk, N. and Coppen, P-A. (2010) 'What is not in the bag of words for why-QA?', *Computational Linguistics*, Vol. 36, No. 2, pp.229–245.

Vu, T., Nguyen, D.Q., Nguyen, D.Q., Dras, M. and Johnson, M. (2018) 'VnCoreNLP: a Vietnamese natural language processing toolkit', Paper presented at the *Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, New Orleans, Louisiana.

Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A. et al. (2020) 'Transformers: state-of-the-art natural language processing', Paper presented at the *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Yang, J. and Zhang, Y. (2018) 'NCRF++: an open-source neural sequence labeling toolkit', Paper presented at the *ACL 2018, System Demonstrations*, Melbourne, Australia.

Yu, N., Zhang, M. and Fu, G. (2018) 'Transition-based neural RST parsing with implicit syntax features', Paper presented at the *International Conference on Computational Linguistics*.

Yu, Y., Zhu, Y., Liu, Y., Liu, Y., Peng, S., Gong, M. and Zeldes, A. (2019) 'GumDrop at the DISRPT2019 shared task: a model stacking approach to discourse unit segmentation and connective detection', Paper presented at the *Workshop on Discourse Relation Parsing and Treebanking*.

Zeldes, A., Das, D., Maziero, E.G., Antonio, J. and Iruskieta, M. (2019) 'The DISRPT 2019 shared task on elementary discourse unit segmentation and connective detection', Paper presented at the *Workshop on Discourse Relation Parsing and Treebanking*.

Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 6550871, 16 pages
https://doi.org/10.1155/2021/6550871

Research Article

Building a Discourse-Argument Hybrid System for Vietnamese Why-Question Answering

Chinh Trong Nguyen [1] **and Dang Tuan Nguyen** [2]

[1]*University of Information Technology, VNU-HCM, Ho Chi Minh City, Vietnam*
[2]*Saigon University, Ho Chi Minh City, Vietnam*

Correspondence should be addressed to Dang Tuan Nguyen; dangnt@sgu.edu.vn

Received 12 October 2021; Accepted 3 December 2021; Published 28 December 2021

Academic Editor: Thippa Reddy G

Recently, many deep learning models have archived high results in question answering task with overall F_1 scores above 0.88 on SQuAD datasets. However, many of these models have quite low F_1 scores on why-questions. These F_1 scores range from 0.57 to 0.7 on SQuAD v1.1 development set. This means these models are more appropriate to the extraction of answers for factoid questions than for why-questions. Why-questions are asked when explanations are needed. These explanations are possibly arguments or simply subjective opinions. Therefore, we propose an approach to finding the answer for why-question using discourse analysis and natural language inference. In our approach, natural language inference is applied to identify implicit arguments at sentence level. It is also applied in sentence similarity calculation. Discourse analysis is applied to identify the explicit arguments and the opinions at sentence level in documents. The results from these two methods are the answer candidates to be selected as the final answer for each why-question. We also implement a system with our approach. Our system can provide an answer for a why-question and a document as in reading comprehension test. We test our system with a Vietnamese translated test set which contains all why-questions of SQuAD v1.1 development set. The test results show that our system cannot beat a deep learning model in F_1 score; however, our system can answer more questions (answer rate of 77.0%) than the deep learning model (answer rate of 61.0%).

1. Introduction

Question answering is a branch of information retrieval. Many early question answering systems used named entity extraction models to extract answer candidates from the retrieved documents; then, they selected the best five answer candidates for each question. These systems were designed for answering factoid questions; thus, their answers were usually nominal phrases of place, time, person's name, etc. These systems did not answer why-question well because the answers of why-questions are not always nominal phrases. Answering why-questions is a big question for not only many early systems but also recent deep learning models. According to the results of Microsoft Research Asia's R-NET+ (ensemble) model [1], Alibaba iDST NLP's SLQA+ (ensemble) [2], Singapore Management University's Match-LSTM (boundary + ensemble) [3], and Google AI Language's BERT (ensemble) [4] model on SQuAD development set v1.1 published in SQuAD website (https://rajpurkar.github.io/SQuAD-explorer/), we have calculated the why-question F_1 scores of these models which are shown in Table 1. We can see that the F_1 scores of why-questions are lower than those of all questions by about 23% in all models. We exploited the SQuAD v1.1 dataset and found that the number of samples with why-question is only about 2700 in training set. This means those models were mostly trained for answering factoid questions.

Why-question answering is an interesting problem. Like how-questions or definition questions, answering why-questions needs a different method from the methods of applying information extraction on information retrieval results. The answers of why-questions usually occur in the form of explanations. The explanations may be arguments or opinions. The important difference between an argument

Table 1: The results of some deep learning models on SQuAD development set v1.1.

Model	F_1	
	All questions	Why-questions
R-NET+ (ensemble)	88.48%	66.90%
SLQA+ (ensemble)	88.38%	65.69%
Match-LSTM (boundary + ensemble)	76.76%	56.95%
BERT (ensemble)	**92.2%**	**69.66%**

and an opinion is that an argument is either true or false while an opinion is an expression about what a person thinks [5]. Apart from that, many arguments are possibly presented with the same rhetorical structures [6] as opinions. For example, "*The price of book is rising because we have to pay 50$ for it when it was 40$ last week*" is an argument because we can judge whether it is true or false, while "*I love this book because its cover is nice*" is just an opinion and we cannot judge it. According to our surveys, the research on why-question answering is presented in Table 2.

Verberne's why-question answering method is one of the early studies on rhetorical structure approach [7–12]. According to this method, the relevant documents of a why-question are retrieved; then, all text spans which are relevant to the question are selected as answer candidates. These candidates will have additional scores if they are presented in one of six rhetorical structures named Background, Circumstance, Purpose, Result, Cause, and Motivation [13]. In preliminary research on why-question answering [11, 12], Verberne has shown that rhetorical structure of documents plays an important role in answer selection. However, the full rhetorical parses of documents were not easy to obtain; thus, a list of cue words has been used [9, 10] for rhetorical features. The output of this method is a list of passages because it was found that the answer of a why-question may be a passage. Verberne's method has the MRR@150 score of 0.34 with a test set including 187 why-questions.

In the research of why-question answering for Japanese, Higashinaka and Isozaki's method is also a rhetorical structure approach [14]. In this method, Higashinaka and Isozaki use a classifier for identifying which sentence or paragraph has a causal relation to the why-question. Then, the highest-ranking ones are chosen as the final answer. The causal classifier is used because there are many causal structures that do not use any cue word. In other words, a cue word-based feature may miss many causal structures. Therefore, the authors have collected a causal dataset [15] for training a SVM classifier which does not rely on cue words. This method has the MRR@20 score of 0.339 on a Japanese why-question test set. This result cannot compare to Verberne's result because they are not evaluated with the same test set.

The causal classification is also the approach of Oh et al. to why-question answering [16–19]. In early work of Oh et al. [18], the authors solve the problem of causal relation recognition as a sequential labeling problem. They use five tags, namely, B-C, I-C, B-E, I-E, and O, for annotating the beginning of causal part, the inside of causal part, the beginning of effect part, the inside of effect part, and the outside in a text span, respectively. For causal relation recognition, the authors train a CRF (conditional random field) classifier and use it for predicting the causal and effect parts of causal relations. The extracted causal parts are the answer candidates, and they are selected to choose the final answers. This method can find the answers with the precision P@1 score of 41.8% on their developed dataset named WhySet. This result cannot also compare to Higashinaka's and Verberne's results because they use different test sets and evaluation measures. In research on improving why-question answering, Oh et al. also use this causality recognizer to build a large training set for improving the performance of a question-answer classifier [17]. This question answering classifier is used for reranking the answer candidates. In [17], the system using this reranking method has the precision P@1 score of 50% which is higher than that in their previous work on the WhySet dataset. In [19], the authors also use the causality recognizer for extracting causal-effect fragments from 4 billion web pages. These fragments are the references for evaluating the relevance of answer candidates to a why-question. The authors use a multicolumn CNN (convolutional neural network) model called CA-MCNN [19] whose input is a four-tuple containing the why-question, an answer candidate, the causal-effect fragments of the answer candidate, and a reference causal-effect fragment which is the most appropriate to the answer candidate. This method has the precision P@1 score of 54% on the WhySet dataset. The newest work of Oh et al. proposes a GAN-like neural network architecture, which is inspired by generative adversarial nets (GAN) [20], for answer score computation. This network receives a passage and a why-question as input. Then, it generates the compact answer representation of the passage, and the representations of the question and the passage. After that, it computes the answer score of the passage using the representations of the compact answer, the why-question, and the passage [16]. The why-QA system of Oh et al. using this GAN-like neural network has the F_1 score of 54.8% on the WhySet dataset. When applying this framework to English question answering, the F_1 scores are from 49.9% to 65.3% and the EM (exact match) scores are from 42.9% to 59.7% on many English datasets including TriviaQA [21]. These datasets contain many question types including why-questions.

The above works show that why-question answering needs a different approach from that of answering factoid questions. The reasonable approach is to select the answers from rhetorical structure parses of answer passages. However, parsing full rhetorical structure of a paragraph or a document is still a big question; thus, these methods focus on recognizing causal-effect relation in the answer passages and use this recognition result as a feature for reranking answer passages. Therefore, we propose our why-question answering method which focuses on five rhetorical relation types, namely, Cause, Result, Purpose, Circumstance, and Motivation [13], and the arguments existing in document for selecting the answers for why-question in Vietnamese. For recognizing the discourse relation of those five types, we

Table 2: Research works on why-question answering.

Author	Year	Methodology	Dataset	Result
Verberne	2006–2010	IR + RST relation classification	Selected 186 English why-questions on **INEX** corpus	MRR@150 = 0.34
Higashinaka and Isozaki	2008	IR + causal relation classification using SVM	Dataset developed in Japanese	MRR@20 = 0.339
Oh et. al.	2013	IR + causal extraction using CRF	**WhySet**, dataset developed in Japanese	P@1 = 41.8%
	2016	IR + causal extraction using CRF, augmented by adding more training data	**WhySet**	P@1 = 50%
	2017	IR + causal extraction using CRF, answer selection using CNN network	**WhySet**	P@1 = 54%
	2019	IR + GAN-like network (GAN–generative adversarial network)	**WhySet**	P@1 = 54.8%
			Quasar-T (https://github.com/bdhingra/quasar)	EM = 43.2% F_1 = 49.7%
			SearchQA	EM = 59.6% F_1 = 65.3%
			TriviaQA	EM = 49.6% F_1 = 54.8%

analyze the rhetorical structures of answer passages at intersentence level with the five rhetorical relations by using discourse markers and connectives. For recognizing the arguments existing in a document which are not recognized using discourse markers, we use an NLI model to check whether the relation of the two text spans is entailment. For question matching, we also use NLI model with the simple rule that a text matches the question if it implies the question. Our work has three main contributions to why-question answering system. First, we define the answer of why-question using the reason relation concept for explicitly listing the cases where we can find the answer for why-question. Second, we propose a discourse-argument hybrid approach in why-question answering problem to find the answer of why-question as our answer definition. In this novel approach, we analyze the discourse structures of texts with rhetorical structure theory (RST) [6] for identifying the reason parts of the five rhetorical relation types, and we also identify the reason parts by constructing simple arguments in which the contents of the why-questions are the conclusions. Third, we propose a Vietnamese why-question answering model with our approach and implement it with the most appropriate techniques. In this model, we propose a question matching method using an NLI model.

This paper will present our work on building a Vietnamese discourse-argument hybrid system for Vietnamese why-question answering. Our system is the first system integrating both textual argumentation and discourse analysis in identifying the arguments and explanations in a text for answer selection. For building our system, we firstly propose the definition of reason relation and the definition of why-question's answer in reading comprehension context as foundations of answer selection. Then, we apply state-of-the-art models in sequential labeling and natural language inference for solving the problems in argument generation and discourse analysis at intersentential level. Finally, we propose our system architecture for answering Vietnamese why-questions in reading comprehension context. Our contributions are to firstly introduce the why-question answering problem in argumentation and discourse perspective, to propose solutions for the two main problems in this approach, and to finally propose the argumentation-discourse hybrid system for Vietnamese why-question answering in reading comprehension context. Our paper is presented in six sections. Section 1 introduces our approach in why-question answering and shows the differences between our approach and existing approaches. Section 2 presents a background on discourse analysis with RST, NLI, and argument generation problems. Section 3 describes our problem, the approach to solving this problem, and our proposed method for why-question answering. Section 4 presents our system model for implementing our why-question answering method. Section 5 describes the datasets and the settings for our system evaluation. Then, some conclusions and future directions are shown in Section 6.

2. Background

2.1. RST-Style Parsing. Rhetorical structure theory (RST) [13] views documents as sets of rhetorical relations between text units called elementary discourse units (EDUs) [22]. These EDUs are independent clauses. They are nonoverlapping text spans and are not possibly divided into smaller units in documents. The EDUs can combine within certain relations to make larger discourse units, arguments, or opinions [23]. Therefore, RST-style parsing is very important to understand texts at document level. We can identify the premises and the conclusions of an argument or the reasons and the claims of an opinion easily if we have an efficient RST-style parser. Delmonte's example of why-question answering has the RST structure as shown in Figure 1: "*Maple syrups come from sugar maple trees. At one time, maple syrup was used to make sugar. This is why the tree is called a 'sugar' maple tree.*" This text fragment presents an argument to explain the name "*sugar maple*." We can easily recognize this argument and identify its premises and the conclusion by exploring its RST structure. This means we can find the answer of why-question in RST structures.

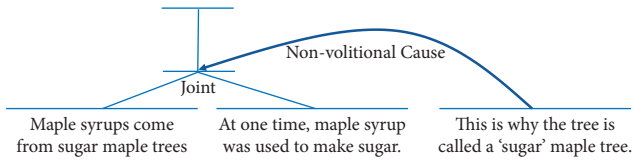


Figure 1: The RST structure of an argument.

RST-style parsing aims at identifying the document's discourse structure according to rhetorical structure theory [13]. There are two approaches in RST-style parsing. Rule-based parsers [22, 24–26] rely on discourse markers, connectives, and lexicon semantics defined in a verb net or an ontology to identify the discourse parse trees. The rule-based parsers have quite low performances with highest reported F_1 scores in EDU segmentation and in document level parse of, respectively, 70.35% and 35.44% [26]. Machine-learning-based parsers [27–32] employ sequential labeling and multiclass classification methods for EDU segmentation and discourse relation identification. The performance of machine-learning-based parsers is higher than that of rule-based ones. The highest F_1 scores of these machine learning parsers are 93.8% [32] in EDU segmentation and 59.9% in document level parse [27]. Although machine leaning parsers have better performance, they have to be trained on a large RST-style discourse treebank which is rare and costly especially in low-resource languages.

2.2. Argumentation by Analogy. Argumentation aims at studying the argument patterns for generating valid arguments or considering the validity of arguments. People use arguments in all activities in which the analogy arguments are very popular [33]. In research of argument from analogy, Walton et al. [5] have introduced many argument schemes from which a person can make valid arguments; however, these argument schemes are quite difficult to implement in computer programs because each argument scheme is independent guidance which is only understood by humans. Juthe [34] proposes an argument scheme which is possibly applied to make valid arguments. Figure 2, referenced in [34], illustrates Juthe's argument scheme.

In Juthe's argument scheme, the Assigned-Predicate∗ (the Target) is an argument whose validity should be considered and the Assigned-Predicate (the Analog) is a valid argument. If every element of the Assigned-Predicate has a corresponding element of the Assigned-Predicate∗, and the Assigned-Predicate and the Assigned-Predicate∗ have the same determining relation, then the Assigned-Predicate∗ is a valid argument. In this scheme, an element and its corresponding one must be analogous [34]. This means they must have the same important properties or roles in the arguments. The determining relation is one of many relations, supervenience, causal, truthmaking, correlation, inferential, etc. [34]. Juthe's argument scheme has an important advantage; that is, if we can compute the similarity of two text spans, we might apply this argument scheme for argument validity computation.

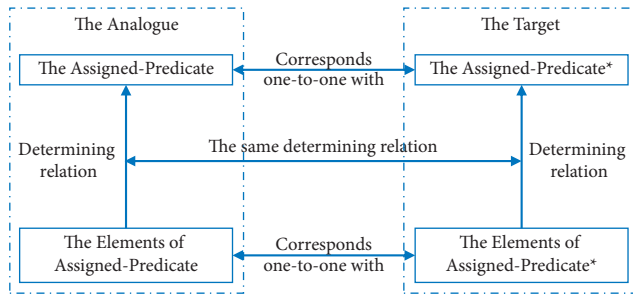


Figure 2: Juthe's argument scheme proposed in [34].

2.3. BERT Architecture. Bidirectional Encoder Representation from Transformers (BERT) [4] is a multilayer neural network architecture in which each layer is an encoder [35]. Figure 3 illustrates BERT architecture. BERT architecture is used to train neural language models with two tasks: masked language modeling and next sentence prediction. These models, called BERT pretrained models, generate an output vector V_{token} for each input token and an output vector V_{CLS} for the whole input text. These vectors are calculated from word embeddings, positional embeddings, and segment embeddings of input tokens all at once at each encoder layer. Word embeddings represent the lexicon semantic in distributional semantics. Positional embeddings and segment embeddings represent the effect of a token's position on other tokens' output vectors, so they are possibly considered as syntactic features. Therefore, BERT pretrained model may compute the output vector of each token with both semantic and syntactic features. Many studies [36–38] have shown that BERT architecture computes the context vector of each input token with syntactic and semantic aspects. BERT pretrained models are used in many natural language processing (NLP) downstream tasks by fine-tuning specific training data. The fine-tuned models have shown their state-of-the-art results in many NLP tasks [4].

In BERT models, the input length M, the number of encoder layers L, the dimension of output vector H, and the number of attention heads A have significant effect on downstream tasks. These parameters will be selected due to the computation capability in training, fine-tuning, and inference. Devlin's BERT models [4] have two settings. $\text{BERT}_{\text{base}}$ has the number of input tokens $M = 512$, the number of encoder layers $L = 12$, the dimension of output vector $H = 768$, and the number of attention heads $A = 12$. $\text{BERT}_{\text{large}}$ has the number of input tokens $M = 512$, the number of encoder layers $L = 24$, the dimension of output vector $H = 1024$, and the number of attention heads $A = 16$. PhoBERT models [39], which are Vietnamese pretrained BERT models, also have two settings as BERT models do; however, PhoBERT models only have number of input tokens $M = 256$, which means we can analyze shorter input text. The performances of these two settings of PhoBERT are slightly different [39]; therefore, we should choose $\text{PhoBERT}_{\text{base}}$ for fine-tuning downstream NLP tasks in Vietnamese.

BERT pretrained models are used to generate feature vector for each input token; therefore, we need a classifier at

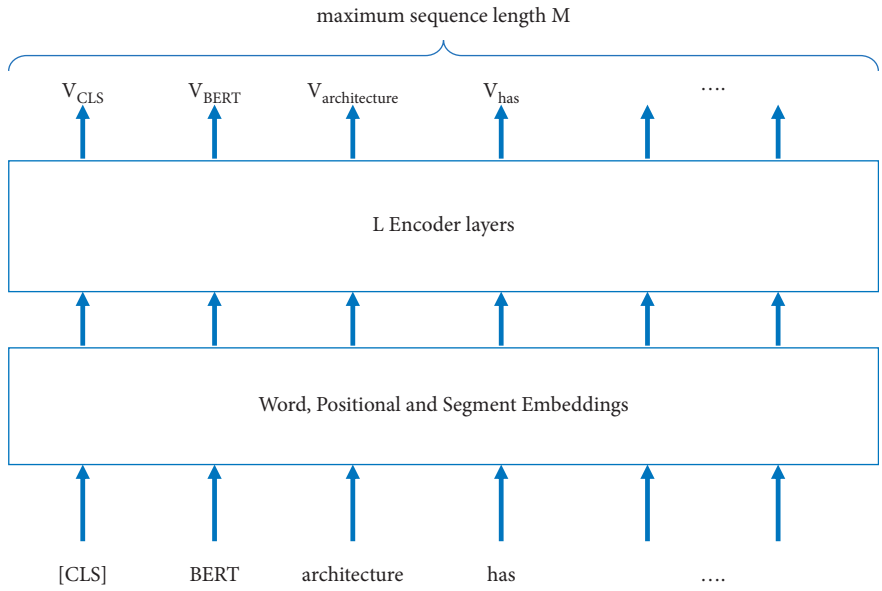


Figure 3: BERT architecture [4].

the end of BERT architecture for each specific task. The output of each token V_{word} or of the whole input V_{CLS} will be the input of the classifier. In fine-tuning step, this classifier will be trained jointly with the BERT model with the number of fine-tuning epochs from 2 to 4 to avoid overfitting [4]. Therefore, building an NLP model by fine-tuning a BERT pretrained model is an efficient approach.

3. Our Approach

Our approach is to define the answer of a given why-question with a text content by characteristics first. Then, we propose a method of finding the answer in the text content and the model of answering why-question in reading comprehension problem with the necessary techniques for implementing a Vietnamese why-question answering system.

3.1. Why-Question Answering with a Single Document. The above why-question answering methods [8, 14, 16–19] have been studied as a task in information retrieval. They find the answers in two phases: passage retrieval and answer ranking. These methods focus on answer ranking which identifies the answer candidates in passages and computes the relevance of these candidates. Recently, many deep models have been proposed for answering questions in SQuAD dataset, where these models have to identify only one answer for a given question and context. The results of these models are shown in SQuAD website (https://rajpurkar.github.io/SQuAD-explorer/). This means the answer candidate extraction has a key role in question answering, and we focus on answer extraction rather than passage retrieval. Therefore, our problem is to find the answer A for a given why-question Q and context D.

Why-questions are raised when people need the reasons. The reasons may be found in arguments or explanations. There is one important difference between an argument and an explanation. According to Johnson and Blair [40], an argument is a claim and the reasons for supporting that claim while an explanation is to provide the information about the origin, cause, meaning, or significance of an event or a phenomenon. When presented in natural language, an argument and an explanation may use similar sentence structures. For example, "*The price of this product is rising because its raw material cost is rising*" is an argument while "*She buys a lot of dresses because it is her preference*" is an explanation. These two sentences are compound sentences linked by the connective "*because*." This characteristic has been utilized in some research on why-question answering. However, if we build a text classifier by training it on an automatic built dataset for recognizing whether a text span is the answer of a why-question, this classifier may not be efficient because the automatic built dataset may contain both explanations and arguments and these two types are different.

In our approach, we will analyze discourse structure of a document for identifying the arguments and explanations, and we compute the entailment relation of a pair of text spans for identifying the arguments containing one premise and one conclusion. The explanations may be extracted from discourse relations of five types named Cause, Result, Purpose, Motivation, and Circumstance [8, 41]. We use both arguments and explanations in the same way when finding the answer for why-question because they are both used to provide the reasons for an event or a phenomenon. We will find the answer by processing these arguments and explanations.

3.2. Definitions. We define the answer A of a why-question $Q =$ "*Why C?*" given a context D for formal answer identification. Our definition about the answer of why-question uses the reason relation concept which is defined as follows.

Definition 1. (reason relation of two text spans).

Given text spans sp_1 and sp_2 in natural language, the reason relation of two text spans sp_1 and sp_2, expressed as $sp_1 > sp_2$, is a binary relation defined as follows:

$$sp_1 \triangleright sp_2 \Leftrightarrow \begin{cases} sp_1 \prec sp_2, \\ \text{Cause}(sp_2, sp_1), \\ \text{Result}(sp_1, sp_2), \\ \text{Purpose}(sp_2, sp_1), \\ \text{Motivation}(sp_2, sp_1), \\ \text{Circumstance}(sp_1, sp_2). \end{cases} \quad (1)$$

Here,

(i) $sp_1 \prec sp_2$ means sp_1 is the premise and sp_2 is the conclusion of an analogy argument

(ii) $\text{Cause}(sp_2, sp_1)$ means sp_1 is the satellite and sp_2 is the nuclei of a Cause relation (Volitional Cause or Nonvolitional Cause) [22]

(iii) $\text{Result}(sp_1, sp_2)$ means sp_2 is the satellite and sp_1 is the nuclei of a Result relation (Volitional Result or Nonvolitional Result) [22]

(iv) $\text{Purpose}(sp_2, sp_1)$ means sp_1 is the satellite and sp_2 is the nuclei of a Purpose relation [22]

(v) $\text{Motivation}(sp_2, sp_1)$ means sp_1 is the satellite and sp_2 is the nuclei of a Motivation relation [22]

(vi) $\text{Circumstance}(sp_2, sp_1)$ means sp_1 is the satellite and sp_2 is the nuclei of a Circumstance relation [22]

The reason relation defined in Definition 1 has two properties as follows:

(i) Reflexivity: given text units sp_1 and sp_2 in natural language, $sp_1 \triangleright sp_2$

(ii) Transitivity: given text units sp_1, sp_2, and sp_3 in natural language, if $sp_1 \triangleright sp_2$ and $sp_2 \triangleright sp_3$, then $sp_1 \triangleright sp_3$

Intuitively, we can examine whether these two properties are true. For the reflexivity, it is obviously true that everything is the reason of itself, although this does not provide any further valuable information. For transitivity, if sp_1 is the reason of sp_2 and sp_2 is the reason of sp_3, then we can say that sp_1 is the deep reason of sp_3 and thus sp_1 is the reason of sp_3 too.

We define the answer of a why-question in Definition 2, which is the foundation for proposing our solution in Vietnamese why-question answering problem. According to this definition, an answer of why-question should be chosen from a discourse structure of a text and the implicit arguments. A discourse structure contains many explanations while arguments in which the content of why-question is the conclusion may not appear in discourse structure. The approaches of Verberne [7–12], Higashinaka and Isozaki [14], and Oh et al. [16–19] try to identify the reason part with a classifier. Because the explanations and arguments are different and the explanations may be explicitly presented in discourse structure while arguments need real world knowledge to be identified, they cannot be identified exactly with one classifier. Therefore, Definition 1 and Definition 2 constitute a novel approach to finding the answer of why-question.

Definition 2. (the answer of a why-question).

Given a document D and a why-question $Q =$ "*Why C?*" in natural language, $A = \{sp_1, sp_2, \ldots, sp_k\}$ is the answer of question Q according to document D if all the following conditions are satisfied:

(i) $sp_i \in D$, sp_i is a nonoverlapping text span in D.

(ii) $sp_i \triangleright C$.

(iii) $\forall i, j \in [1, k]$, $j \neq l$, $sp_i \overline{\triangleright} sp_j$. This means two arbitrary text spans of the answer **A** do not make a reason relation. In order words, **A** does not contain any redundant text span.

3.3. Finding the Answer for Why-Question. We find the answer of a given why-question and a document with Definition 2. In our approach, we split the document into EDUs for improving F_1 score because the EDU is the smallest independent clause. Although some why-questions in SQuAD datasets [42, 43] are possibly answered with noun phrases, the answers as clauses are more formal than these phrases. Our answer A is a set of EDUs $\{sp_1, sp_2, \ldots, sp_k\}$ satisfying Definition 2.

For identifying the reason relations in document D, we will employ a sentence level RST parser to recognize the five discourse relation types described in Definition 1 and an argument generator to generate arguments which contain one premise and one conclusion in document D. Our argument generator needs many presuppositions which are valid arguments for entailment recognition. When training or fine-tuning an NLI model, its parameters will be modified to separate the entailment relation from other relations. This means it can encode the valid arguments and compute the analogy of a pair of text spans and the valid arguments. Therefore, we propose using an NLI model for building argument generator.

From reason relations, we can build a directed reason graph in which the vertices are EDUs and the edges are the reason relations of the document. An edge is in the reverse direction of the corresponding reason relation. We will find the answer of question $Q =$ "*Why C?*" by identifying the most appropriate EDU, named S, for the question Q. This means the relation of S and C is the entailment with the highest score. Then, we find all vertices $\{sp_i\}$ connected to S by breadth-first search. Finally, we select the vertices $\{sp_j\}$ which do not have any path to other vertices. $A = \{sp_i\}$ is the answer of question Q according to Definition 2.

3.4. Vietnamese RST-Style Parsing at Intersentence Level. According to the result of many RST parsers, we will not build a full parser at document level, but we will build a restricted RST parser at intersentence level with five discourse relations, Cause, Result, Purpose, Motivation, and Circumstance. In our RST parsing method, we segment a document into EDUs, and

then we apply a rule-based parser to recognize those five relations at three levels, named inner-EDU level, inner-sentence level, and intersentence level. At intersentence level, we just recognize the relation between two consecutive sentences. The result of our method is many discourse relations which may not connect to others to form a discourse parse tree because we do not recognize the rest of discourse relations.

3.4.1. EDU Segmentation. We fine-tune a $\text{PhoBERT}_{\text{base}}$ [39] pretrained model, called UNISeg, for identifying the boundaries of EDUs. First, we create an EDU boundary annotated dataset by exploiting 9046 parse trees from NIIVTB treebanks [44]. We identify all independent clauses in each parse tree and annotate them with a simple rule; that is, all words at the beginning of an independent clause are labeled with "*BC*," and all remaining words are labeled with "*O*." With this annotation, an EDU begins with a word labeled "*BC*" and ends at the word before a "*BC*" labeled word or at the last word of the sentence. We use the BERT sequential labeling architecture [4] for fine-tuning $\text{PhoBERT}_{\text{base}}$ pretrained model on our EDU segmentation dataset. We use the predicted results of UNISeg model to segment a sentence into EDUs with the span based F_1 score of 0.8. The details of our UNISeg model have been presented in a research article being published.

3.4.2. Intersentence Reason Parser. Our parser recognizes the five discourse relations through inner-EDU, inner-sentence, and intersentence levels and converts them to reason relation according to Definition 1. It identifies the discourse relations at inner-EDU level first; because an EDU is an independent clause, it may include the discourse relations, and if we do not recognize these relations first, they might be wrongly recognized at inner-sentence level. This is also the reason why our method recognizes the discourse relations at inner-sentence level before intersentence level. We build our rule-based parser in 2 phases. The first phase is to identify two context-free grammars (CFG) $G1 = <Dis, N, \Sigma, P1>$ and $G2 = <Dis, N, \Sigma, P2>$ for inner-sentence and intersentence parsing, respectively. The components of *G1* and *G2* are as follows:

(i) *Dis* is a primitive symbol which will generate other symbols.

(ii) $N = \{ReasonNS, ReasonSN, ReasonNN, ReasonTM, P, Word\}$ is a set of nonterminal symbols. *ReasonNS*, *ReasonSN*, *ReasonNN*, and *ReasonTM* mean the reason relation with nuclei in the left, in the right, and in both the left and the right and the reason relation being recognized, respectively. *P* means a text span including several text spans and discourse markers. Word means a discourse marker.

(iii) Σ is a set of terminal symbols. The terminal symbols are <*span*>, several discourse markers with the form <*discourse-marker*>, and <*punc*> for "," character.

(iv) *P1* is a set of production rules for inner-sentence parsing.

(v) *P2* is a set of production rules for intersentence parsing.

The symbol <*span*> in Σ set is the representation of a text span which does not include any "," characters or discourse markers. This means <*span*> does not contain any discourse relations. Our parser recognizes a string of terminal symbols; thus, an EDU must be converted to string of terminal symbols before passing through the parser. The terminal symbol conversion begins with discourse marker recognition. We recognize discourse markers with the corresponding regular expression patterns. We use a list of discourse markers [45] and specify the recognition pattern for each discourse marker. Then, we split the EDU with discourse markers and "," characters. Finally, we replace split texts, discourse markers, and "," characters with <*span*> symbols, corresponding <*discourse-marker*> symbols, and <*punc*> symbols, respectively.

The two sets *P1* and *P2*, which contain context-free production rules, have been built considering text fragments from [45]. These fragments may be sentences or pairs of consecutive sentences. *P1* set contains inner-sentence discourse relation recognition rules which are manually extracted from each sentence. In *P1*'s production rules, the discourse markers may occur at the beginning or in the middle of an EDU or of a sentence. If a discourse relation of the five relations is recognized, we will identify the discourse markers, the nuclei, and the satellite; then, we convert this discourse relation into reason relation according to Definition 1 before adding it to *P1* set. *P2* set contains intersentence discourse relation recognition rules. These rules are extracted from two consecutive sentences using discourse markers. In the five discourse relation types, discourse markers of intersentence relations usually occur at the beginning of the second sentence and rarely occur at the end of the first sentence. We also recognize them and convert them into reason relation according to Definition 1 before adding them to *P2* set. In this building step of grammars *G1* and *G2*, we apply discourse relation patterns which are illustrated in Table 3. Our complete list contains 64 patterns.

For illustration, assume that "*Lý do cho quy tắc số đông là nguy cơ xung đột lợi ích cao và/hoặc tránh quyền lực tuyệt đối*" (in English: "*The reason for the majority rule is the high risk of a conflict of interest and/or the avoidance of absolute powers*") is a sentence for extracting rules. We consider that this sentence explains the reason of "*quy tắc số đông*" (in English: "*majority rule*") and the reason is "*nguy cơ xung đột lợi ích cao và/hoặc tránh quyền lực tuyệt đối*" (in English: "*the high risk of a conflict of interest and/or the avoidance of absolute powers*"); thus, "lý do cho" (in English: "*the reason for*") and "*là*" (in English: "*is*") are discourse markers. Therefore, we note the pattern "*lý do cho N là S*" with its reason relation and add these rules "*ReasonSN* ⟶ <*lydocho*> *P* <*la*> *P*," "*Word* ⟶ <*lydocho*>," and "*Word* ⟶ <*la*>" to *P1*. In these rules, <*lydocho*> and <*la*> stand for discourse markers "*lý do cho*" and "*là*," respectively. *P2* is built in the same way as *P1*.

The second phase is to propose an algorithm for recognizing intersentence level reason relation from the five discourse relation types. Algorithm 1 recognizes the reason relations from each EDU with grammar *G1*, then from each sentence with grammar *G1*, and then from multiple

Table 3: The illustration of discourse relation patterns (*N*: nuclei, *S*: satellite; italics: intersentence relation pattern).

Ord.	Pattern	Pattern meaning	Discourse relation type	Level	Reason relation
1	*S* là nguyên nhân dẫn đến *N*	*S* is the reason of *N*	Cause	Inner-sentence	Reason (S, N)
2	*S. Đây là lý do tại sao N*	***S** . This is why **N***	*Cause*	*Intersentence*	*Reason (S, N)*
3	*N* với mục đích *S*	*N* with the purpose of *S*	Purpose	Inner-sentence	Reason (S, N)
4	Với mục đích *S*, *N*	For *S*, *N*	Purpose	Inner-sentence	Reason (S, N)
5	*N* phát sinh từ *S*	*N* comes from *S*	Result	Inner-sentence	Reason (S, N)
6	Phát sinh từ *S*, *N*	From *S*, *N*	Result	Inner-sentence	Reason (S, N)
7	*N* nguyên nhân là *S*	*N* because *S*	Cause	Inner-sentence	Reason (S, N)
8	Lý do cho *N* là *S*	The reason for *N* is *S*	Cause	Inner-sentence	Reason (S, N)
9	*N* trong khi *S*	*N* while *S*	Circumstance	Inner-sentence	Reason (N, S) Reason (S, N)
10	Trong khi *S*, *N*	While *S*, *N*	Circumstance	Inner-sentence	Reason (N, S) Reason (S, N)
11	*S. Trong khi đó, N*	*S. Meanwhile, N*	*Circumstance*	*Intersentence*	*Reason (N, S)* *Reason (S, N)*
...	...	...	...	...	...

sentences with grammar *G2*. In Algorithm 1, each EDU is converted into string of terminal symbols before parsing, and the parsed results are converted into text spans after parsing. In this algorithm, we use function *SentDetect()* for splitting a text into sentences, function *EDUSegment()* for segmenting a sentence to EDUs, function *ConvertToSymbol()* for converting a natural language text to symbols string and a lookup table of pairs of symbols and text spans, function *Earley()* for getting the parse tree containing the highest number of reason relations among many parse trees from a string of symbols, and function *GetRelation()* for getting reason relation from all parse trees.

For evaluation, we use this parser for recognizing the reason relations from 250 text fragments. The results show that it can recognize 78% of reason relations in these 250 text fragments.

3.5. Argument Generation. Definition 1 shows that the arguments are also reason relations. Therefore, we employ the NLI solution to make arguments. Our approach is to build an NLI model for verifying if a pair of text spans has a text entailment relation. With this NLI model, we can generate arguments by picking two EDUs P and H, in which P is premise and H is hypothesis, and then predict their relation. If the predicted relation is entailment, we have an argument $P \prec H$. According to Juthe's study in argumentation by analogy [34], if P and H are analogous to the premise and conclusion of a certain valid argument, then $P \prec H$ is also an argument. Our NLI model may be considered as a function computing the analogy of P and H with the premises and the conclusions of many valid arguments. These arguments are the entailment samples in training dataset, and the training process also encodes these arguments as the parameters of the NLI model.

We use BERT architecture [4] for building our NLI model because this architecture can compute both syntactic and semantic information of the input text [36–38]. We apply transferred learning approach in building our model. First, we build a Vietnamese NLI dataset, called VSupMNLI, by combining Vietnamese version of MultiNLI dataset [46] with XNLI dataset [47] and our VSupNLI dataset. Our VSupNLI dataset is a Vietnamese native dataset. We combine these two datasets for enriching the Vietnamese version of MultiNLI dataset with Vietnamese native samples from VSupNLI. VSupNLI also provides many samples with which the trained model cannot learn some marks in premises or hypotheses for predicting the relations without computing the semantic similarity of those pairs. Then, we fine-tune $\text{PhoBERT}_{\text{base}}$ pretrained model on our VSupMNLI and build our model vNLI. Our vNLI model has accuracies of 0.7658 and 0.9665 on Vietnamese XNLI test set and on our Vietnamese VSup test set, respectively.

With vNLI model, we can generate arguments from a document with a simple process. The generated arguments have only one premise and only one conclusion because we can encode a premise and a conclusion as an input text for BERT models only. The argument generating process is presented in Algorithm 2. In this algorithm, we use function *isEntailment()* for verifying if $P \prec H$ is valid with an NLI model.

4. Vietnamese Discourse-Argument Hybrid QA System

We propose our novel Vietnamese discourse-argument hybrid QA system based on our novel approach. Our system is the first system applying discourse analysis and argumentation in solving why-question answering problem. As shown in Figure 4, our system has three key components (discourse parser, argument generator, and answer selector) and one simple component (sentence transformer). Given a document D and a question "*Tại sao C*?" (In English: "*Why C*?"), the discourse parser produces a list of EDUs and a list of intersentence reason relations of the document D while the sentence transformer converts the interrogative form to affirmative form of the question "*Tại sao C ?*" Then, the list of EDUs and the list of Rels are passed to the answer selector and the list o EDUs is passed to the argument generator. The argument generator chooses valid arguments in which there are one premise and one conclusion using presuppositions. These arguments are also passed to answer selector. The answer selector builds a reason graph and selects the best

(i) Input: Text, a text being parsed. UNISeg, a Vietnamese EDU segmentation model. Patterns, a list of patterns for recognizing discourse markers and their symbols being used in grammar *G1* and *G2*. G1, *CFG for recognizing reason relations at inner-sentence level.* G2, *CFG for recognizing reason relations at intersentence level.* Output: Spans, a list of text spans which are EDUs or parts of EDUs from the input Text. Rels, a list of reason relations in form (i, j) where i is the text span index which is the reason of the text span index j.
(1) Sents ⟵ SentDetect(Text)
(2) LookupTable ⟵ {}
(3) TextSyms
(4) for sent_id = 1 to |Sents|
(5) EDUs ⟵ EDUSegment(Sents[sent_id])
(6) SentSyms ⟵ []
(7) for edu_id = 1 to |EDUs|:
(8) ConvertToSymbol(EDUs[edu_id], symbols, lookup)
(9) LookupTable.append(lookup)
(10) tree ⟵ Earley(symbols, G1)
(11) SentSyms.append(tree.childNodes())
(12) tree ⟵ Earley(SentSyms, G1)
(13) TextSyms.append(tree.childNodes())
(14) tree ⟵ Earley(TextSyms, G2)
(15) subtrees ⟵ tree.childNodes()
(16) base_index ⟵ 0
(17) Rels ⟵ []
(18) for subt_id = 1 to |subtrees|
(ii) rel ⟵ GetRelation(subtrees[subt_id], base_index)
(19) Rels.append(rel)
(20) base_index + = |subt.leaves()|
(21) Spans ⟵ LookupTable.values()
(22) return Spans, Rels

Algorithm 1: Intersentence reason relation parsing.

Input: EDUs, a list of EDUs from which the arguments are generated. vNLI, a Vietnamese NLI model. Output: Args, a list of arguments presented as (i, j) meaning the i^{th} EDU is the premise and j^{th} EDU is the conclusion.
(1) Args ← []
(2) for $i = 1$ to |EDUs| − 1
(3) for $j = i + 1$ to |EDUs|
(4) if isEntailment(EDUs[i], EDU[j], vNLI)
(5) Args.append((i, j))
(6) if isEntailment(EDUs[j], EDU[i], vNLI)
(7) Args.append((j, i))
(8) return Args

Algorithm 2: Argument generation.

answer in the document D for the question "*Tại sao C*?" The specific processes of those components are described below.

With vNLI model, we can generate arguments from a document with a simple process. The generated arguments have only one premise and only one conclusion because we can encode a premise and a conclusion as an input text for BERT models only. The argument generating process is presented in Algorithm 2. In this algorithm, we use function *isEntailment()* for verifying if $P \prec H$ is valid with an NLI model.

4.1. Discourse Parser. The process of discourse parser is presented in Figure 5. The input of this component is the document D. The sentence detection step splits D into sentences $\{s_i\}$. The EDU labeling step, for each sentence s_i, predicts the EDU label for all words Ann_i in the sentence using an EDU segmentation model. The EDU segmenting step splits each sentence s_i into EDUs $\{EDU_i\}$ using label predicting results. After that, Each EDU_i of a sentence will be parsed for recognizing all reason relations within each EDU, and then the parsed results of each EDU_i of a sentence will be parsed for recognizing all reason relations within the sentence in relation parsing step, which returns a list of EDUs $\{EDU_i\}$ and a list of reason relations $\{Rel_i\}$ of each sentence. Finally, the parsed results of sentences will be parsed at intersentence level for recognizing intersentence reason relation in intersentence reason relation parsing step. The results of this component are a list of EDUs and a list of reason relations of the document D.

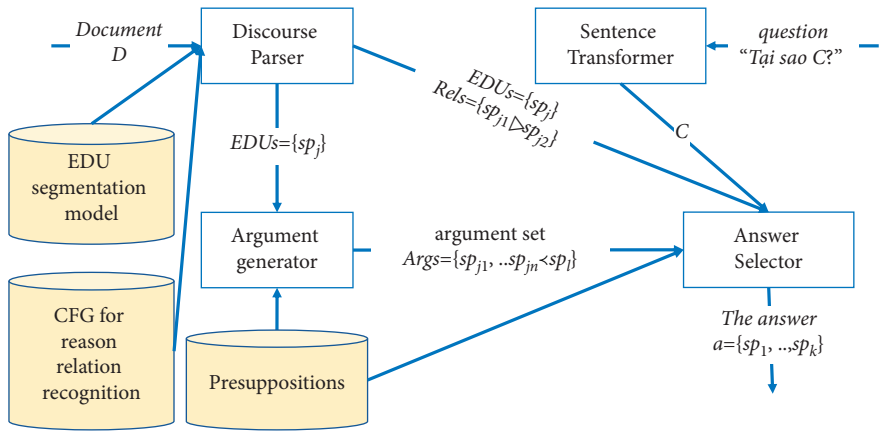


Figure 4: The Vietnamese discourse-argument hybrid QA system model.

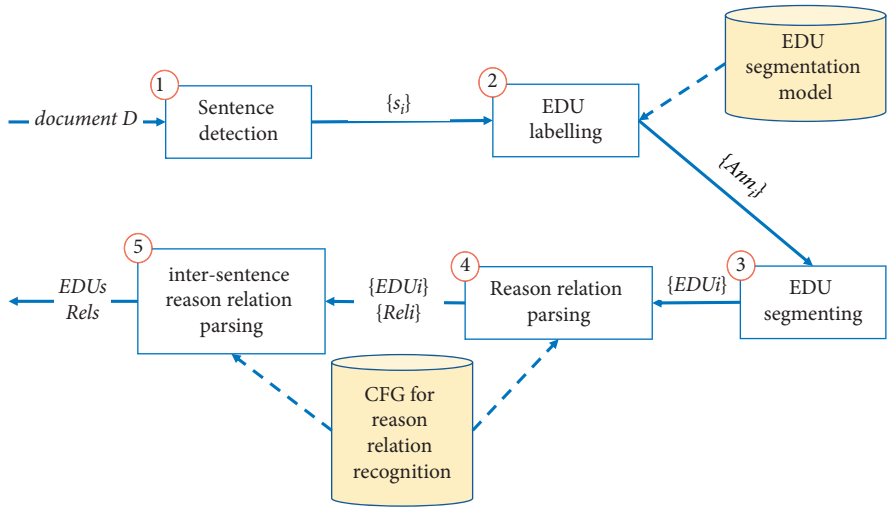


Figure 5: The process of discourse parser component.

4.2. Argument Generator. The process of argument generator, which is the implementation of the Algorithm 2, is presented in Figure 6. The input of this component is a list of EDUs. In the first step, this component picks all pairs of a premise and a conclusion. These pairs may not be arguments; therefore, this component uses presuppositions which are encoded in our vNLI model for computing the arguments' validity in the second step. The result of this component is a list of valid arguments in which there are one premise and one conclusion.

4.3. Answer Selector. The process of answer selector is presented in Figure 7. In the first step, this component builds a reason graph from an EDU list, an Args list, and a Rels list. The graph's vertices are EDUs of the document D, and its directed edges are identified by Args list and Rels list. Each edge has a corresponding argument or relation, where the in-vertex is the premise or the nuclei and the out-vertex is the conclusion or the satellite. In this graph, a tree shows chains of explanations, where the root vertex of the tree is a claim and the leaf vertices of the tree are its reasons according to Definition 2.

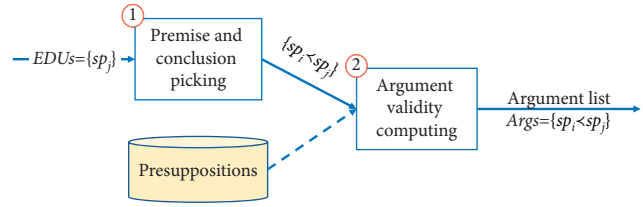


Figure 6: The process of argument generator component.

In the second step, therefore, it selects an EDU, named S, which is the most appropriate to the content C of the question Q. The appropriate measure of an order pair (S, C) is the sum of F_1 score of S over C, number of nodes in tree S, and entailment score of the implication $Sent \longrightarrow C$ using presuppositions, which is implemented as vNLI model. $Sent$ is the sentence containing S. We use entailment score of implication $Sent \longrightarrow C$ because the EDU S may not have enough context information; thus, the entailment score of the implication $S \longrightarrow C$ may be very low although S is the most appropriate to C. The number of nodes in tree S is a heuristic number which is added for choosing the right EDUs because not all EDUs have reason relations in a sentence. A bigger number of reasons

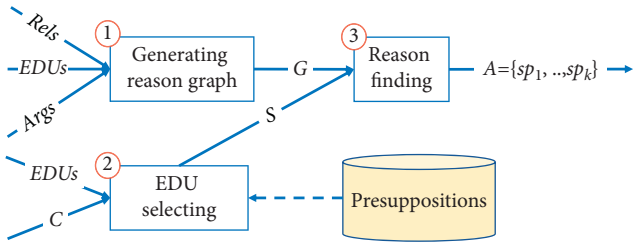


Figure 7: The process of answer selector.

means better explanation. The F_1 score is also added to augment the entailment score. The entailment relation of *Sent* and *C* may have lower score when predicted with vNLI models in practice because vNLI models may not focus on overlapping words which have very different positions in *Sent* and *C*.

In the third step, this component finds the reasons by depth-first search from *S* vertex for identifying the tree with root *S* in the reason graph. Then, all the leaves of *S* tree will be extracted to make the answer *A*. If many EDUs have the same appropriate measure *S* has, this component will identify all the trees and extract all their leaves to make the answer *A*.

5. Evaluation

We evaluate our model by implementing a system and testing it as a black box. We use a Vietnamese why-question dataset in which each sample contains a why-question, a context, and an answer for evaluation. Our system predicts the answer of each sample for calculating the F_1 score. We also compare our results with the results of a sentence retrieval model, of the BERT question answering model, and of a model implemented based on Oh et al. approach [19] to show the advantages and disadvantages of our model.

5.1. Datasets

5.1.1. Training Sets. We use a Vietnamese machine translation version of SQuAD v1.1 training set, called viSQuAD, for fine-tuning PhoBERT-YQA model. This training set contains 74,532 samples because we have removed many samples in which the translated answer does not appear in the translated context.

We build a dataset, called VNCE, by extracting causality sentence from Vietnamese news for training a causality recognition model. We use causality patterns defined in regular expressions with many discourse connectives [45], such as "*vì*" or "*bởi_vì*" (in English: "*because*") and "*để*" (in English: "*for*" or "*in order to*"). We apply these patterns to Vietnamese POS tagged sentences to extract 14,930 sentences. These sentences are automatically tagged with a tag set containing five tags "*B-C*," "*I-C*," "*B-E*," "*I-E*," and "*O*" as described in Oh et al. [18]. We pick 13,437 annotated sentences for training set and 1,493 annotated sentences for test set.

We also build a training set, called VNANS, for training answer selection model. The VNANS is built with causality sentences of VNCE dataset. Each causality sentence is possibly converted to a why-question and answer pair in which the why-question is the effect part and the answer is the causal part; therefore, we use causality sentences to make positive samples. For creating negative samples, we swap the questions and the answers from positive samples in which the overlapping words of two questions are not nouns or verbs. After creating negative samples, VNANS has a training set containing 13,930 positive samples and 97,510 negative samples and a test set containing 1,000 positive samples and 7,000 negative samples. Thus, we duplicate the positive samples in VNANS training set for balance. As a result, VNANS training set has 208,950 samples.

We use VnCoreNLP [48] for Vietnamese word segmentation and POS tagging when building these above datasets.

5.1.2. Test Sets. We use a Vietnamese human translation version of SQuAD v1.1 development set, called VnYQA, for testing. This test set contains 100 samples which contain only why-questions. We use this translated testing set because the samples are selected by many crowd workers; thus, these samples may be diverse. This set is preprocessed with VnCoreNLP [48] for word segmentation. The statistics of our testing set are shown in Table 4. The test samples may be divided into three groups. In the easy group, the answer of a sample is in a sentence of the context which contains almost the words of the why-question. The answers of easy samples may be easy to identify because we can easily select them using their number of overlapping words with the questions. In the moderate group, the answer of a sample is in a sentence of the context which contains some words of the why-question. With the moderate samples, the TF-IDF scores do not ensure the answer sentence selection because some sentences not containing the answers may have higher TF-IDF scores. In the hard group, the answer of a sample is in a sentence of the context which does not contain any word of the why-question or cannot be identified using our vNLI model and its number of overlapping words with the question. To answer the questions of this group, the model must have some type of inference technique because it cannot rely on word matching. The rates of these groups in our test are shown in Table 5.

5.2. Evaluation Settings

5.2.1. VSY-QA Model. We implement sentence retrieval with vector space model, named VSY-QA. For selecting the answer from a context with a why-question ("*Tại sao C*?"), VSY-QA splits the context into sentences and computes the TF-IDF score of each sentence over *C*. Then, it selects the sentence having the highest TF-IDF score.

5.2.2. PhoBERT-YQA Model. We fine-tune a BERT question answering model from $\text{PhoBERT}_{\text{base}}$ pretrained model [39], named PhoBERT-YQA, using neural network architecture proposed by Devlin et al. [4]. We use Hugging Face library for implementing this task. For answer

Table 4: Statistics of test set VnYQA.

Criteria	Size (words)
#context	88
#question/answer	100
#context max. length	899
#context avg. length	198
#question max. length	34
#question avg. length	14
#answer max. length	33
#answer avg. length	10

Table 5: The rates of easy, moderate, and hard groups in VnYQA.

Groups	#samples	Rate (%)
Hard	15	15.0
Moderate	26	26.0
Easy	59	59.0

selection, we select the valid start position and the valid end position where the sum of these positions' scores is the maximum. When predicting the start and end positions with a BERT question answering model, the context is appended after the question to make the input; therefore, the predicted start and end positions may appear in the question span, or the number of tokens between the start and end positions is too big. The valid start and end positions mean these positions are in context span and the number of tokens between them is appropriate. This number is 15 tokens in our setting. We fine-tune PhoBERT-YQA model on viSQuAD with 4 epochs and select the best checkpoint which has F_1 of 71.26% on Vietnamese version of XSQuAD test set [49].

5.2.3. OH-YQA Model. We implement a why-question answering system, named OH-YQA$_{causal}$, following Oh et al. answer selection method [19] because this method has P@1 of 54% while their latest method [16] has P@1 of 54.8% which is slightly higher than the previous one. In OH-YQA system, we replace the CNN model by our BERT fine-tuned model because a BiLSTM with attention model is better than a CNN model in a text classification task as shown in [50] while a BERT fine-tuned model is better than a BiLSTM with attention model as shown in [4]. We build a causality recognition model by fine-tuning a PhoBERT$_{base}$ pretrained model on VNCE training set and an answer selection model by fine-tuning PhoBERT$_{base}$ pretrained model on VNANS training set. We choose causality recognition model and answer selection model as the best checkpoints when fine-tuning is done with 4 epochs. The causality recognition model has tag-based accuracy of 93.58% on VNCE test set, and the answer selection model has F_1 score of 78.16% in selecting correct answer.

We also implement a why-question answering system, named OH-YQA$_{sentence}$. This system has only one difference from OH-YQA$_{causal}$; that is, OH-YQA$_{sentence}$ selects the answer from context's sentences; it does not extract the causal part for answer selection.

5.2.4. DA-YQA Model. We build our system, named DA-YQA, following our model described in Section 4. We use Hugging Face library for implementing vNLI and UNISeg models. The vNLI and UNISeg are fine-tuned from PhoBERT$_{base}$ pretrained model with the appropriate architectures proposed by Devlin [4].

5.2.5. Model Fine-Tuning Costs. We use a NVIDIA Tesla M40 12GB GPU to fine-tune all necessary BERT models for our experiment models. The fine-tuning costs are shown in Table 6.

5.3. Results. We test the experiment systems on VnYQA dataset with NVIDIA Tesla M40 12GB GPU. The execution time and the GPU memory size of these models are shown in Table 7. The results in Table 7 show that our system needs more resources and it consumes more time than other systems because it uses two BERT fine-tuned models for EDU segmentation and natural language inference, and two stages of RST parsing at inner-sentential and intersentential levels. However, its results in Vietnamese why-question answering are promising.

The test results of the experiment systems are shown in Tables 8 and 9. In Table 8, the answer rate column indicates the number of system's answers containing the gold answer. In general, a system can choose an answer containing more information than the gold answer; thus, its F_1 score will be low. Therefore, we use answer rate as an additional criterion for comparison. The results in Table 8 show that our system DA-YQA has a better F_1 score than VS-YQA, OH-YQA$_{causal}$, and OH-YQA$_{sentence}$ systems but it has a lower F_1 score than PhoBERT-YQA system. However, our system has the best answer rate of 77.0%. This means our system may identify the answer more efficiently than systems PhoBERT-YQA, OH-YQA$_{causal}$, and OH-YQA$_{sentence}$ using other deep neural network models.

Table 9 shows the efficiency of our system compared to the four systems VS-YQA, PhoBERT-YQA, OH-YQA$_{causal}$, and OH-YQA$_{sentence}$. We can see these results in Figure 8. Although our system cannot identify all answers in easy samples as VS-YQA system does, it can identify more answers than the four systems in moderate and hard samples. In particular, our system is the best system in identifying the answers in hard samples. These results may indicate that our system has better inference capability than the other four systems. Our system has lower F_1 score than that of PhoBERT-YQA because our system identifies longer answers than PhoBERT-YQA, and many gold answers are noun phrases while our system's answers are usually clauses. This is also the reason why OH-YQA$_{causal}$ has higher F_1 score than that of OH-YQA$_{sentence}$. The OH-YQA$_{causal}$ system has lower answer rate than OH-YQA$_{sentence}$ because there are errors in causality recognition which cause wrong result in answer candidate extraction.

The results of OH-YQA$_{causal}$ and OH-YQA$_{sentence}$ systems are the lowest because the answer selection model is not effective with F_1 score of 78.16% in selecting correct answer. Besides, the method of identifying the causal part in causality

Table 6: Costs for fine-tuning BERT models used in Why-QA models.

Why-QA model	Costs in fine-tuning time (hour)					
	Answer extraction	EDU segmentation	Causality recognizer	Answer selection	Natural language inference	Total
PhoBERT-YQA	7	—	—	—	—	7
OH-YQA	—	—	1	9	—	10
DA-YQA	—	1	—	—	22	23

Table 7: Execution cost of the experiment systems.

	VS-YQA	PhoBERT-YQA	DA-YQA	OH-YQA$_{causal}$	OH-YQA$_{sentence}$
Execution time (seconds per a question)	0.005	0.1	1.93	0.22	0.13
GPU memory size (MB)	—	1.725	2.821	2.273	1.723

Table 8: The why-question answering results of the experiment systems.

System	F_1 (%)	Answer rate (%)
VS-YQA	27.91	68.0
PhoBERT-YQA	52.27	61.0
DA-YQA	46.49	77.0
OH-YQA$_{causal}$	16.95	17.0
OH-YQA$_{sentence}$	23.24	55.0

Table 9: The answer rates of the experiment systems.

Models	Hard		Moderate		Easy	
	#samples	Rates (%)	#samples	Rates (%)	#samples	Rates (%)
VS-YQA	0	0.0	9	34.6	59	100.0
PhoBERT-YQA	0	0.0	17	65.4	44	74.6
DA-YQA	5	33.3	19	73.1	53	89.8
OH-YQA$_{causal}$	1	6.7	4	15.4	12	20.3
OH-YQA$_{sentence}$	1	6.7	11	42.3	43	72.9

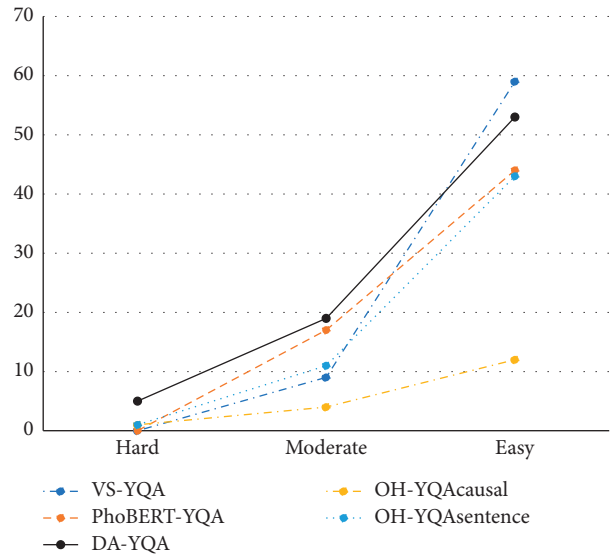


Figure 8: The number of acceptable answers by question groups of VS-YQA, PhoBERT-YQA, DA-YQA, and OH-YQA models.

sentences needs to be improved because it cannot recognize the causal part in a sentence which contains two nested causal relations. For example, the sentence "*This model is effective because it can run in a low resource configuration thus we apply is in our solution*" has the phrase "*This model is effective*" which is a causal part as well as an effect part. Therefore, the sequential labeling may not be a good choice in causal part extraction. In addition, our training data for answer selection problem is not very large. This is also the reason why our implementations of OH-YQA do not have the expected results.

5.4. Discussions. We explore the answers of hard questions from the experiment systems for more details. Table 10 shows all the hard questions answered by one of the experiment systems and their characteristics to explain the way the systems can find the answers.

According to Table 10, DA-YQA system selects four correct answers from discourse relations and one answer from discourse relations with natural language inference. DA-YQA uses vNLI model for question matching; therefore, it can infer the appropriate sentence of a why-question with related words. Then, DA-YQA selects the discourse related EDU group which is the most appropriate to the question; thus, it can select EDUs in reason relations as the answer. However, the vNLI model is effective in our Vietnamese test set, but it is not effective in XNLI test set or in our Vietnamese why-question answering test; therefore, DA-YQA

Table 10: The details of the answers from the experiment systems.

Q-ID	Characteristics	DA-YQA	OH-YQA$_{causal}$	OH-YQA$_{sentence}$
9	(i) Circumstance relation at intersentential level	Yes	No	No
12	(i) Circumstance relation at intersentential level	Yes	No	No
44	(i) Inferring related words (ii) Result relation at intersentential level	Yes	No	Yes
67	(i) Inferring related words	No	Yes	No
81	(i) Circumstance relation at intersentential level	Yes	No	No
99	(i) Cause relation at intersentential level	Yes	No	No

system does not select correct answers in many cases. The OH-YQA systems do not select correct answers in many cases also because the answer selection model is not effective. Another reason is that OH-YQA systems cannot analyze intersentential discourse relations other than inner-sentential causal-effect relations; therefore, it does not select many correct answers.

6. Conclusion and Future Work

In this paper, we would like to present our work on studying a discourse-argument hybrid model for answering a why-question in Vietnamese and implementing a system using this model for evaluation. Our model aims at solving the reading comprehension problem with why-question. For solving this problem, we consider the characteristics of the answers of why-question and then define the answer of the why-question using the concept of reason relation which is also defined in this paper. Our reason relation is a combination of the argument and the five discourse relation types which are used for presenting explanations or arguments. By using reason relations, our model can find 77.0% correct answers while PhoBERT question answering model can find 61.0% correct answers in our test set. This means that our model has better inference capability than PhoBERT question answering fine-tuned model. However, our model has lower F_1 score (46.49%) because it returns EDU-based answers which are usually longer than the gold answers.

At present, our model can recognize the arguments having one premise and one conclusion, and the intersentence level discourse relations of the five types named Cause, Result, Purpose, Circumstance, and Motivation. These limitations come from the computing limitation of PhoBERT pretrained models which can compute the semantic similarity of two sentences and the lack of large Vietnamese RST discourse bank. However, our model still finds 33.3% of answers from hard samples, which indicates that the approach of combining discourse analysis and argument generation in why-question answering is a promising solution.

At present, our argument generating methods and reason relation parsing are limited at intersentence level; thus, our model cannot find the answer for many moderate and hard samples. In future, we will improve these important methods by researching a model which can compute the validity of arguments containing many premises and many conclusions and researching a discourse parsing model which parses full discourse relations at document level. We believe that these two methods will boost our model's performance significantly.

Data Availability

The data used to support the findings of this study have not been made available because they are used in an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 189–198, Vancouver, Canada, 2017.

[2] W. Wang, M. Yan, and C. Wu, "Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1705–1714, Melbourne, Australia, July 2018.

[3] S. Wang and J. Jiang, "Machine comprehension using match-LSTM and answer pointer," in *Proceedings of the International Conference on Learning Representations*, pp. 1–15, Toulon, France, 2017.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[5] D. Walton, C. Reed, and F. Macagno, *Argumentation Schemes*, Cambridge University Press, Cambridge, UK, 2008.

[6] A. Peldszus and M. Stede, "Rhetorical structure and argumentation structure in monologue text," in *Proceedings of the Workshop on Argument Mining*, pp. 103–112, Berlin, Germany, 2016.

[7] S. Verberne, "Developing an approach for why-question answering," in *Proceedings of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 39–46, Trento, Italy, 2006.

[8] S. Verberne, *In Search of Why: Developing a System for Answering Why-Questions*, Radboud University Nijmegen, Nijmegen, Germany, 2009.

[9] S. Verberne, L. Boves, N. Oostdijk, and P. Coppen, "Using syntactic information for improving why-question

answering," in *Proceedings of the Conference on Computational Linguistics*, pp. 953–960, Manchester, UK, 2008.

[10] S. Verberne, L. Boves, N. Oostdijk, and P.-A. Coppen, "What is not in the bag of words for why-QA?" *Computational Linguistics*, vol. 36, no. 2, pp. 229–245, 2010.

[11] S. Verberne, L. Boves, N. Oostdijk, and P. Coppen, "Discourse-based answering of why-questions," *Traitement Automatique des Langues*, vol. 47, no. 3, pp. 21–41, 2007.

[12] S. Verberne, L. Boves, N. Oostdijk, and P.-A. Coppen, "Evaluating Discourse-Based Answer Extraction for Why-Question Answering," in *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 735-736, Amsterdam, Netherlands, 2007.

[13] W. C. Mann and S. A. Thompson, "Rhetorical structure theory: toward a functional theory of text organization," *Text-Interdisciplinary Journal for the Study of Discourse*, vol. 8, no. 3, pp. 243–281, 1988.

[14] R. Higashinaka and H. Isozaki, "Corpus-based question answering for why-questions," in *Proceedings of the International Joint Conference on Natural Language Processing*, pp. 418–425, Hyderabad, India, 2008.

[15] R. Higashinaka and H. Isozaki, "Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions," *ACM Transactions on Asian Language Information Processing*, vol. 7, no. 2, pp. 1–29, 2008.

[16] J.-H. Oh, K. Kadowaki, J. Kloetzer, R. Iida, and K. Torisawa, "Open-domain why-question answering with adversarial learning to encode answer texts," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4227–4237, Florence, Italy, July 2019.

[17] J.-H. Oh, K. Torisawa, C. Hashimoto, R. Iida, M. Tanaka, and J. Kloetzer, "A semi-supervised learning approach to why-question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, February 2016.

[18] J.-H. Oh, K. Torisawa, C. Hashimoto, M. Sano, S. De Saeger, and K. Ohtake, "Why-question answering using intra-and inter-sentential causal relations," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1733–1743, Sofia, Bulgaria, August 2013.

[19] J.-H. Oh, K. Torisawa, C. Kruengkrai, R. Iida, and J. Kloetzer, "Multi-column convolutional neural networks with causality-attention for why-question answering," in *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 415–424, Cambridge, UK., 2017.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 2672–2680, Cambridge, MA, USA, 2014.

[21] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, Vancouver, Canada, 2017.

[22] D. Marcu, *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*, Department of Computer Science, University of Toronto, Toronto, Canada, 1998.

[23] M. Azar, "Argumentative text as rhetorical structure: an application of rhetorical structure theory," *Argumentation*, vol. 13, no. 1, pp. 97–114, 1999.

[24] D. Marcu, "The rhetorical parsing of unrestricted natural language texts," in *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 96–103, Madrid, Spain, July 1997.

[25] L. Polanyi, C. Culy, M. Van Den Berg, G. L. Thione, and D. Ahn, "A rule based approach to discourse parsing," in *Proceedings of the Workshop on Discourse and Dialogue at HLT-NAACL*, pp. 108–117, Boston, MA, USA, April 2004.

[26] R. Subba and B. Di Eugenio, "An effective discourse parser that uses rich linguistic information," in *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 566–574, Boulder, CO, USA, May 2009.

[27] N. Yu, M. Zhang, and G. Fu, "Transition-based neural RST parsing with implicit syntax features," in *Proceedings of the International Conference on Computational Linguistics*, pp. 559–570, Santa Fe, NM, USA, August 2018.

[28] V. W. Feng and G. Hirst, "Text-level discourse parsing with rich linguistic features," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 60–68, Jeju Island, Korea, 2012.

[29] Y. Liu and M. Lapata, "Learning contextually informed representations for linear-time discourse parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1289–1298, Copenhagen, Denmark, 2017.

[30] S. Joty, G. Carenini, and R. T. Ng, "Codra: a novel discriminative framework for rhetorical analysis," *Computational Linguistics*, vol. 3, no. 41, pp. 385–435, 2015.

[31] Q. Li, T. Li, and B. Chang, "Discourse parsing with attention-based hierarchical neural networks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 362–371, Austin, TX, USA, September 2016.

[32] H. Hernault, H. Prendinger, D. A. d. Verle, and M. Ishizuka, "HILDA: a discourse parser using support vector machine classification," *Dialogue & Discourse*, vol. 1, no. 3, pp. 1–33, 2010.

[33] P. Bartha, "Analogy and analogical reasoning," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Online: Metaphysics Research Lab, Stanford University, Stanford, CA, USA, 2019.

[34] L. J. A. Juthe, *Argumentation by Analogy: A Systematic Analytical Study of an Argument Scheme*, Faculty of Humanities, Universiteit van Amsterdam, Amsterdam, Netherkands, 2016.

[35] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, 2017.

[36] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, 2019.

[37] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: what we know about how bert works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.

[38] M. E. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih, "Dissecting contextual word embeddings: architecture and representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Brussels, Belgium, 2018.

[39] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: pre-trained language models for Vietnamese," in *Proceedings of the Conference on Empirical Methods in Natural Language*, pp. 1037–1042, Punta Cana, Dominican Republic, November 2020.

[40] R. H. Johnson and J. A. Blair, *Logical Self-Defense, (United State edition)*, McGraw-Hill, New York, NY, USA, 1994.

[41] R. Delmonte and E. Pianta, "Answering why-questions in closed domains from a discourse model," in *Proceedings of the*

Conference on Semantics in Text Processing, pp. 103–114, Stroudsburg, PA, USA, September 2008.

[42] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: unanswerable questions for SQuAD," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 784–789, Melbourne, Australia, 2018.

[43] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, TX, USA, 2016.

[44] Q. T. Nguyen, Y. Miyao, H. T. T. Le, and N. T. H. Nguyen, "Ensuring annotation consistency and accuracy for Vietnamese treebank," *Language Resources and Evaluation*, vol. 52, no. 1, pp. 269–315, 2018.

[45] C. T. Nguyen and D. T. Nguyen, "Construction of Vietnamese argument annotated dataset for why-question answering method," in *Proceedings of the International Conference on Nature of Computation and Communication*, pp. 124–132, Kien Giang, VietNam, 2016.

[46] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1112–1122, New Orleans, LA, USA, 2018.

[47] A. Conneau, R. Rinott, G. Lample et al., "XNLI: Evaluating cross-lingual sentence representations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2475–2485, Brussels, Belgium, 2018.

[48] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: a Vietnamese natural language processing toolkit," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 56–60, New Orleans, LA, USA, 2018.

[49] M. Artetxe, S. Ruder, and D. Yogatama, "On the cross-lingual transferability of monolingual representations," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4623–4637, Stroudsburg, PA, USA, July 2020.

[50] M. Z. Asghar, F. Subhan, H. Ahmad et al., "Senti-eSystem: a sentiment-based eSystem -using hybridized fuzzy and deep neural network for measuring customer satisfaction," *Software: Practice and Experience*, vol. 51, no. 3, pp. 571–594, 2021.

SN Computer Science (2022) 3:395
https://doi.org/10.1007/s42979-022-01267-x

ORIGINAL RESEARCH

Building a Vietnamese Dataset for Natural Language Inference Models

Chinh Trong Nguyen[1] · **Dang Tuan Nguyen**[2]

Received: 20 April 2022 / Accepted: 22 June 2022

Abstract
Natural language inference models are essential resources for many natural language understanding applications. These models are possibly built by training or fine-tuning using deep neural network architectures for state-of-the-art results. That means high-quality annotated datasets are essential for building state-of-the-art models. Therefore, we propose a method to build a Vietnamese dataset for training Vietnamese inference models which work on native Vietnamese texts. Our approach aims at two issues: removing cue marks and ensuring the writing style of Vietnamese texts. If a dataset contains cue marks, the trained models will identify the relationship between a premise and a hypothesis without semantic computation. For evaluation, we fine-tuned a BERT model, viNLI, on our dataset and compared it to a BERT model, viXNLI, which was fine-tuned on XNLI dataset. The viNLI model has an accuracy of 94.79%, while the viXNLI model has an accuracy of 64.04% when testing on our Vietnamese test set. In addition, we also conducted an answer selection experiment with these two models in which the P@1 of viNLI and of viXNLI are 0.4949 and 0.4044, respectively. That means our method can be used to build a high-quality Vietnamese natural language inference dataset.

Keywords Natural language inference · Textual entailment · NLI dataset · Transfer learning

Introduction

Natural language inference (NLI) research aims at identifying whether a text $\boldsymbol{p}$, called the premise, implies a text $\boldsymbol{h}$, called the hypothesis, in natural language. NLI is an important problem in natural language understanding (NLU). It is possibly applied in question answering [1–3] and summarization systems [4, 5]. NLI was early introduced as RTE [6] (Recognizing Textual Entailment). The early RTE researches were divided into two approaches [6], similarity-based and proof-based. In a similarity-based approach, the premise and the hypothesis are parsed into representation structures, such as syntactic dependency parses, and then the similarity is computed on these representations. In general, the high similarity of the premise-hypothesis pair means there is an entailment relation. However, there are many cases where the similarity of the premise-hypothesis pair is high, but there is no entailment relation. The similarity is possibly defined as a handcraft heuristic function or an edit-distance based measure. In a proof-based approach, the premise and the hypothesis are translated into formal logic then the entailment relation is identified by a proving process. This approach has an obstacle of translating a sentence into formal logic which is a complex problem.

Recently, the NLI problem has been studied on a classification-based approach; thus, deep neural networks effectively solve this problem. The release of BERT architecture [7] showed many impressive results in improving NLP tasks' benchmarks, including NLI. Using BERT architecture will save many efforts in creating lexicon semantic resources,

This article is part of the topical collection "Future Data and Security Engineering 2021" guest edited by Tran Khanh Dang.

Biographical Notes: This paper is a revised and expanded version of our paper entitled "Building a Vietnamese Dataset for Natural Language Inference Models" presented at The 8th International Conference on Future Data and Security Engineering: Big Data, Security and Privacy, Smart City and Industry 4.0 Applications, FDSE 2021, Virtual Event, November 24–26, 2021. Communications in Computer and Information Science 1500, Springer 2021.

✉ Dang Tuan Nguyen
dangnt@sgu.edu.vn

Chinh Trong Nguyen
chinhnt@uit.edu.vn

[1] University of Information Technology, VNU-HCM, Ho Chi Minh City, Vietnam

[2] Saigon University, Ho Chi Minh City, Vietnam

Published online: 25 July 2022

parsing sentences into appropriate representation, and defining similarity measures or proving schemes. The only problem when using BERT architecture is the high-quality training dataset for NLI. Therefore, many RTE or NLI datasets have been released for years. In 2014, SICK [8] was released with 10 k English sentence pairs for RTE evaluation. SNLI [9] has a similar SICK format with 570 k pairs of text span in English. In SNLI dataset, the premises and the hypotheses may be sentences or groups of sentences. The training and testing results of many models on SNLI dataset was higher than on SICK dataset. Similarly, MultiNLI [10] with 433 k English sentence pairs was created by annotating on multi-genre documents to increase the dataset's difficulty. For cross-lingual NLI evaluation, XNLI [11] was created by annotating different English documents from SNLI and MultiNLI.

For building the Vietnamese NLI dataset, we may use a machine translator to translate the above datasets into Vietnamese. Some Vietnamese NLI (RTE) models was created by training or fine-tuning on Vietnamese translated versions of English NLI dataset for experiments. The Vietnamese translated version of RTE-3 was used to evaluate similarity-based RTE in Vietnamese [12]. When evaluating PhoBERT in NLI task [13], the Vietnamese translated version of MultiNLI was used for fine-tuning. Although we can use a machine translator to automatically build Vietnamese NLI dataset, we should build our Vietnamese NLI datasets for two reasons. The first reason is that some existing NLI datasets contain cue marks which was used for entailment relation identification without considering the premises [14]. The second reason is that the translated texts may not ensure the Vietnamese writing style or may return weird sentences.

In this paper, which is the extended version of our paper [15], we propose our method of building a Vietnamese NLI dataset that is annotated from Vietnamese news to ensure writing style and contains more "*contradiction*" samples for removing cue marks. When proposing our method, we would like to reduce the annotation cost by using entailment sentence pairs existing on news webpages. Our contributions are:

(1) To propose Vietnamese NLI dataset creation guidelines based on simple logic rules to ensure that there are no cue marks to determine the relation of a premise-hypothesis pair without semantic computation.
(2) To propose a method to create Vietnamese NLI samples with lower annotation cost by utilizing the title and the introductory sentence of every news from many news websites. In this method, the introductory sentence and the news title are the premise and the hypothesis of a sample, respectively. An annotator is required to check if a premise-hypothesis pair is an entailment sample and provide the contrary sentences from given sentences using our simple guidelines.

Our paper has six sections. The previous section introduces the demand for building the Vietnamese NLI dataset for building Vietnamese NLI models. The following section reviews related works on creating NLI datasets. "The Constructing Method" presents our proposed method of building the Vietnamese NLI dataset. In "Building Vietnamese NLI Dataset", we present the process of building the Vietnamese NLI dataset and some experiments and the subsequent section presents some experiments on our dataset in Vietnamese NLI. Then, some conclusions and our future works are presented in the next section.

Related Works

The early NLI datasets were created for RTE shared tasks. These datasets was manually annotated thus they are good but not large datasets. In 2014, the SICK dataset [8] was released in SemEval 2014. This dataset was created with a three-step process, including sentence normalization, sentence expansion and sentence pair generation. In this process, the sentence expansion step was to automatically create entailment and contradiction sentences by applying syntactic and lexical transformations. In 2015, The SNLI dataset [9] was released to address small datasets' problems and ungrammatical generated sentences. The SNLI dataset was totally annotated by about 2.500 workers [9]. In SNLI creating process, a group of workers had to provide the entailment, contradiction and neutral sentences for every given sentence to ensure the quality of the samples. After that, every five workers had to specify if the relation of a premise-hypothesis pair is entailment, contradiction or neutral. Finally, the relation of each sample was identified as the highest voted relation of the sample. In 2017, MultiNLI dataset was released [10] to provide multi-genre NLI dataset. The MultiNLI dataset was created using the same process of SNLI; however, its data were collected from both written and spoken speech in ten genres.

The Constructing Method

According to the information about SICK, SNLI and MultiNLI datasets, the processes of creation of those datasets required these three steps:

(1) The first step was sentence selection. The conformed sentences are selected as the premises in NLI examples.
(2) The second step was sentence generation. In this step, the contradiction, entailment and neutral sentences of

a given sentence were generated manually or automatically. This step affected the quality of the dataset.

(3) The third step was sample generation. This step had two options to generate samples. In the first option, the workers provided their judgement about given premise-hypothesis pairs for voting the final relations of those pairs. The premise-hypothesis pairs were generated from selected sentences and their entailment, contradiction sentences in the second option.

Our approach to building the Vietnamese NLI dataset is generating samples from existing entailment pairs. These entailment pairs will be crawled from Vietnamese news websites to reduce entailment annotation costs and ensure writing style and multi-genre. We have to annotate contradiction sentences to create our dataset only manually.

NLI Sample Generation

The first requirement of our NLI dataset is that it does not contain cue marks. If a dataset contains these marks, the model trained on this dataset will identify "*contradiction*" and "*entailment*" relations without considering the premises or hypotheses [14]. Therefore, we will generate samples in which the premise and the hypothesis have many common words while their relation varies. We used some logical implication rules for this generation task. For example, given A and B are propositions, we will have the relations of eight premise-hypothesis types, as shown in Table 1.

We used premise-hypothesis types 1 to 4 for removing the cues marks. When training a model, the model will learn from samples of types 1 to 4 the ability to recognize the same sentences and contradiction sentences. We also used types 5 and 6 for training the ability to identify the summarization and paraphrase cases. Type 6 is added in the attempt to remove special marks, which can occur when creating type 5 samples. We also added types 7 and 8 for recognizing the contradiction in paraphrase and summarization cases in which proposition B is the paraphrase or the summary of proposition A, respectively. Types 7 and 8 are valid only if B is the paraphrase or A's summary.

In general, the types 7 and 8 cannot be applied in cases where proposition A implies proposition B by using pre-suppositions. For example, assuming A is the proposition "*we are hungry*", B is the proposition "*we will have lunch*" and A⇒B is the valid proposition "*if we are hungry then we will have lunch*" because we have two pre-suppositions that we should eat when we are hungry and we eat when we have lunch. We see that ¬B, which is the proposition "*we will not have lunch*", is not a contradiction of proposition A.

Table 1 The relations of premise-hypothesis types used for building supplement dataset

Type	Condition	P	H	Relation
1		A	A	Entailment
2		¬A	¬A	Entailment
3		A	¬A	Contradiction
4		¬A	A	Contradiction
5	A⇒B	A	B	Entailment
6	A⇒B	¬B	¬A	Entailment
7	A⇒B	A	¬B	Contradiction*
8	A⇒B	¬A	B	Contradiction*

Entailment Pair Collection

Entailment pairs exist in text documents, but it is difficult to extract them from the text documents. Therefore, after considering many news posts on Vietnamese news websites such as VnExpress, we found that the title usually paraphrases or summarizes the introductory sentence in a news post. Therefore, we can divide these news posts into four types. In type 1, the title is the paraphrase of the introductory sentence in the news post. In the example shown in Fig. 1, the title "*Nhiều tài xế dừng xe đậy nắp cống suốt 10 ngày*" (in English: "*many drivers was stopping to close the drain cover in 10 days*") is a paraphrase of the introductory sentence "*Nhiều tài xế dừng ôtô giữa ngã tư để đậy lại miệng cống hở do chiếc nắp cong vênh và câu chuyện diễn ra suốt 10 ngày ở Volgograd*" (in English: "*Many drivers was stopping the cars at the crossroad to close the slightly opened drain cover because the drain cover was bent*").

In type 2, the title summarizes the introductory sentence in the news post. In the example shown in Fig. 2, the title "*Gạo chữa nhiều bệnh*" (in English: "*rice used for curing many diseases*") is the summary of the introductory sentence

Fig. 1 An example of type-1 news post from vnexpress.net website

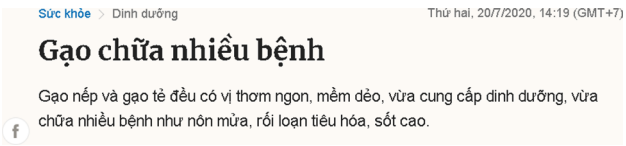


Fig. 2 An example of type-2 news post from vnexpress.net website

Fig. 3 An example of type-3 news post from vnexpress.net website

Fig. 4 An example of type-4 news post from vnexpress.net website

"*Gạo nếp và gạo tẻ đều có vị thơm ngon, mềm dẻo, vừa cung cấp dinh dưỡng, vừa chữa nhiều bệnh như nôn mửa, rối loạn tiêu hóa, sốt cao*" (in English: "*Glutinous rice and plain rice, which are delicious and soft when cooked, provide nutrition and are used for curing many diseases such as vomiting, digestive disorders, high fever*").

In type 3, the title is possibly inferred from the introductory sentence in the news post. Some pre-suppositions are perhaps used in this inference. In the example shown in Fig. 3, the title "*Xuất khẩu rau quả tăng mạnh*" (in English: "*Vegetable export increases significantly*") can be inferred from the introductory sentence "*Bốn tháng đầu năm nay, giá trị xuất khẩu rau quả đạt 1,35 tỷ USD, tăng 9,5% so với cùng kỳ năm ngoái.* " (In English: "*in the first four months this year, vegetable export reaches 1.35 billion USD, increases 9.5% in comparison with the same period in last year*"). In this inference, we have used a pre-supposition which defines that increasing 9.5% means significantly growing exports.

In type 4, the title is a question which cannot have an entailment relation to the introductory sentence in the news post. In the example shown in Fig. 4, the title, which is a question "*Vì sao giá dầu lao dốc chỉ trong 6 tuần?* " (In English: "*why does the oil price dramatically decreases in 6 weeks only*"), cannot have an entailment relation with the introductory sentence "*Chỉ mới cách đây hơn một tháng, giới buôn dầu còn lo ngại thiếu cung có thể đẩy dầu thô lên 100 USD một thùng.* " (In English: "*just more than one month ago, oil traders still worried that the insufficient supply could increase the oil price by 100 USD per barrel*").

We collected only title-introductory sentence pairs of type 1 and type 2 to make entailment pair collection because the pairs of type 3 and 4 cannot be applied 8 relation types when generating NLI samples. The type of a sentence pair is identified manually for high quality. In every pair in our collection, its title is the hypothesis, and its introductory sentence is the premise.

Building Vietnamese NLI Dataset

We built our NLI dataset with a three-step process. In the first step, we extracted title-introductory pairs from Vietnamese news websites. In the second step, we manually selected the entailment pair and made the contradiction sentences from titles and introductory sentences. Finally, in the third step, we automatically generate NLI samples from entailment pairs and their contradiction sentences by applying eight relation types shown in Table 1. In Table 1, the relations of type 1 and type 2 are apparent thus, we created a different version of our dataset in which there have no samples of type 1 and type 2 to show if these samples are meaningful.

Contradiction Creation Guidelines

We made the contraction of a sentence manually for a high-quality result. In our approach, the contradiction sentences are generated in two ways. The first way is to transform them from affirmative structure to negative structure and vice versa. The second way is to use antonyms. We proposed three types of making the contradiction in which type 1 and type 2 are to use structure transformations, and type 3 is to use antonyms. These are simple ways to make the contradiction of a sentence using syntactic transformation and lexicon semantic.

In type 1, a given sentence will be transformed from affirmative to negative or vice versa by adding or removing the negative adverb. If the given sentence is affirmative, we will add a negative adverb to modifier the sentence's main verb. If the given sentence is negative, we will remove the negative adverb, which is modifying the sentence's main verb. The negative adverbs used in our work are "*không*", "*chưa*", and "*chẳng*" (in English: they mean "*not*" or "*not…yet*"). We used one of these adverbs according to the

sentence to ensure the Vietnamese writing style. We have four cases of making contradictions with this type.

Case 1 of type 1, making contradiction from an affirmative sentence containing one verb. We will add one negative adverb to modify the verb. For example, making the contradiction of the sentence "*Đài Loan bầu lãnh đạo*" (in English: "*Taiwan voted for a Leader*"), we will add the negative adverb "*không*" ("*not*") to modify the main verb "*bầu*"("*voted*") for making the contradiction "*Đài Loan không bầu lãnh đạo*" (in English: "*Taiwan did not vote for a Leader*").

Case 2 of type 1, making contradiction from an affirmative sentence containing the main verb and other verbs. We will add one negative adverb to modify the main verb only. For example, making the contradiction of the sentence "*Báo Mỹ đánh giá Việt Nam chống Covid-19 tốt nhất thế giới*" (in English: "*US news reported that Vietnam was the World's best nation in Covid-19 prevention*"), we will only add negative adverb "*không*" to modify the main verb "*đánh giá*" ("*reported*") for making the contradiction "*Báo Mỹ không đánh giá Việt Nam chống Covid-19 tốt nhất thế giới* " (in English: " *US news did not report that Vietnam was the World's best nation in Covid-19 prevention*").

Case 3 of type 1, making contradiction from an affirmative sentence containing two or more main verbs. We will add negative adverbs to modify all main verbs. For example, making the contradiction of the sentence "*Bão Irma mang theo mưa lớn và gió mạnh đổ bộ Cuba cuối tuần trước, biến thủ đô Havana như một 'bể bơi khổng lồ'*" (in English:"*Storm Irma brought heavy rain and winds to Cuba last week, making the Capital Havana a 'giant swimming pool'*"), we will add two negative adverbs "*không*" to modify two main verbs "*mang*" and "*biến*" for making the contradiction "*Bão Irma không mang theo mưa lớn và gió mạnh đổ bộ Cuba cuối tuần trước, không biến thủ đô Havana như một "bể bơi khổng lồ*" (in English: " *Storm Irma did not bring heavy rain and winds to Cuba last week, not making the Capital Havana a 'giant swimming pool'*").

Case 4 of type 1, making contradiction from a negative sentence containing negative adverbs. We will remove all negative adverbs in the sentence. In our data, we did not see any sentence of this case; however, we put this case in our guidelines for further use.

In the type 2, a given sentence or phrase will be transformed using the structure "*không có …*" (in English: "*there is/are no*") or "*không … nào …*" (in English: "*no …*"). We have two cases of making contradiction with this type.

Case 1 of type 2, making contradiction from an affirmative sentence by using structure *"không có …*". We use this case when the given sentence has a quantity adjective or a cardinal number modifying the subject of the sentence and it is non-native if we add a negative adverb to modifying the main verb of the sentence. The quantity adjective or cardinal number will be replaced by the phrase "*không có*". For example, making the contradiction of the sentence "*120 người Việt nhiễm nCoV ở châu Phi sắp về nước*" (in English: "*120 Vietnamese nCoV-infested people in Africa are going to return home*"), we will replace "120" by "*không có*" because if we add negative adverb "*không*" to modify the main verb "về" ("*return*"), the sentence "*120 người Việt nhiễm nCoV ở châu Phi sắp không về nước*" (in English: "*120 Vietnamese nCoV-infested people in Africa are not going to return home*") sounds non-native. Therefore, the contradiction should be "*không có người Việt nhiễm nCoV ở châu Phi sắp về nước*" (in English: "*no Vietnamese nCoV-infested people in Africa is going to return home*"). Case 1 of type 2 will be used when we are given a phrase instead of a sentence. For example, making the contradiction of the phrase "*trường đào tạo quản gia cho giới siêu giàu Trung Quốc*" (in English: "*the butler training school for Chinese super-rich class*"), we will add the phrase "*không có*" at the beginning of the phrase to make the contradiction "*không có trường đào tạo quản gia cho giới siêu giàu Trung Quốc*" (in English: "*there is no butler training school for Chinese super-rich class*").

Case 2 of type 2, making contradiction from an affirmative sentence by using the structure *"không …nào …*". We will use this structure when we have case 1 of type 2 but the generated result of that case is not native. For example, making the contradiction of the sentence "*gần ba triệu ngôi nhà tại Mỹ mất điện vì bão Irma*" (in English: "*nearly three million houses in U.S. were without power because of Irma storm*"), if we replace "*gần ba triệu*" (in English: "*nearly three million*") by "*không có*", we will have a non-native sentence "*không có ngôi nhà tại Mỹ mất điện vì bão Irma*" therefore we should use the structure "*không … nào …*" to

make the contradiction "*không ngôi nhà nào tại Mỹ mất điện vì bão Irma*" (in English: "*There are no houses in U.S. were without power because of Irma storm*").

In type 3, a contradiction sentence is generated using lexicon semantics. A word of the given sentence will be replaced by its antonym. This way will make the contradiction of the given sentence. Although we can use all cases of type 1 and type 2 to make the contradiction, we still recommend this type because the samples generated with this type may help the fine-tuned models learn more about antonymy. We have two cases of making contradiction with this type.

Case 1 of type 3, making contradiction from a sentence by replacing the main verb of the sentence with its antonym. For example, making the contradiction of the sentence "*Mỹ thêm gần 18.000 ca nCoV một ngày*" (in English: "*the number of nCoV cases in U.S. increases about 18,000 in one day*"), we can replace the main verb "*thêm*" ("*increase*") by its antonym "*giảm*" ("*decrease*") to make the contradiction "*Mỹ giảm gần 18.000 ca nCoV một ngày*" (in English: "*the number of nCoV cases in U.S. decreases about 18,000 in one day*").

Case 2 of type 3, making contradiction from a given sentence by replacing an adverb or a phrase modifying the main verb by the antonym or the contradiction of that adverb or that phrase, respectively. We use this case when we need to make the samples containing antonyms, but the main verb does not have any antonyms because many verbs do not have their antonym. For example, making the contradiction of the sentence "*Mỹ viện trợ nhỏ giọt chống Covid-19*" (in English: "*the U.S. aided a little in Covid-19 prevention*"), we cannot replace the main verb "*viện trợ*" ("*aid*") with its antonym because it does not have an antonym. Therefore, we will replace "*nhỏ giọt*" ("*a little*") by "*ào ạt*" ("*a lot*") to make the contradiction "*Mỹ viện trợ ào ạt chống Covid-19*" (in English: "*the U.S. aided a lot in Covid-19 prevention*").

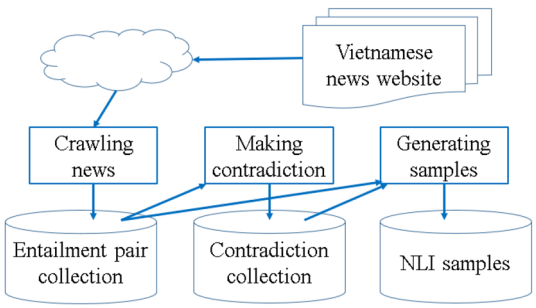


Fig. 5 Our three-step process of building Vietnamese NLI dataset

In this example, "*nhỏ giọt*" and "*ào ạt*" have the opposite meanings; and the phrases "*nhỏ giọt*" and "*ào ạt*" have the adverb role in the sentence when modifying the main verb "*viện trợ*".

Building Steps

We built our Vietnamese NLI dataset follow the three-step process which is a semi-automatic process shown in Fig. 5.

In the first step—crawling news, we used a crawler to fetch unique webpages from sections of international news, business, life, science, and education in the website *vnexpress.net*. Then we extracted their titles and introductory sentences by a website-specific pattern defined with regular expression. The results are sentence pairs stored in an entailment pair collection with unique numbers. These pairs are not always the types 1 or 2; therefore, the entailment pairs will be manually selected right before making contradiction sentences.

In the second step—making contradiction, we firstly manually identified if each pair of the collection was type 1 or 2 for entailment pair selection. When an entailment pair was selected, we made the contradiction sentences for the title and the introductory sentence using the contradiction creation guidelines. The introductory sentences are the premises in the entailment pairs, and the titles are the hypotheses. As a result, we have a collection of pairs of sentences ¬A and ¬B stored in a contradiction collection in which each sentence pair ¬A and ¬B has a condition A⇒B. In this step, we have two people making contradiction sentences. These people are society science bachelors. Because the guidelines for making contradiction sentence are simple, there are no disagreements in the annotation results.

In the third step—generating samples, we used a computer program implemented from our Algorithm 1 for combining the premises, hypotheses stored in entailment pair collection and their contradiction sentences stored in contradiction collection by their unique numbers. The combination rules follow Table 1 in generating NLI samples. The computer program generates "neutral" samples to combine sentences from different premise-hypothesis pairs. In Algorithm 1, the function *getContradict()* return the contradiction sentence stored in contradiction collection. The three functions *ent()*, *neu()*, and *con()* is used for creating entailment, neutral and contradiction samples from a premise and a hypothesis, respectively. For data balancing, we added some duplicated entailment samples in Algorithm 1.

Algorithm 1. Generating NLI samples.

```
Input: E, a list of premise-hypothesis pairs.
Output: SD, the NLI sample list with SNLI format.

SD←[]
PL←[]  //premise list
HL←[]  //hypothesis list
cPL←[] //premise contradiction list
cHL←[] //hypothesis contradiction list
for i←1 to |E|
   prem ← E[i].premise
   hyp ← E[i].hypothesis
   nprem ← genContradict(prem)
   nhyp ← genContradict(hyp)
   if nprem = NULL and nhyp = NULL then
      continue
   end if
   PL←PL+[prem]
   HL←HL+[hyp]
   cPL←cPL+[nprem]
   cHL←cHL+[nhyp]
end for
PL←PL+[PL[1]], HL←HL+[HL[1]]
cPL←cPL+[cPL[1]], cHL←cHL∪[cHL[1]]
for i←2 to len(PL)
   SD←SD+[ent(PL[i],HL[i]), neu(PL[i],PL[i-1])]
   SD←SD+[ent(PL[i],HL[i]), neu(HL[i],HL[i-1])]
   SD←SD+[ent(PL[i],PL[i]), ent(HL[i],HL[i])]
   SD←SD+[neu(HL[i],PL[i-1]), neu(PL[i],HL[i-1])]
   if cPL[i]!=NULL and cHL[i]!=NULL then
      SD←SD+[ent(cHL[i],cPL[i]), neu(cHL[i],HL[i-1])]
      SD←SD+[ent(cHL[i],cPL[i]), neu(cPL[i],PL[i-1])]
   end if
   if cPL[i]!=NULL then
      SD←SD+[con(PL[i],cPL[i]), con(cPL[i],PL[i])]
      SD←SD+[con(PL[i],cPL[i]), con(cPL[i],PL[i])]
      SD←SD+[ent(cPL[i],cPL[i]), neu(PL[i-1],cPL[i])]
   end if
   if cHL[i]!=NULL then
      SD←SD+[con(HL[i],cHL[i]), con(cHL[i],HL[i])]
      SD←SD+[con(HL[i],cHL[i]), con(cHL[i],HL[i])]
      SD←SD+[ent(cHL[i],cHL[i]), neu(HL[i-1],cHL[i])]
   end if
return SD
```

Given a list of entailment samples *E*, Algorithm 1 firstly select from *E* a list of entailment samples in which the premise and the hypothesis of the *i*th sample are *PL[i]* and *HL[i]*. The *i*th sample is only selected if its premise *PL[i]* or hypothesis *HL[i]* has the contradiction premise *cPL[i]* or *cHL[i]*, respectively. Then, entailment and contradiction pairs are generated using the rules in Table 1. For example, a type 1 sample is generated as *ent(PL[i],PL[i])*, a type 3 sample is generated as *con(PL[i], cPL[i])* if the premise *PL[i]* has its contradiction *cPL[i]*, a type 5 sample is generated as *ent(PL[i], HL[i])*. The neutral samples are generated by pairing the premise, hypothesis, premise contradiction or hypothesis contradiction of the *i*th sample and the premise, hypothesis, premise contradiction or hypothesis contradiction of the *i*–1th sample as in building SICK dataset [8].

To show the necessity of the type 1 and type 2 relation in Table 1, we also used a different version of our Algorithm 1 to generate samples. In this version, which is presented in Algorithm 2, the samples of type 1 and type 2 are not generated when creating the dataset.

Algorithm 2. Generating NLI samples without type 1 and type 2.

```
Input: E, a list of premise-hypothesis pairs.
Output: SD, the NLI sample list with SNLI format.

SD←[]
PL←[]  //premise list
HL←[]  //hypothesis list
cPL←[] //premise contradiction list
cHL←[] //hypothesis contradiction list
for i←1 to |E|
   prem ← E[i].premise
   hyp ← E[i].hypothesis
   nprem ← genContradict(prem)
   nhyp ← genContradict(hyp)
   if nprem = NULL and nhyp = NULL then
      continue
   end if
   PL←PL+[prem]
   HL←HL+[hyp]
   cPL←cPL+[nprem]
   cHL←cHL+[nhyp]
end for
PL←PL+[PL[1]], HL←HL+[HL[1]]
cPL←cPL+[cPL[1]], cHL←cHL∪[cHL[1]]
for i←2 to len(PL)
   SD←SD+[ent(PL[i],HL[i]), neu(PL[i],PL[i-1])]
   SD←SD+[ent(PL[i],HL[i]), neu(HL[i],HL[i-1])]
   if cPL[i]!=NULL and cHL[i]!=NULL then
      SD←SD+[ent(cHL[i],cPL[i]), neu(cHL[i],HL[i-1])]
      SD←SD+[ent(cHL[i],cPL[i]), neu(cPL[i],PL[i-1])]
   end if
   if cPL[i]!=NULL then
      SD←SD+[con(PL[i],cPL[i]), con(cPL[i],PL[i])]
   if cHL[i]!=NULL then
      SD←SD+[con(HL[i],cHL[i]), con(cHL[i],HL[i])]
return SD
```

Building Results

In our updated NLI dataset, VnNewsNLI, the rates of making contradiction sentences by applying type 1, type 2 and type 3 are 60.16%, 19.01% and 20.83%, respectively. We also created the VnNewsNLI$_R$, the types 1 and 2 sample removal version of VnNewsNLI using Algorithm 2. The rates of entailment, neutral and contradiction samples in our VnNewsNLI dataset are shown in Table 2. In Table 2, the rates of NLI relation categories are approximately 33.3%.

The statistics of the VnNewsNLI dataset by syllable are shown in Table 3. Table 3 and the distribution of the sentence length (in a syllable) on entailment, neutral and contradiction are shown in Table 4. We used syllables as text length units in Tables 3 and 4 because many multi-lingual pretrained models were trained on unsegmented Vietnamese text datasets. According to Tables 3 and 4, the premises and hypotheses are often short (≤ 14 syllables) and quite long (≥ 20 syllables) sentences; therefore, this dataset may provide the characteristic of short and long sentences. There is a difference between the VnNewsNLI dataset and the SNLI dataset in that the premises and hypotheses are almost sentences in the VnNewsNLI dataset. At the same time, they are groups of sentences in many cases in the SNLI dataset.

We also calculated the frequency distribution of words in our both development set and test set to view the most discussing topics of the samples briefly. The 40 highest frequency words, common nouns and verbs, are presented in Table 5. The frequency distribution of words shows that the politics, military and life topics are most discussed in VnNewsNLI samples.

Experiments

We did some experiments on our VnNewsNLI dataset and on the Vietnamese XNLI dataset [11] and then compared their results to determine if our dataset is useful when building a Vietnamese NLI model. XNLI dataset was manually annotated from English texts then the annotated results were translated into different languages using machine translators. Therefore, Vietnamese XNLI dataset is a Vietnamese translation of XNLI dataset. We also conducted an experiment to show the application of our dataset in answer selection. In this experiment, we used the Vietnamese NLI model for selecting the sentence containing the answer in machine reading comprehension tests. We selected the sentence with highest entailment score as the retrieval result and evaluating with the precision at top 1 (P@1) score. We used UIT-ViQuAD 2.0 dataset [16], which was the expansion of UIT-ViQuAD 1.0 [17], after removing no-answer samples for our evaluation. In our experiments, we used BERT architecture for training Vietnamese NLI models as shown in Fig. 6.

According to the BERT architecture in Fig. 6, a premise and a hypothesis of a sample will be concatenated into an input. This input has the following order: the "*[CLS]*" token, then all premise's tokens, then the "*[SEP]*" token, then all hypothesis' tokens, and the "*[SEP]*" token at the end. Each input token will be converted to a tuple of word embedding, segment embedding and position embedding. These embeddings will go through BERT architecture to generate a context vector for each input token and a context vector for the whole input. The context vector of the whole input is returned at the "*[CLS]*" position. This vector will be used for identifying the relation between the premise and the hypothesis by a classifier. This classifier is a feed forward neural network fully connected to the context vector of the input. It will be trained in fine-tuning steps. We chose BERT architecture for experiment because it can compute the context vector with syntactic and semantic features of the input [18–20].

Experiment Settings

We built three Vietnamese NLI models using BERT architecture as shown in Fig. 6. The first model, viXNLI, was fine-tuned from PhoBERT pretrained-model [13] on Vietnamese version of XNLI development set with word segmentation. The second model, viNLI, was fine-tuned from PhoBERT pretrain-model on our VnNewsNLI development set with Vietnamese word segmentation. The third model, viNLI$_R$, was fine-tuned from PhoBERT pretrained-model on our VnNewsNLI$_R$ development set with Vietnamese word segmentation. We compared viNLI to viNLI$_R$ for showing the effect of type 1 and type 2 samples in NLI datasets. We used Huggingface python library[21] for implementing the BERT architecture and fairseq python library[22] for tokenizing Vietnamese words into sub-words. We also used VnCoreNLP [23] for Vietnamese word segmentation before tokenization.

We fine-tuned these models in 2–8 epochs with learning rate of 3.10^{-5}, batch size of 16 and input maximum length of 200 because the PhoBERT$_{base}$ pretrained model has the limit input length of 258 tokens. In addition, the lengths of the premises and hypotheses are rarely greater than 100 syllables in our datasets. Other parameters were left with default settings. We chose the best models from checkpoints for testing.

Experiment Results

The results of the three models viXNLI, viNLI and viNLI$_R$ on XNLI and VnNewsNLI test sets are shown in Table 6. We conducted this experiment to show the necessary of a Vietnamese native NLI training set for building Vietnamese NLI models. The results show that our Vietnamese native NLI

training set, VnNewsNLI, has improved the performance of our Vietnamese NLI model on Vietnamese native test set with the highest accuracy of 94.79% but it has not with the accuracy of 41.47% on Vietnamese translation of XNLI test set. Meanwhile, the Vietnamese translation of XNLI development set shows its role when viXNLI model has the accuracy of 68.64% but it does not when viXNLI model has the accuracy of 64.04% on VnNewsNLI test set. The reason of these results is that Vietnamese translation of XNLI did not preserve the writing style of Vietnamese texts and the premises and the hypotheses may be a group of sentences. In addition, this experiment also shows that the type 1 and type 2 samples have their important roles in building NLI models for recognizing the equivalent sentences through the accuracy of viNLI model (41.47% and 94.79%) in comparison to viNLI$_R$ model (37.62% and 74.54%) on the two test sets.

We evaluated the three models on a test set consisting of type 1 and type 2 samples of VnNewsNLI test set for more evident results. The results are shown in Table 7. The results of the viNLI model (accuracy of 95.67%) confirm that type 1 and type 2 samples are necessary in NLI datasets

Table 2 The statistics of NLI samples in VnNewsNLI and VnNewsNLI$_R$ dataset

Dataset	Samples	Entailment		Neutral		Contradiction	
	#*n*	#*n*	Rate	#*n*	Rate	#*n*	Rate
VnNewsNLI-dev	20,246	6756	33.37%	6754	33.36%	6736	33.27%
VnNewsNLI-test	11,878	3964	33.37%	3962	33.36%	3952	33.27%
VnNewsNLI$_R$-dev	10,115	3374	33.35%	3373	33.35%	3368	33.30%

Table 3 The statistics of NLI samples by syllable in VnNewsNLI dataset (ent. – entailment, neu. – neutral, con. – contradiction)

Length in syllable	Development set			Test set		
	#ent	#neu	#con	#ent	#neu	#con
Premises, ≤ 8	1578	1808	1684	909	1079	994
Premises, 9–14	1786	1568	1672	1036	889	958
Premises, 15–20	601	598	572	299	285	260
Premises, 20–26	2232	2223	2216	1286	1276	1266
Premises, > 26	559	557	592	432	431	470
Hypotheses, ≤ 8	1814	1807	1684	1085	990	1077
Hypotheses, 9–14	1572	1569	1672	894	960	891
Hypotheses, 15–20	545	597	572	225	260	286
Hypotheses, 20–26	2198	2223	2216	1246	1268	1276
Hypotheses, > 26	627	558	592	512	470	430

Table 4 The distribution of the sentence length on entailment, neutral and contradiction. (ent. – entailment, neu. – neutral, con. – contradiction)

Length in syllable	Development set			Test set		
	ent. (%)	neu. (%)	con. (%)	ent. (%)	neu. (%)	con. (%)
Premises, ≤ 8	23.4	26.8	25.0	22.9	27.2	25.2
Premises, 9–14	26.4	23.2	24.8	26.1	22.4	24.3
Premises, 15–20	8.9	8.9	8.5	7.5	7.2	6.6
Premises, 20–26	33.0	32.9	32.9	32.5	32.2	32.1
Premises, > 26	8.3	8.2	8.8	10.9	10.9	11.9
Total	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Hypotheses, ≤ 8	26.9	26.8	25.0	27.4	25.1	27.2
Hypotheses, 9–14	23.3	23.2	24.8	22.6	24.3	22.5
Hypotheses, 15–20	8.1	8.8	8.5	5.7	6.6	7.2
Hypotheses, 20–26	32.5	32.9	32.9	31.4	32.1	32.2
Hypotheses, > 26	9.3	8.3	8.8	12.9	11.9	10.9
Total	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

The highest values are in bold

Table 5 The 40 highest frequency words which are common nouns and verbs in VnNewsNLI dataset

Ord	Word	Ord	Word	Ord	Word	Ord	Word
1	Tổng thống *(President)*	11	An ninh *(Security)*	21	Chỉ trích *(Criticize)*	31	Thủ tướng *(Prime Minister)*
2	Vắc xin *(Vaccine)*	12	Quốc hội *(Congress)*	22	Tranh cử *(Run for Election)*	32	Trở thành *(Become)*
3	Bang *(State)*	13	Điều tra *(Investigate)*	23	Cáo buộc *(Allegate)*	33	Vượt *(Excess)*
4	Bầu cử *(Vote)*	14	Súng *(Gun)*	24	Nhậm chức *(Take office)*	34	Dịch *(Disease)*
5	Biểu tình *(Protest)*	15	Tấn công *(Attack)*	25	Công bố *(Publish)*	35	Luật *(Law)*
6	Ủng hộ *(Support)*	16	Nhằm *(Aim)*	26	Thành phố *(City)*	36	Ứng viên *(Candidate)*
7	Chống *(Against)*	17	Cảnh báo *(Warn)*	27	Yêu cầu *(Require)*	37	Người dân *(Citizen)*
8	Tuyên bố *(Declare)*	18	Bạo loạn *(Violence)*	28	Y tế *(Medical)*	38	Hoạt động *(Activity)*
9	Kêu gọi *(Call)*	10	Phiếu *(Vote)*	29	Tuổi *(Age)*	39	Mạng *(Life)*
10	Cảnh sát *(Police)*	20	Tên lửa *(Rocket)*	30	Quốc gia *(Nation)*	40	Xe *(Vehicle)*

to recognise the equivalent sentences that are special cases of entailment samples.

To show the usefulness of our Vietnamese NLI dataset, we also conducted an answer selection experiment on has-answer samples of UIT-viQuAD 2.0. The results of this experiment are shown in Table 8. In Table 8, the viNLI model has the highest P@1 score of 0.4949 indicating the ability to choose the most appropriate sentence in a short paragraph with a given sentence. This result is higher than the results of two baselines TF-IDF with P@1 score of 0.4056 and BM25 with P@1 score of 0.3833, showing that viNLI model is applicable in Vietnamese answer selection.

In our experiments, we fine-tuned the viXNLI model on a small development set with about 2500 samples and tested it on two larger test sets with about 5000 and 12,000 samples. The results show that BERT pre-train models are possibly fine-tuned on small datasets to build effective models [7].

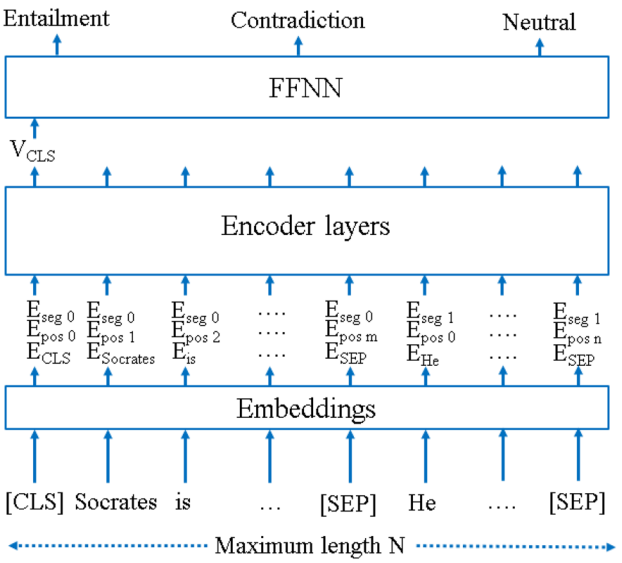


Fig. 6 The illustration of NLI BERT architecture [7]

Conclusion and Future Works

In this paper, we proposed a method of building a Vietnamese NLI dataset for fine-tuning and testing Vietnamese NLI models. This method aims at two issues. The first issue is the trained model's cue marks for identifying the relationship between a premise and a hypothesis without considering the premise. We addressed this issue by generating samples using eight types of premise-hypothesis pairs. The second issue is the Vietnamese writing style of samples. We addressed this issue by generating samples from titles and introductory sentences of Vietnamese news webpages. We used title-introductory pairs of appropriate webpages to reduce annotation costs. These samples were generated by applying a semi-automatic process. To evaluate our method, we built our VnNewsNLI dataset by extracting the title and the introductory sentence of many web pages in a Vietnamese news website VnExpress and applying our building process. When creating our VnNewsNLI, we had two people manually annotate each sentence to generate contraction sentences.

Table 6 The accuracy of viXNLI, viNLI and $viNLI_R$ models on test datasets

Model	Accuracy (%)	
	XNLI	VnNewsNLI
viXNLI	**68.64**	64.04
viNLI	41.47	**94.79**
$viNLI_R$	37.62	74.54

The highest values are in bold

Table 7 The accuracy of viXNLI, viNLI and $viNLI_R$ models on type 1 and type 2 samples of VnNewsNLI test set

Model	Accuracy (%) of type 1 and type 2 entailment
viXNLI	82.23
viNLI	**95.67**
$viNLI_R$	0.00

The highest values are in bold

Table 8 The P@1 scores of viXNLI, viNLI and $viNLI_R$ models on answer selection with two baselines TF-IDF and BM25

Model	P@1
viXNLI	0.4044
viNLI	**0.4949**
$viNLI_R$	0.1733
TF-IDF	0.4056
BM25	0.3833

The highest values are in bold

We evaluated our proposed method by comparing the results of a NLI model, viXNLI, fine-tuned on Vietnamese XNLI dataset and of a NLI model, viNLI, fine-tuned on our VnNewsNLI dataset. We used the same deep neural network architecture BERT for building these NLI models. The results showed that viNLI model had a higher accuracy (94.79% vs. 64.04%) on our VnNewsNLI test set while it had a lower accuracy (41.47% vs. 68.64%) on the Vietnamese XNLI test set when compared to viXNLI. To show the usefulness of our NLI dataset, we also conducted an answer selection experiment using viXNLI model, viNLI model and two baselines TF-IDF and BM25. The accuracy of 94.79% and the highest P@1 score of 0.4949 of viNLI model in the two experiments promised to build a high-quality Vietnamese NLI dataset from Vietnamese documents to ensure writing style.

Currently, our VnNewsNLI dataset contains a pretty small number of samples, with about 32,000 samples. In future, we will apply our proposed process for building a large and high-quality multi-genre Vietnamese NLI dataset. We will also train a Vietnamese NLI model to help develop our dataset by automatically suggesting the relation of a premise-hypothesis pair. This model might reduce our effort in building our dataset.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Punyakanok V, Roth D, Yih W-T. Natural language inference via dependency tree mapping: an application to question answering. Comput Linguist. 2004;6:10.
2. Lan W, Xu W. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. International Conference on Computational Linguistics, pp. 3890--3902. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018)
3. Minbyul J, Mujeen S, Gangwoo K, Donghyeon K, Wonjin Y, Jaehyo Y, Jaewoo K. Transferability of natural language inference to biomedical question answering. Conference and Labs of the Evaluation Forum, Thessaloniki, Greece (2020)
4. Falke T, Ribeiro LFR, Utama PA, Dagan I, Gurevych I. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. Annual Meeting of the Association for Computational Linguistics, pp. 2214--2220. Association for Computational Linguistics, Florence, Italy (2019)
5. Pasunuru R, Guo H, Bansal M. Towards Improving abstractive summarization via entailment generation. Workshop on New Frontiers in Summarization, pp. 27--32. Association for Computational Linguistics, Copenhagen, Denmark (2017)
6. Dagan I, Roth D, Sammons M, Zanzotto FM. Recognizing textual entailment: models and applications. Morgan & Claypool Publishers (2013)
7. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171--4186. Association for Computational Linguistics (2019)
8. Marelli M, Menini S, Baroni M, Bentivogli L, Bernardi R, Zamparelli R. A SICK cure for the evaluation of compositional distributional semantic models. International Conference on Language Resources and Evaluation, pp. 216--223. European Language Resources Association, Reykjavik, Iceland (2014)
9. Bowman SR, Angeli G, Potts C, Manning CD. A large annotated corpus for learning natural language inference. Conference on Empirical Methods in Natural Language Processing, pp. 632--642. Association for Computational Linguistics, Lisbon, Portugal (2015)
10. Williams A, Nangia N, Bowman SR. A broad-coverage challenge corpus for sentence understanding through inference. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1112--1122. Association for Computational Linguistics, New Orleans, Louisiana (2017)
11. Conneau A, Rinott R, Lample G, Williams A, Bowman S, Schwenk H, Stoyanov V. XNLI: Evaluating cross-lingual sentence representations. Conference on Empirical Methods in Natural Language Processing, pp. 2475--2485. Association for Computational Linguistics, Brussels, Belgium (2018)
12. Nguyen M-T, Ha Q-T, Nguyen T-D, Nguyen T-T, Nguyen L-M. Recognizing textual entailment in vietnamese text: an experimental study. International Conference on Knowledge and Systems Engineering, pp. 108--113. IEEE, Ho Chi Minh City, Vietnam (2015)

13. Nguyen DQ, Nguyen AT. PhoBERT: pre-trained language models for Vietnamese. Conference on Empirical Methods in Natural Language; 2020. pp. 1037—1042.
14. Jiang N, de Marneffe M-C. Evaluating BERT for natural language inference: a case study on the CommitmentBank. Conference on Empirical Methods in Natural Language Processing, pp. 6086--6091. Association for Computational Linguistics, Hong Kong, China (2019).
15. Nguyen CT, Nguyen DT. Building a Vietnamese dataset for natural language inference models. Future Data and Security Engineering; 2021, pp. 185--199. Springer, Singapore.
16. Nguyen KV, Tran SQ, Nguyen LT, Huynh TV, Luu ST, Nguyen NL-T. VLSP 2021 Shared Task: Vietnamese Machine reading comprehension. Kiet Van Nguyen, The 8th International Workshop on Vietnamese Language and Speech Processing (VLSP 2021) (2021).
17. Nguyen KV, Nguyen V, Nguyen A, Nguyen N. A Vietnamese dataset for evaluating machine reading comprehension. International Conference on Computational Linguistics, pp. 2595--2605. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020)
18. Tenney I, Das D, Pavlick E. BERT rediscovers the classical NLP pipeline. Annual Meeting of the Association for Computational Linguistics, pp. 4593--4601. Association for Computational Linguistics, Florence, Italy (2019)
19. Rogers A, Kovaleva O, Rumshisky A. A primer in BERTology: What we know about how BERT works. Trans Assoc Comput Linguistics. 2020;8:842–66.
20. Peters ME, Neumann M, Zettlemoyer L, Yih W-T. Dissecting contextual word embeddings: architecture and representation. Conference on Empirical Methods in Natural Language Processing, pp. 1499--1509. Association for Computational Linguistics, Brussels, Belgium (2018)
21. Wolf T, Chaumond J, Debut L, Sanh V, Delangue C, Moi A, Cistac P, Funtowicz M, Davison J, Shleifer S, others. Transformers: state-of-the-art natural language processing. Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38--45. Association for Computational Linguistics (2020).
22. Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, Grangier D, Auli M. fairseq: a fast, extensible toolkit for sequence modeling. Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 48--53. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
23. Vu T, Nguyen DQ, Nguyen DQ, Dras M, Johnson M. VnCoreNLP: a vietnamese natural language processing toolkit. Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 56--60. Association for Computational Linguistics, New Orleans, Louisiana (2018).