A Factoid Question Answering System for Vietnamese

Phuong Le-Hong
Hanoi University of Science, Vietnam
FPT Technology Research Institute, FPT University
phuonglh@vnu.edu.vn

Duc-Thien Bui
FPT Technology Research Institute
FPT University, Hanoi, Vietnam
thienbd@fpt.edu.vn

ABSTRACT

In this paper, we describe the development of an end-to-end factoid question answering system for the Vietnamese language. This system combines both statistical models and ontology-based methods in a chain of processing modules to provide high-quality mappings from natural language text to entities. We present the challenges in the development of such an intelligent user interface for an isolating language like Vietnamese and show that techniques developed for inflectional languages cannot be applied “as is”. Our question answering system can answer a wide range of general knowledge questions with promising accuracy on a test set.

KEYWORDS

question answering, ontology, knowledge bases, hybrid, Vietnamese

ACM Reference Format:
Phuong Le-Hong and Duc-Thien Bui. 2018. A Factoid Question Answering System for Vietnamese . In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3184558.3191535

1 INTRODUCTION

Question Answering (QA) has been an important line of research in natural language processing in general and human-machine interface in particular. The ultimate goal of a QA system is to provide a concise and exact answer to a question asked in a natural language. For example, the answer to the question “*Which French city has the largest population?*” should be “*Paris*”.

Open-domain QA is a challenging task because the research and validation of a precise answer to a question require a good understanding of the question itself and of the text containing the potential answer. Typically we need to carry out both syntactic and semantic analyses in order to fully understand a question and pinpoint an answer. This is much more difficult than the task of common information retrieval, where one only needs to present a ranked list of documents in response to a question, which can be efficiently performed by available search engines.

The state-of-the-art techniques in open-domain QA can be classified into two main categories, namely semantic parsing based techniques and information retrieval based techniques [2]. Semantic parsing systems try to interpret the meaning of a question correctly by semantic analysis. A correct interpretation converts the question into an exact database query that returns a correct answer. On the other hand, information retrieval based systems first transform a question into a valid query, then retrieve a set of candidate answers by querying a corpus and/or a knowledge base, and finally use fine-grained heuristics to identify the exact answer.

Although both kinds of system require human expertise to handcraft linguistic resources including lexicons, grammars and knowledge bases, the information retrieval based approach is more suitable to less-resourced languages since many advanced natural language processing tools such as syntactic and semantic parsers are not readily available. Furthermore, as shown in many previous studies on building QA systems, existing methods developed for well-studied languages are not easily and conveniently applied or scaled up to natural languages other than English.

In this paper we present a QA system for the Vietnamese language which combines both statistical models and knowledge-based methods in a chain of processing modules to provide high-quality mappings from natural language text to entities. We present the challenges in the development of such an intelligent user interface for an isolating language such as Vietnamese and show that techniques developed for inflectional languages cannot be applied “as is”. Our question answering system can answer a wide range of general knowledge questions with a promising accuracy on a test set. The system is released as open-source software in the hope that it will serve as a baseline for future developments of question answering systems for Vietnamese.

The remainder of this paper is structured as follows. First, the next section gives a survey of existing work in the line of this research. Next, we describe the methodology that we use to develop our QA system. Then, we present our experiments and evaluation results. Finally, the last section concludes the paper and suggests some directions for future work.

2 RELATED WORK

There have been some existing studies on building and evaluating QA system for Vietnamese. In this section, we present a survey of existing work, compare and hightlight the difference between them and this work.

Tran [21] discussed a specific QA system for Vietnamese person named entity which focuses on only “*who*”, “*whom*” and “*whose*” questions. To this end, the diversity of answerable questions are rather limited. A prior work of the same research group [22] presented an experimental study of a QA system for Vietnamese which utilized a search engine to search for answers. This system is restricted to travelling domain and was tested on only a small test set containing one hundred questions. Duong [5] presented a QA system for use in Vietnamese legal documents which is able to answer simple questions about procedures and sanctions in law on business. This system uses a similarity-based model and the Lucene

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3191535

document search engine to retrieve candiate documents and extract answers. Compared to these works, our QA system differs in three aspects. First, it is open domain which can provide answers to a much wider range of questions other than a specific domain or person named entity question types. Second, our system does not use a search engine to retrieve and rank documents but relies on a large knowledge base. And third, our system is evaluated on a test set of about ten times larger which covers a wide variety of questions, resulting in a promising accuracy.

Most recently, Nguyen [17] presented a QA system for Vietnamese which uses semantic web information to provide answers to user's queries. Together with a series of previous publications in the same line of research, this group developed the KbQAS system which is claimed to be the first knowledge-based QA system for the language.[1] A key component of their system is a knowledge acquisition module which utilizes the single classification ripple down rules method for question analysis. This is a typical rule-based system. Although their method is able to acquire rules in a consistent and systematic manner, the knowledge bases are required to be built from scratch and an adaptation to a new domain or language still requires time and effort of human expertise. As reported, this system contains 92 manual rules and was tested in a test set of 74 Vietnamese questions. In contrast to this work, our system utilizes both statistical and rule-based approaches, a large ontology base (DBPedia), and the Cypher query language – a high-performance declarative language for query graph database. Our system is also validated on a much larger test set of diverse questions, totaling nearly 900 question and answer pairs.

3 METHODOLOGY

3.1 DBPedia and Graph Model

Our QA system uses an ontology developed by the DBPedia project [13].[2] DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web for a wide number of languages, including Vietnamese. The DBPedia knowledge bases have become an important source of structured information on the emerging *Web of Data* [16].

DBPedia is an ontology according to the definition of W3C[3] in that it defines the terms used to describe and represent an area of knowledge. Figure 1 shows an excerpt of the DBPedia ontology. This ontology says that there is a class called `Writer` which is a subclass of `Artist` which is in turn a subclass of `Person`. There is a property relating an instance of the class `Work` to an instance of the class `Person`. For example, the novel titled "*Angel and Daemon*" is an instance of class `Work` and related via property `author` to its author "*Dan Brown*".

The DBPedia ontology can also be viewed as a property graph model made up of nodes, relationships and properties [20]. Nodes contain properties in the form of key-value pairs; the keys are strings and the values are arbitrary data types. Relationships connect and structure nodes. A relationship always has a direction, a label, a start node and an end node; the direction and label add semantic clarity to the structuring of nodes. It is noted that like nodes, relationships can also have properties which provide not only additional semantics but also metadata for graph algorithms and help constrain queries at runtime.

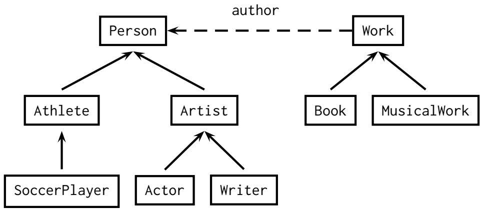

Figure 1: An excerpt of the DBPedia ontology

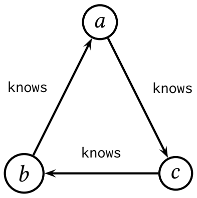

Figure 2: A simple graph pattern, expressed using a diagram

We have constructed a graph database from the dump files of the Vietnamese DBPedia ontology. The total size of these files is about 5 GB. The database is of size 1.5GB, consisting of one million nodes, 2.5 million links and 7.5 million properties.

3.2 Query Language

We use Cypher to query the DBPedia ontology. Cypher is an expressive and compact graph database query language. This language is specific to Neo4j[4] which is a good and well-known graph database used by many organizations in production applications. The major advantages of Cypher are that it is easy to learn, easy to use and ideal for describing graphs programmatically in a precise fashion.

It is noted that other graph databases have other means of querying data. Many graph databases support the RDF query language SPARQL. However, in building a QA system, we are interested in the expressive power of a property graph combined with an advantageous delarative query language. For this reason we chose Cypher to query the database to find data matching a specific pattern.

Cypher enables a user (or an application) to ask the database to find data that matches a specific pattern. Figure 2 shows an example of a simple pattern. This pattern describes three mutual friends.

Like most query languages, Cypher is composed of clauses. The simplest queries consist of a `START` clause followed by a `MATCH` and a `RETURN` clause. An example of a Cypher query that uses these three clauses to find the mutual friends of user named `Michael` is:

```
START a = node:user(name="Michael")
MATCH (a)-[:knows]->(b)-[:knows]->(c),
  (a)-[:knows]->(c)
RETURN b, c
```

[1] In their work, the term "knowledge base" may lead to confusion in that it really refers to a set of rules rather than an ontology base of entities and relations.
[2] http://www.dbpedia.org/
[3] http://www.w3.org/standards/semanticweb/
[4] http://www.neo4j.com/

The other clauses that we can use in a Cypher query include `WHERE`, `CREATE` and `CREATE UNIQUE`, `DELETE`, `SET`, `FOREACH`, `UNION` and `WITH`. These clauses allow for expressive and efficient querying and updating of the graph database. For details please refer to the documentation page of the Cypher query language.[5]

In the following subsections we present briefly some important Vietnamese language processing modules which are integrated in our QA system. These modules deal with basic processing tasks of Vietnamese including word segmentation, part-of-speech tagging and question classification. Due to space restriction, we do not present in this paper the general characteristics of the Vietnamese language which are discussed in detail in [10].

3.3 Word Segmentation

Word segmentation or tokenization is the problem of dividing a string of written language into its component words. In English and many occidental languages, the space is a good approximation of a word delimiter. However, many languages do not have a trivial word segmentation process. For example, in Chinese or Japanese, sentences but not words are delimited; in Thai and Lao, phrases and sentences but not words are delimited, and as presented in the previous section, in Vietnamese syllables but not words are delimited. Word segmentation is in fact a difficult problem for these languages.

In particular, there are two types of ambiguities that we must deal with in Vietnamese word segmentation. The first ambiguity is called *overlap ambiguity* where some adjacent syllables can have different word segmentations and their validity cannot be determined completely without resorting to a syntactic or semantic resolution of the entire sentence. For example, the three-syllable phrase "*thuộc địa bàn*" can have two word segmentations, either "*(thuộc địa) (bàn)*" or "*(thuộc) (địa bàn)*", depending on context. A more complicated example is with a four-syllable phrase "*tổ hợp âm tiết*" where all the words *tổ hợp*, *hợp âm*, and *âm tiết* are valid, and hence all possibly different overlapping word segmentations are plausible. The second ambiguity is called the *combination ambiguity* where two adjacent syllables can either be divided or combined to make words. For example, two syllables "*chanh chua*" can form an adjective which means to have a sharp tongue, or they can form two words *chanh* and *chua*, a noun phrase which means a sour lemon.

Although Vietnamese word segmentation is difficult, there exists efficient approaches to solve this problem which have been published by the Vietnamese language processing community. In this work we adopt the approach of [9] which is consistent and has a good accuracy in the range of 96%–98% on different test sets.

3.4 Part-of-Speech Tagging

Part-of-speech (POS) tagging, also called grammatical tagging or word-category disambiguation, is the problem of automatically determining each word in a sentence as corresponding to a particular part-of-speech such as noun, verb, adjective, adverb, etc. POS tagging is not an easy problem since many words can represent more than one part of speech on different occasions.

For well-studied languages like English or certain other occidental languages, POS tagging is a solved problem with very high accuracy, about 97.3%, which is believed to be as high as human performance [15]. However the accuracy of Vietnamese POS tagging is much lower than that of English. The combination of the best machine learning algorithms and the best features in discriminative sequence models has achieved an accuracy of about 93.5% [11]. As presented in the previous section, an important reason for the inferior accuracy of Vietnamese POS tagging is its inherent difficulty. It is not easy to determine a clear syntactic function of many Vietnamese words while syntactic category mutation is a frequent phenomenon. Furthermore, POS tagging depends heavily on word segmentation, which is a difficult task as presented in the previous section.

3.5 Question Classification

The first step of understanding a question is to perform question analysis. Question classification is an important task of question analysis which detects the answer type of the question. Question classification helps not only to filter out a wide range of candidate answers but also to determine answer selection strategies. For example, if one knows that the answer type is *city*, one can restrict candidate answers as cities instead of consider every noun phrase of a document providing the answer.

At first glance, one may think that question classification can be framed as a text classification task. However, there exists characteristics of question classification that distinguish it from the common task. Firstly, a question is relatively short and contains less word-based information than an entire text. Secondly, a short question needs deeper analysis to reveal its hidden semantics. Therefore application of text classification algorithms *per se* to question classification cannot produce good results. Furthermore natural languages are inherently ambiguous, thus the question classification is not trivial, especially for *what* and *which* type questions. For example "*What is the capital of France?*" is of location (city) type, while "*What is the Internet of things?*" is of definition type. Consider also these examples: *(1) What tourist attractions are there in Reims? (2) What do most tourists visit in Reims? (3) What are the names of the tourist attractions in Reims? (4) What attracts tourists to Reims? (5) What is worth seeing in Reims?* [14]; all these questions are of the same answer type: location. Different wording and syntactic structures classification difficult [6].

With the increasing popularity of statistical approaches to natural language processing in general and to question classification in particular, recent years have seen many machine learning approaches which have been applied to the problem of question classification. The main advantage of machine learning approaches is that one can learn a statistical model using useful features extracted from a sufficiently large set of labeled questions and then use it to automatically classify new questions.

We use the method proposed by [12] in our question classification module. In contrast to many existing approaches for question classification which make use of very rich feature spaces or hand-crafted rules, this method proposes a compact yet effective feature set. In particular it uses typed dependencies as semantic features. It has been shown that by integrating only two simple dependencies

[5] http://neo4j.com/docs/stable/cypher-query-lang.html

of types nominal subject and prepositional object, one can improve the accuracy of question classification by over 8.0% using common statistical classifiers over two benchmark datasets, the UIUC dataset for English and a recently introduced FPT question dataset for Vietnamese. With unigram feature and typed dependency feature, one can obtain accuracy of 87.6% and 80.5% using maximum entropy classification for the UIUC and FPT question dataset respectively. It is worth noting that the best question classification accuracy on the UIUC dataset is 89.00% by [6], where important features like head words and their hypernyms are included. Such semantic features are not readily available for less-resourced languages such as Vietnamese, where a WordNet is still in its first stage of construction.

3.6 Textual Question to Cypher Query Transformation

An important module of our knowledge-based QA system is the module that transforms textual questions in Vietnamese into equivalent Cypher queries. The queries are then executed to search for answers to the questions. This section describes the main processing steps of this module.

First, a textual question is processed by the NLP chain presented above, ranging from word segmentation to part-of-speech tagging and question classification. For example a question such as "*Thành viên chủ chốt của tập đoàn FPT là những ai?*" ("Who are the most important people of FPT Corporation?") will be analysed as follows:

Word Segmentation: The output is a sequence of words: *[Thành_viên, chủ_chốt, của, tập_đoàn, FPT, là_những_ai]*. Here the underscore character is used to connect the syllables of a word and words are separated by commas.

Part-of-speech Tagging: The output is a sequence of tagged words: *[Thành_viên*/N , *chủ_chốt*/N, *của*/E, *tập_đoàn*/N, *FPT*/Np, *là_những_ai?*/QW*]*. In this step, each word of the question is tagged as a part-of-speech, where N denotes a common noun, Np denotes a proper noun, E denotes a preposition and so on.

Key Word Extraction: In this step stop words or unimportant words are stripped out of the question. Only key words are retained. In the example above, the word *của*/E is removed.

Question Classification: This step determines the answer type of the question, that is the information type we need to find. In this example the answer type is HUM (human) since the question asks for a person (or a group of people). Some other answer types are NUM (number), DTIME (datetime), YESNO (yes/no), etc. Details of the question types, statistical models and classification techniques are presented in [12].

Entity Construction: Since we are querying a graph model which is made up of entities (nodes), relationships and properties, we need to construct a set of entities, relationships and properties which are implied in the query at hand. This step is crucial in building a good corresponding Cypher query for a textual question. In essence:

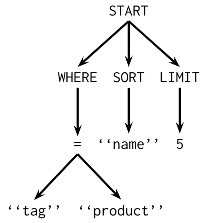

Figure 3: An example of a Cypher syntactic tree

- The proper nouns correspond to the names of the nodes in the graph database. They are recognized by using their part-of-speech tag Np.
- The remaining words are classified as either properties or relationships, depending on their probabilities on a datasets, using a built-in dictionary.

More specifically, our approach combines a rule-based extractor and a statistical-based classifier to perform entity construction. A rule-based extractor is used to extract named entities such as persons, organizations or locations by relying on the output of a part-of-speech tagger. A logistic regression model is used to predict the likelihood of being a property or a relationship for each remaining keyword in the query.

To continue with the example above, this step determines FPT as entity, and thànhViênChủChốt as relationship, and there is no property for this question.

Cypher Query Construction: In this last step we first build a syntactic tree representing a cypher query which corresponds to the textual input question. The nodes in the syntactic tree correspond to either the Cypher clauses or the operators ("=", ">", "<", ...), starting from the root node whose name is START. The leaf nodes of the tree correspond to key words or values. Figure 3 shows an example of a syntactic tree. Once the syntactic tree has been built, we search for appropriate replacements of leaf nodes with the elements determined in the previous step. Since there may be multiple plausible replacements for leaf nodes, a syntactic tree may generate multiple Cypher queries. This is expected because in the Vietnamese DBPedia graph database a key word can either be a link of a node or be a property of another node. A Cypher query for the original question is

```
START x=node:DBPediaIndex(key="FPT")
RETURN DISTINCT x.thànhViênChủChốt
```

As another complete example, consider the following question: "*Dân số và diện tích của Hà Nội là bao nhiêu?*" (What is the population and area of Hanoi?). This question is analysed by the processing chain above, where the intermediate results and the final Cypher query are as follows:

(1) Word Segmentation and Part-of-Speech Tagging:
[Dân_số/N *và*/E *diện_tích*/N *của*/A *Hà_Nội*/Np *là_bao_nhiêu?/QW*]

(2) Key Word Extraction:
[Dân_số/N, *diện_tích*/N, *Hà_Nội*/Np*]*

(3) Question Classification: The answer type is NUM

Table 1: Some question types and examples

Question Type	Example	English Translation
E là [ai | gì | ở đâu | . . .]?	Nguyễn Tấn Dũng là ai? Facebook là gì?	Who is Nguyen Tan Dung? What is Facebook?
p của E [như thế nào | bằng bao nhiêu | . . .]?	Dân số của Hà Nội bằng bao nhiêu?	What is the population of Hanoi?
r của E là gì?	Thủ đô của Thái Lan là gì?	What is the capital of Thailand?
r của E [như thế nào | bằng bao nhiêu | . . .]?	Vợ của thủ tướng Nguyễn Tấn Dũng là ai? Chủ tịch HQT tập đoàn FPT là ai?	Who is the wife of Prime Minister Nguyen Tan Dung? Who is the chairman of FPT Corp.?
p của r của E là gì?	Tên của vợ vua Trần Thái Tông là gì? Nơi sinh của chủ tịch UBND TP. Hà Nội ở đâu?	What is the name of King Tran Thai Tong's wife? What is the place of birth of the chairman of Hanoi People's Committee?
E_1 và E_2 có r là gì?	Việt Nam và Thái Lan có thủ đô là gì?	What are the capitals of Vietnam and Thailand?

(4) Entity Construction: Properties = {Dân_số, Diện_tích};
Named Entity = {Hà_Nội}
(5) Cypher Query Construction:
START n=node:DBPedia(key="Hà_Nội")
RETURN n.dânSố, n.diệnTích

In the next section, we present experimental results of our QA system and discussion.

4 EXPERIMENTS

Our aim is to build a QA system which is able to answer Vietnamese factoid questions on a broad range of topics from the DBPedia ontology with high accuracy. We have developed an algorithm to transform different questions to corresponding Cypher queries following the methodology described above. The system can answer a wide variety of questions of diverse types, which are shown in the Table 1.

In this table E, p and r represents an entity, a property and a relationship respectively, and the vertical character '|' is used to represent alternative choices. The last row of the table shows a complicated question type where we seek the same relationship of two different entities (here, the capital). It is also extended further to account for more complicated questions where a user wants to seek for some comparative information, such as in the following example question where the area and population of two different Vietnamese provinces are queried:

"*Diện tích và dân số của Hà Nội và Thái Bình bằng bao nhiêu?*" (What are the area and population of Hanoi and Thaibinh?)

It is worth noting that the system can effectively deal with different variants of the same question since different syntactically correct word orders are identified and analysed. For example, to query the population of Hanoi, one can use either of the two following paraphrases:

Dân số của Hà Nội là bao nhiêu?
Hà Nội có dân số bằng bao nhiêu?
(What is the population of Hanoi?)

Or to ask for the country whose capital is Bangkok, a Vietnamese speaker can use either of the two following choices:

Bangkok là thủ đô của nước nào?
Nước nào có thủ đô là Bangkok?
(What country's capital is Bangkok?)

To evaluate the performance of the system, we have manually built a dataset of 879 question-answer pairs about person, location and other facts where the answers can be found in the Vietnamese Wikipedia.[6] To understand the performance further, in addition to the accuracy of the final answer, the accuracy of query transformation is also evaluated. The accuracy of our system is shown in Table 2.

Table 2: Accuracy of the system

Accuracy of the QA system	76.90%
Accuracy of the query construction module	97.50%

The system is able to give correct answers for 76.70% of the questions in the test set. If the system does not find an answer to a question, it is counted as an incorrect result for that question.

The current test set contains the following types of questions of different difficulty levels:

- Questions about an entity of the form "*Who/What/... is E*". For example, "*Who is Barack Obama?*" or "*Where is Paris?*" (in Vietnamese language).
- Questions about a feature or property of an entity, for example "*Who is the spouse of Barack Obama?*", "*What is the population density of Hanoi?*", or a trickier question such as "*What is the population of the capital of Argentina?.*
- Questions about the same relationship of two different entities, for example "*What are the capitals of France and Germany?*. Here, France and Germany are two entities and the same relationship is the capital. The correct answer for this question should be "Paris" and "Berlin".

Table 1 gives some more examples of these types of questions along with their English translation.

Our manual test set also contains the correct Cypher query for each question so that the automatic query construction module can also be evaluated. Table 3 shows two samples in our test set.

Our QA system is designed to provide a short answer to factoid questions. However by querying the DBPedia graph database built on top of the Wikipedia it can also answer many questions in great details in the sense that it can also show the answer text for the question whenever it is available. For example, the following snippet shows the answer to the question "*Tác giả của Truyện Kiều là ai?*" (Who is the author of The Tale of Kieu?). Once the system finds the short answer "*Nguyễn Du*" for this question, it can retrieve and show the summary of the corresponding entry in the Wikipedia, giving the following result:

[6] Currently, the Vietnamese Wikipedia contains about 1, 140, 000 articles according to https://stats.wikimedia.org/EN/.

Table 3: Some samples in our test set

```
<qa type="NUM">
  <question>Số nhân viên của FPT là bao nhiêu?</question>
  <cypher>
    START x=node:DBPediaIndex(key="FPT") RETURN DISTINCT x.sốNhânViên, x.chiTiết LIMIT 20
  </cypher>
  <correctCypher>true</correctCypher>
  <answer>
    FPT, tên viết tắt bằng tiếng Anh của Công ty cổ phần FPT (tên cũ của Công ty là
    Công ty cổ phần phát triển đầu tư công nghệ FPT), là một tập đoàn kinh tế tại Việt Nam
    với lĩnh vực kinh doanh chính là cung cấp các dịch vụ liên quan công nghệ thông tin.
  </answer>
  <correctAnswer>true</correctAnswer>
  <speed>134</speed>
</qa>
<qa type="HUM">
  <question>Ai là vợ của Tôn Đức Thắng?</question>
  <cypher>
    START x=node:DBPediaIndex(key="Tôn Đức Thắng") MATCH x-[r:vợChồng]-y RETURN DISTINCT
      y.chiTiết LIMIT 20 START x=node:DBPediaIndex(key="Tôn Đức Thắng") RETURN DISTINCT
      x.vợChồng, x.chiTiết LIMIT 20
  </cypher>
  <correctCypher>true</correctCypher>
  <answer>Đoàn Thị Giàu</answer>
  <correctAnswer>true</correctAnswer>
  <speed>153</speed>
</qa>
```

<question>Tác giả của Truyện Kiều là ai?</question> <answer>Nguyễn Du tên chữ Tố Như, hiệu Thanh Hiên, biệt hiệu Hồng Sơn lạp hộ, là một nhà thơ nổi tiếng thời Lê mạt, Nguyễn sơ ở Việt Nam. Ông là một nhà thơ lớn của Việt Nam, được người Việt kính trọng gọi ông là "Đại thi hào dân tộc". Năm 1965, Nguyễn Du được Hội đồng hòa bình thế giới công nhận là danh nhân văn hóa thế giới và ra quyết định kỷ niệm trọng thể nhân dịp 200 năm năm sinh của ông. </answer>

Our QA system has a good speed in that it can answer a question in average 0.04 second on a personal computer. Our system will be released as an open-source project and freely available for research purpose. We believe that our system will be useful for the Vietnamese language processing community. At the moment, our demo system is available for testing at http://124.158.5.68:8080/wiki-qa/.

5 CONCLUSION

This paper presented the development of an open-domain question answering system for the Vietnamese language. The system combines both statistical models and ontology-based methods in a chain of processing modules to provide high-quality mappings from natural language text to entities. It can answer a wide range of general knowledge questions with promising accuracy on a test set. It is released as an open-source software project in the hope that it will serve as a baseline for future development of question answer systems for Vietnamese.[7].

With the rise of available large scale structured knowledge bases, we think that the most promising approach to open-domain question answering is the ability to query efficiently such databases in natural languages. In this work we concentrated on exploiting DBPedia, a freely available database of facts which are extracted from the Wikipedia. Nevertheless there exists other good knowledge bases such as Freebase [1], an open shared database of the world's knowledge, which has been shown to be very useful for many applications including question answering. We plan to investigate how we can use the Vietnamese section of this knowledge base in our system in a future work. We also plan to perform some comparisons with other approaches that can find answers directly from Wikipedia texts to show the benefit of quering an ontology.

Current good question answering systems make use of additional natural language processing modules such as dependency parsing or semantic role labelling [4]. We would like to improve further the performance of our system by integrating recently available dependency parser [8], semantic role labeller [18] and named entity recognizer for Vietnamese [7].

[7] The temporary demo link of our system is at http://124.158.5.68:8080/wiki-qa/

Finally recent works on open-domain question answering [2, 3] have shown the efficiency of embedding models, which learn low dimensional vector representations of words and knowledge bases constituents to achieve better accuracy. How these models can be used to improve our current system is another interesting line of research that we would like to research in a future work, following some recent results [19].

ACKNOWLEDGMENTS

This research is partly funded by FPT Technology Innovation, FPT Corporation. The first author is partly funded by the Vietnam National University, Hanoi (VNU) under project number QG.15.04. We are grateful to our anonymous reviewers for their helpful comments which helped us improve the quality of the article.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaborative created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 1247–1250.

[2] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. In *Proceedings of EMNLP*. Doha, Qatar, 615–620.

[3] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, Vol. 8724. Springer Berlin Heidelberg, 165–180.

[4] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of ACM SIGIR*. Salvador, Brazil, 400–407.

[5] Huu-Thanh Duong and Bao-Quoc Ho. 2014. A Vietnamese Question Answering System in Vietnam's Legal Documents. In *Proceedings of the 13th IFIP TC8 International Conference, CISIM*. Ho Chi Minh City, Vietnam, 186–197.

[6] Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. Question Classification using Head Words and their Hypernyms. In *Proceedings of the 2008 Conference on EMNLP*. 927–936.

[7] Phuong Le-Hong. 2016. Vietnamese named entity recognition using token regular expressions and bidirectional inference. In *arXiv preprint arXiv:1610.05652*.

[8] Phuong Le-Hong, Thi-Minh-Huyen Nguyen, Thi-Luong Nguyen, and My-Linh Ha. 2015. Fast Dependency Parsing using Distributed Word Representations. In *Trends and Applications in Knowledge Discovery and Data Mining*. Lecture Notes in Artificial Intelligence, Vol. 9441. Springer.

[9] Phuong Le-Hong, Thi Minh Huyen Nguyen, Azim Roussanaly, and Tuong Vinh Ho. 2008. A Hybrid Approach to Word Segmentation of Vietnamese Texts. In *Language and Automata Theory and Applications*. Lecture Notes in Computer Science, Vol. 5196. Springer Berlin Heidelberg, 240–249.

[10] Phuong Le-Hong, Azim Roussanaly, and Thi-Minh-Huyen Nguyen. 2015. A syntactic component for Vietnamese language processing. *Journal of Language Modelling* 3, 1 (2015), 145–184.

[11] Phuong Le-Hong, Azim Roussanaly, Thi Minh Huyen Nguyen, and Mathias Rossignol. 2010. An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts. In *Actes de Traitement Automatique des Langues*. Montreal, Canada, 50–61.

[12] Phuong Le-Hong, Phan Xuan-Hieu, and Nguyen Tien-Dung. 2014. Using Dependency Analysis to Improve Question Classification. In *Knowledge and Systems Engineering*. Advances in Intelligent Systems and Computing, Vol. 326. Springer, 653–665.

[13] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. 2015. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.

[14] Xin Li and Dan Roth. 2006. Learning Question Classifiers: the Role of Semantic Information. *Natural Language Engineering* 12, 3 (2006), 229–249.

[15] Christophe D. Manning. 2011. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In *Computational Linguistics and Intelligent Text Processing*. Theoretical Computer Science and General Issues, Vol. 6608. Springer-Verlag Berlin Heidelberg, 171–189.

[16] Mohamed Morsey, Jens Lehmann, Sören Auer, Claus Stadler, and Sebastian Hellmann. 2012. DBpedia and the live extraction of structured data from Wikipedia. *Program: electronic library and information systems* 46, 2 (2012), 157–181.

[17] Dat Quoc Nguyen, Dai Quoc Nguyen, and Bao Son Pham. 2017. Ripple Down Rules for Question Answering. *Semantic Web* 8, 4 (2017), 511–532.

[18] Thai-Hoang Pham, Xuan-Khoai Pham, and Phuong Le-Hong. 2015. Building a Semantic Role Labelling System for Vietnamese. In *Proceedings of the 10th International Conference on Digital Information Management*. IEEE, Jeju Islands, South Korea, 77–84.

[19] Thai-Hoang Pham, Xuan-Khoai Pham, Tuan-Anh Nguyen, and Phuong Le-Hong. 2017. NNVLP: A Neural Network-Based Vietnamese Language Processing Toolkit. In *Proceedings of IJCNLP, Demonstration papers*. Taipei, Taiwan.

[20] Ian Robinson, Jim Webber, and Emil Eifrem. 2015. *Graph Databases: New Opportunities for Connected Data* (2nd ed.). O'Reilly Media.

[21] Mai-Vu Tran, Duc-Trong Le, Xuan-Tu Tran, and Tien-Tung Nguyen. 2012. A Model for Vietnamese Person Named Entity. In *Proceedings of the 26th Pacific Asia Conference on Language, Information and Computation*. Bali, Indonesia, 325–332.

[22] Mai-Vu Tran, Vinh Duc Nguyen, Oanh Thi Tran, Uyen Thi Thu Pham, and Thuy Quang Ha. 2009. An Experiment Study of Vietnamese Question Answering System. In *Proceedings of the 2009 International Conference on Asian Language Processing*. Singapore, 152–155.