Applied Intelligence (2022) 52:636–651
https://doi.org/10.1007/s10489-021-02460-w

Developing a BERT based triple classification model using knowledge graph embedding for question answering system

Phuc Do [1] · Truong H. V. Phan [1,2]

Accepted: 20 April 2021 / Published online: 8 May 2021

Abstract
The current BERT-based question answering systems use a question and a contextual text to find the answer. This causes the systems to return wrong answers or nothing if the text contains irrelevant contents with the input question. Besides, the systems haven't answered yes-no and aggregate questions yet. Besides that, the systems only concentrate on the contents of text regardless of the relationship between entities in the corpus. This systems cannot validate the answer. In this paper, we presented a solution to solve these issues by using the BERT model and the knowledge graph to enhance a question answering system. We combined content-based and linked-based information for knowledge graph representation learning and classified triples into one of three classes such as base class, derived class, or non-existent class. We then used the BERT model to build two classifiers: BERT-based text classification for content information and BERT-based triple classification for link information. The former was able to make a contextual embedding vector for representing triples that were used to classify into the three above classes. The latter generated all path instances from all meta paths of a large heterogeneous information network by running the Motif Search method of Apache Spark on a distributed environment. After creating the path instances, we produced triples from these path instances. We made content-based information by converting triples into natural language text with labels and considered them as a text classification problem. Our proposed solution outperformed other embedding methods with an average accuracy of 92.34% on benchmark datasets and the Motif Finding algorithm with an average executive time improvement of 37% on the distributed environment.

Keywords BERT based triple classification model · Knowledge graph embedding · Meta-path · Motif finding

1 Introduction

Using the knowledge graph (KG) for enhancing the question answering system is a promising study in recent years and plays an important role in natural language Q&A systems [9]. Plus, the emergence of a pre-trained language model as BERT brings significant improvements in QA systems [12]. However, the KG based QA systems only focus on entities as head, predicate, tail, and links without content. Meanwhile, BERT-based QA systems use the content of the contextual text to find answers without relationships of entities. So, we combined both BERT and knowledge graph to take advantage of their pros. The KG-BERT model that is proposed by Liang Yao also uses BERT and knowledge graph to classify triples [15] for knowledge graph completion. In our model, we use KG-BERT and knowledge graph to improve the question answering system. The aggregate question can be answered by collecting all triples of knowledge graph satisfying the question. Our method generated triples by obtaining all meta-paths from a given HIN and then used Motif Search of Apache Spark to discover all path instances of the meta-paths in distributed environment because of the large knowledge graph. Afterward, our model generated all triples from these path instances. These triples can be based triple for meta-path of length 1 or derived triple for meta-path of length more than 1, then we convert the triple <h, p, r> into text by using the description of head, tail and predicate of triple. The text description of triple is fed into BERT as text classifier. Our model used a softmax classifier to compute the score function of triples. Our model classified triples into three classes such

✉ Phuc Do
phucdo@uit.edu.vn

Truong H. V. Phan
truongphv.ncs@grad.uit.edu.vn; truong.phv@vlu.edu.vn

[1] University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam

[2] Van Lang University, Ho Chi Minh City, Vietnam

as based class, derived class, and non-existent class. These steps are new compared with the original model of by Liang Yao. We compared our method with Yao's model to clarify the difference in terms of accuracy and techniques. Technically, KG-BERT used benchmark datasets such as WN11, FB13, YAGO, our system used Vietnam Tourism KG. Accurately, our study was applied to a question answering system and achieved an accuracy average of 93.34% when compared with the others. The main contributions of our paper are summarized as follows:

- Generating the triples from all meta paths of HIN by scanning Network Schema and using Motif Finding of Apache Spark GraphFrames on large HIN of KG.
- Building a BERT-based triple classification model for triple classification by using the triples generated from all meta paths.
- Building a BERT-based text classification model for the content of triples by converting the generated triples into text and accomplish text classification problems.

2 Related work

BERT [12] and XLNet [36] have great achievements in natural language processing (NLP), these models can learn contextual word embedding with large amounts of textual data. Among them, BERT is the most prominent by pre-training BERT through masked language modeling and next sentence prediction. BERT has been applied in many fields like text classification [4], semantic classification [13], and question answering systems [27].

We surveyed some previous researches that related to our method and summarized them in Table 1. Our solution combined the knowledge graph and BERT model. Specifically, we manipulate meta paths on a heterogeneous information network (HIN) and use BERT for classification.

2.1 Discovering meta paths of HIN

The meta path is an important concept of HIN. Discovering meta-path, path instances of meta-path, and HIN analysis are hot research topics [24]. There have been several studies about automatically discovering the meta-paths of HIN [2, 32, 37]. In [3, 20], the authors proposed an algorithm based on Breadth-First Search to find all the meta-paths of HINs. FSPG utilizes Path Count and Path-Constraint Random Walk (PCWR) [23] to evaluate the similarity between two objects. In [16, 17], the authors proposed algorithms to compute the top-k shortest paths.

In the above studies, the solution of finding the meta-path distance between two vertices is executed with the input HINs in the memory of one computer. These solutions can be restricted when working with a large HIN with millions of vertices and edges like DBLP, YAGO, FreeBase, Vietnam Tourism KG.

2.2 Embedding methods for classification

Wang et.al [35] proposed a method named text enhanced knowledge embedding (TEKE) for knowledge graph representation learning. This method used a knowledge graph and a text corpus as input parameters, and the authors labeled the meanings for entities in the text corpus. A co-occurrence graph was constructed from entities and words to link the knowledge graph with text content. The authors applied a normal translation-based optimization procedure to learn the embedding of the entities and relations.

In contrast, Zhang et.al use rich information from the relational structure of triples instead of an additional text corpus. Zhang et.al proposed the DistMult-HRS that extended from TransE, TransH, and DistMult [34]. The authors defined three tiers including relation clusters, relations, and sub-relations for relation structure. These tiers leverage the rich information to learn knowledge representation. The relation cluster tier is similar to relations that describe the same entities. Relation tier is a predicate of head and tail entities. The sub-relation tier splits a relation that has many semantic meanings into sub-relations. Each triple in a knowledge graph is the sum of three tiers of the DistMult-HRS. The pros of this technique are that it only uses the information of HRS without external text or paths, and the cons are that the authors have to determine relation clusters and sub-relation manually.

Ji et.al proposed a new approach TranSparse to solve the heterogeneity and imbalance of knowledge graph [11].

Heterogeneity is that some relations link many entity pairs (complex relations) and others or not (simple relations). Heterogeneity causes overfitting on simple relations or underfitting on complex relations. The imbalance is that some relations link many head entities and fewer tail entities. Imbalance cannot treat two sides head and tail equally. To process the heterogeneity, the authors used sparse matrices to determine the number of entity pairs and two sides of relations share common transfer matrices. Consequently, the transfer matrices of complex relations are less sparse than simple relations. To overcome the imbalance of relations, the authors separated sparse transfer matrices for each head and tail. The sparse degrees are the number of head entities. The TranSparse model aims to determine a triple correct or not.

The above studies using KG represent text to determine whether the triples are valid or not. This link-based determining has difficulty when working with heterogeneous networks with many different types of entities. Therefore, this leads to the incompleteness of the KG graph. Plus, these studies only analyze links without analyzing content.

Table 1 summarizes the results of state-of-the-art models

Models	State-of-the-art results
FSPG+Greedy Tree [3]	- Detect new meta-paths not provided by an expert is better than using the one provided by experts
	- Improve accuracy of link prediction between 10% and 15%
	- Experiment on DBLP and YAGO dataset
Top-k shortest path join (KPJ) [16]	- Compute the top-k shortest paths from one set of target nodes to another set of target nodes in a graph.
	- Use the best-first paradigm to recursively divide search subspaces into smaller subspaces, and compute the shortest path in each of the subspaces in a prioritized order based on their lower bound.
	- Use an iteratively bounding approach to tightening lower bounds of subspaces to improve efficiency.
	- Proposed index structures to significantly reduce the exploration area of a graph in lower bound testing.
TransH, TransE, TransR [35]	- Take advantage of rich context information in a text corpus to learn a representation of the knowledge graph.
	- Incorporate the semantic structure of the knowledge graph and each relation to solving better 1-to-N, N-to-1, and N-to-N relations
DistMult-HRS ([34])	- extend existing KGE models TransE, TransH, and DistMult to learn knowledge representations by leverage the information from a three-layer hierarchical relation structure (HRS).
TranSparse [11]	- Model knowledge graph by encoding entities and relations into a numerical space.
	- Solve two problems ignore the heterogeneity and imbalance of knowledge graphs.
Fine-tuned BERT for text classification [4]	- Use three steps to fine-tune the pre-trained BERT model for text classification: (1) further pre-trained BERT on within task training data or in-domain data; (2) optional fine-tuning for multi-task learning; (3) fine-tune BERT for the target task
	- Investigate the fine-tuning method for BERT tasks, preprocess of long text, layer selection, layer-wise learning rate, and low-shot learning problems
	- Achieve state-of-the-art results on English and Chinese text classification datasets
KG-BERT for graph completion [15]	- Used pre-trained language model like BERT for graph completion
	- Accessed whether the plausibility of triples exist in a knowledge graph
	- Classified triples into two categories {True, False}
	- Achieved the accuracy triple classification of 91.9%, relation prediction of 96%, and link prediction of 97%.
	- Experimented on WN11, FB13, WN18RR, FB15K, FB15k-237, and UMLS

2.3 BERT for text classification

In the past few years, BERT is a pre-trained language model that gives out state-of-the-art results in text classification, knowledge graph completion, sentiment analysis, so on [18, 31].

For text classification, in [26], the authors compared several models with BERT based model. They conducted experiments to prove the accuracy of using the BERT-based model for text classification. They concluded that the BERT model achieved the top accuracy for text classification. In [4], the author presented how to refine the BERT model for the text classification problem. The author experimented with finding the best way to classify text across eight typed data sets and added a softmax layer to BERT configuration for text classification.

For knowledge graph completion, Liang Yao, et al. used KG-BERT [15] as a BERT and Knowledge Graph model. They converted triple of knowledge graphs to text sequence by using the name/description of entities and relations of the triple. Then they use them for fine-tuned BERT model. The KG-BERT is used to predict the plausibility of the triple. However, with KG-BERT, they only classify the triple as a true or false class. In our system, we want to classify the triple into 3 classes based on whether the triple is base, derived triple or false triple.

BERT shows state-of-the-art results in many fields but its speed is very slow because the model trains too many parameters. Plus, it only accomplishes on text even KG completion task. This restricts the ability of entity relation analysis when the contextual text does not contain the right answer.

To address all the above restrictions of knowledge graph and BERT, we used Apache Spark that was developed in 2009 in UC Berkeley's AMPLab by Matei Zaharia to process huge data [10] [1] [30]. Apache Spark runs on Spark Cluster which contains one master node and several worker nodes. We leverage the large graph processing capabilities of Motif Finding on Apache Spark [33] to find all path instances and triples of a given meta-path of a large HIN network.

In this paper, we will combine linked-based and content-based information of the knowledge graph to learn the representation of triple of the KG. Before our solution, Liang Yao. et al. also combined knowledge graph and BERT for completion but their research was different from ours in two points [15]. Firstly, while they created triples from head and tail entities manually, we used the meta-path to generate new triples. Secondly, Liang Yao used the BERT model to check whether new triples were valid. Meanwhile, we applied BERT to classify triples and their text into multiple labels. We leverage this model for building the triple classification model. Each triple <h, p, t> is converted to text and we use this text and the class label as the input to the BERT-based triple classification model by using a similar model of the BERT-based text classification model [4]. After training the triple classification model can classify the triple <h, p, t> to one of 3 classes. We combine linked-based (meta-path) and content-based (triple description) information into triple representation for building a triple classification model. BERT model is used to generate the contextual embedding vector of triples generated from the knowledge graph. The contextual embedding vectors are used for the triple classification problem.

The rest of the paper is organized as follows: we define crucial concepts and describe components to accomplish our proposed solution in section 3, we conduct experiments to prove the performance of our study in section 4, we conclude our study in section 5, and we suggest issues that need to be improved for future work in section 6.

3 Methodology

In this section, we present how to build a BERT-based triple classification model to support a question answering system by combining linked-based and content-based information of Knowledge Graph.

3.1 Problem definition

In the KG, all entities have various types that are considered heterogeneous information networks (HIN) [5] [7].

Definition 1 (information network) An information network is defined as a directed graph G = (V, E) with an object type mapping function $\varphi: V \rightarrow A$ and a link type mapping function $\psi: E \rightarrow R$. Each object $v \in V$ belongs to one particular object type set A: $\varphi(v) \in A$, and each link $e \in E$ belongs to a particular relation type in the relation type set R: $\psi(e) \in R$. The information network is called heterogeneous information network if the types of objects $|A| > 1$ or the types of relations $|R| > 1$; otherwise, it is a homogeneous information network [5].

Definition 2 (network schema) The network schema of a HIN denoted as TG = (A, R), is a meta template for a HIN G = (V, E) with the object type mapping $\varphi: V \rightarrow A$ and the link mapping $\psi: E \rightarrow R$, which is a directed graph defined over object types A, with arc as relations from R [5].

We build up a Vietnam Tourism KG. We consider this knowledge graph as a HIN. The network schema of this HIN is shown in Fig. 1. In this HIN, we have a set of entities as {Beautiful_Sight, Ethnic_Group, Province, Dish, National_Hero, Folk_Song, Traditional_Music, Festival, Region, ...} and a set of relations as {Organized_At, Located_At, Folk_Song_Of, Born_In, Specialty_Dish, ...}

Definition 3 (Meta-path) A meta-path P is a path defined on the network schema TG = (A, R) of a given HIN, and is denoted in the form of $A_1 \rightarrow^{R1} A_2 \rightarrow^{R2} \ldots \rightarrow^{R_n} A_n$, which defines a composite relation $R = R1 \circ R2 \circ \ldots \circ R_n$ between type A_1 and A_n, where ° denotes the composition operator on relations [29, 32].

Thus, a direct or indirect relation will be determined by a meta-path, where the direct relation has the same name as the name of the link in the meta-path, and the indirect relation has a name created by concatenating of links in a meta-path.

Some meta-paths of Vietnam Tourism KG are shown in Table 2:

Definition 4 (path-instance) A path instance of a specific meta-path $A_1 \rightarrow^{R1} A_2 \rightarrow^{R2} \ldots \rightarrow^{R_n} A_n$ is a path p of HIN where p: $A_1 \rightarrow^{R1} A_2 \rightarrow^{R2} \ldots \rightarrow^{R_n} A_n$ and Ai is a vertex of HIN, $A_i \rightarrow^{Rj} A_j$ is an edge with source vertex Ai and target vertex Aj of HIN.

For example, with meta-path (Beautiful_Sight)-[located_at] → (Province), a path instance of this meta-path is ("Ngu Binh Mountain") - [located_at] → ("Hue"). This path instance creates a triple as ("Ngu Binh

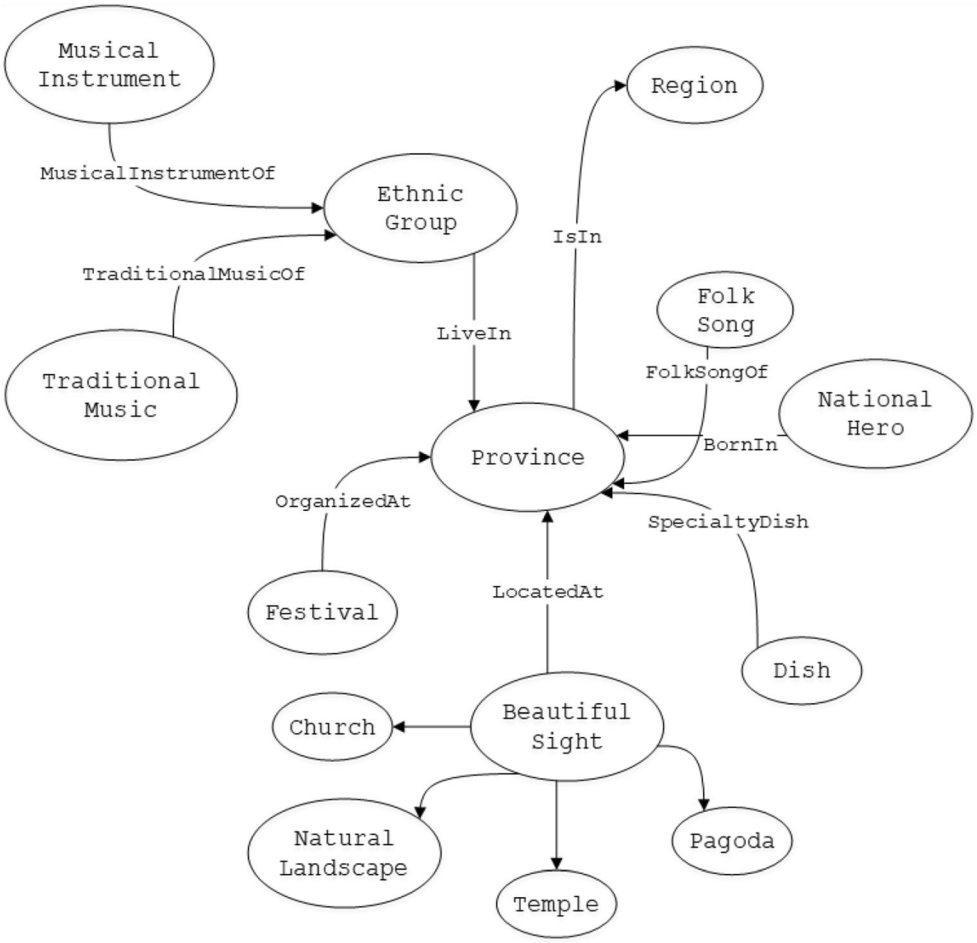

Fig. 1 Network schema of Vietnam Tourism KG

Mountain",located_at,"Hue"). In this entity, "Ngu Binh Mountain" is a head entity, "located_at" is a predicate, and "Hue" is a tail entity.

Some path instances of meta-path Beautiful_Sight $\rightarrow^{\text{LocatedAt}}$ Province is shown in Table 3.

Definition 5 (triple and path instance) The triple of the knowledge graph is a triple of <h, p, t > where h is the first entity, t is the last entity, p is the relation between entity h and entity t. Relation p can be a direct relation or indirect relation. Each triple matches a meta path [5].

3.2 Approach

Our proposed solution had four components which were shown in Fig. 2. First, we searched all the existing meta-paths of a given HIN. Next, we discovered all path instances of all meta paths of large HIN using Motif Search of Apache

Table 2 Some meta-path and relation label in Vietnam Tourism KG

#	Meta-path	Relation label	Relation type
1	Beautiful_Sight→Province	Located_At	Directed
2	National_Hero→Province	Born_In	Directed
3	Dish→Province	Specialty_Dish	Directed
4	Folk_Song→Province	Folk_Song_of_Province	Directed
5	Festival→Province	Organized_at_Province	Directed
6	Beautiful_Sight→Province→Region	Beautiful_Sight_of Region	Un-directed
7	Dish→Province→Region	Specialty_Dish_of Region	Un-directed
8	Folk_Song→Province→Region	Folk_Song_of_Region	Un-directed

Table 3 Some path instances of a meta-path Beautiful_Sight $\rightarrow^{\text{LocatedAt}}$ Province

#	Path instance
1	(Ngu Binh Mountain) $\rightarrow^{\text{LocatedAt}}$ (Hue)
2	(Ngu Hanh Mountain) $\rightarrow^{\text{LocatedAt}}$ (Da Nang)
3	(Hoan Kiem Lake) $\rightarrow^{\text{LocatedAt}}$ (Ha Noi)
4	(Ha Long Bay) $\rightarrow^{\text{LocatedAt}}$ (Quang Ninh)

Spark and generate all triples from these path instances. This is the linked information of triples. Then, we built a BERT-based triple classification model that was trained by all created triples of HIN. Each triple <h, p, t > is changed to text by using the description of h entity, tail entity, and predicate. This is the content of triples. Finally, we used the BERT-based triple classification model to classify triple into a set of three classes. The text of triple is represented by the contextual embedding vector of the BERT model. Each component was presented in detail as follows.

3.2.1 Discovering all meta-paths between two vertices of network schema

This task is simple on the small network schema of HIN. Given any two vertices of this graph, we use Breadth-First Search [3] to find all meta-paths between the two vertices of the network schema. These two vertices are the two types of entities in KG. Table 3 is a list of all asymmetric meta paths in Vietnam Tourism KG.

3.2.2 Find all path instances of a meta-path using motif finding of apache spark

Motif finding refers to the search for structural patterns in a graph. Motif Finding is a function of GraphFrames [30]. Motif Finding [1] runs in a distributed environment and can work with large-scale KG. Motif Finding uses Domain-Specific Language (DSL) for expressing structural queries. We use Motif finding to find the path instances of meta-path.

Table 4 illustrates some meta paths and the Motif Finding function of GraphFrames to find the path instances of meta path. In Table 4, graph g is the GraphFrames structure to express the HIN.

3.2.3 Finding the triples of path instances

From created path-instances, our system generated triples and considered them as the link-based information from the structure of KG.

3.2.4 BERT-based triple classification model

This model is trained on all created triples using path instances. Each triple <h, p, t > is changed to text by using the description of h entity, tail entity, and predicate. This is the content of triples. This text is assigned a text label depending on the class of triple in a predefined class as {based triple, derived triple, non-existent triple}. Then the set of texts generated from the triples and triple labels will be used to train a BERT-based text classification model. We combine linked-based and content-based information for triple representation.

We develop a BERT-based model for triple classification. For triple <h, p, t>, the head entity is represented as a sentence containing words. With the head entity "Ngu Binh Mountain". This head entity is represented by the text "Ngu Binh Mountain is the mountain in Hue. Ngu Binh Mountain is located 30 km from central of Hue". The tail entity is also represented by a text. For example, the tail entity of "Hue" is represented by the sentence "Hue is the ancient capital of Vietnam. There are many beautiful sights in Hue". The predicate of a relation is also described by a text, for example, "Located_At" is described by a phrase as "is located at". The text of this triple is "Ngu Binh is located in Hue. Ngu Binh Mountain is the mountain in Hue. Ngu Binh Mountain has located 30 km from the center of Hue. Hue is the ancient capital of Vietnam. There are many beautiful sights in Hue".

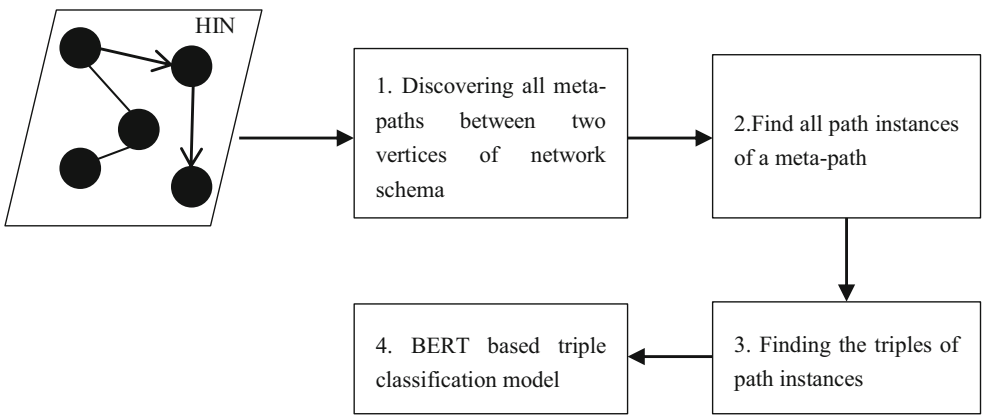

Fig. 2 Four components of solving the problem

Table 4 List of some asymmetric meta-path in Vietnam Tourism KG and Meta-path and Motif Finding Command of Apache Spark GraphFrames

Source vertex	Tail vertex	Meta-path	Motif Finding
Musical Instrument	Ethnic Group	Musical_Instrument→Ethnic_Group	g.find("(a)-[e]->(b)") .filter("a.type='Musical_Instrument' and b.type='Ethnic_Group'"
Musical Instrument	Province	Musical_Instrument →Ethnic_Group→Province	g.find("(a)-[e1]->(b); (b)-[e2]->(c)") .filter("a.type='Musical_Instrument' and b.type='Ethnic_Group' and c.type=' Province'"
Folk Song	Region	Folk_Song→Province→Region	g.find("(a)-[e1]->(b); (b)-[e2]->(c)") .filter("a.type='Folk_Song' and b.type='Province' and c.type=' Region'"
Beautiful Sight	Beautiful Sight	Beautiful_Sight→Province← Beautiful_Sight	g.find("(a)-[e1]->(b); (c)-[e2]->(b);") .filter("a.type=' Beautiful_Sight' and b.type=' Province'and and c.type=' Beautiful_Sight'"

This text will be fed into BERT model architecture which is a multi-layer bidirectional Transformer encoder based on the original implementation described in [12]. The important limitation of BERT is the maximum length of the text is 512 tokens. For shorter description text than 512, we need to add pad tokens [PAD]. On the other hand, if the description text is longer, we need to cut the description text. We consider the triple classification model as the text classification model. We use text and its label to train the model and we call this model a BERT-based triple classification model. These texts may be considered as the content-based information of triples. We combine link-based and content-based information for triple representation in the triple classification model.

With the triple <Ngu Binh Mountain, located_at, Hue>, then we convert this triple to the natural language text as" Ngu Binh Mountain is a mountain in Hue. Ngu Binh Mountain has located 30 km from the center of Hue. Ngu Binh Mountain is located at Hue. Hue is the ancient city of Vietnam. There are many beautiful sights in Hue ". We then assign the "based triple" label to this text.

We also create the invalid triple such as <Ngu Binh Mountain, located_at, Da Nang> then we convert this triple to the natural language text as" Ngu Binh Mountain is a mountain in Hue. Ngu Binh Mountain has located 30 km from the center of Hue. Ngu Binh Mountain is located at Da Nang. Danang is a coastal city in central Vietnam. Da Nang is the commercial and educational center of Central Vietnam". We assign the "non-existent triple" label to this text.

Besides of "non-existent triple" label, we also need to know whether this triple is based triple or derived triple. We need to classify the triple into 3 classes as shown in Table 10.

3.2.5 BERT model for text classification

We using the BERT-based triple classification model to classify triple into a set of three classes. The text of triple is represented by the contextual embedding vector of the BERT model.

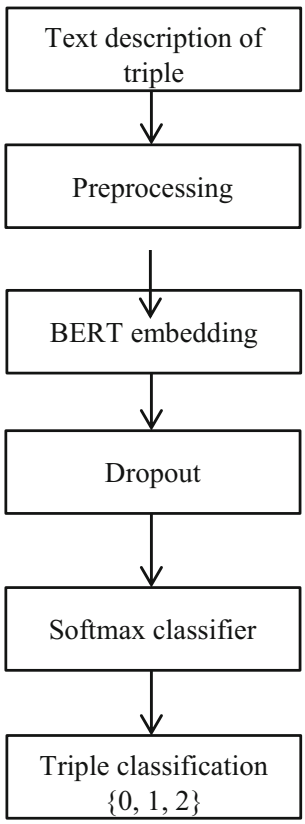

Fig. 3 Triple classification

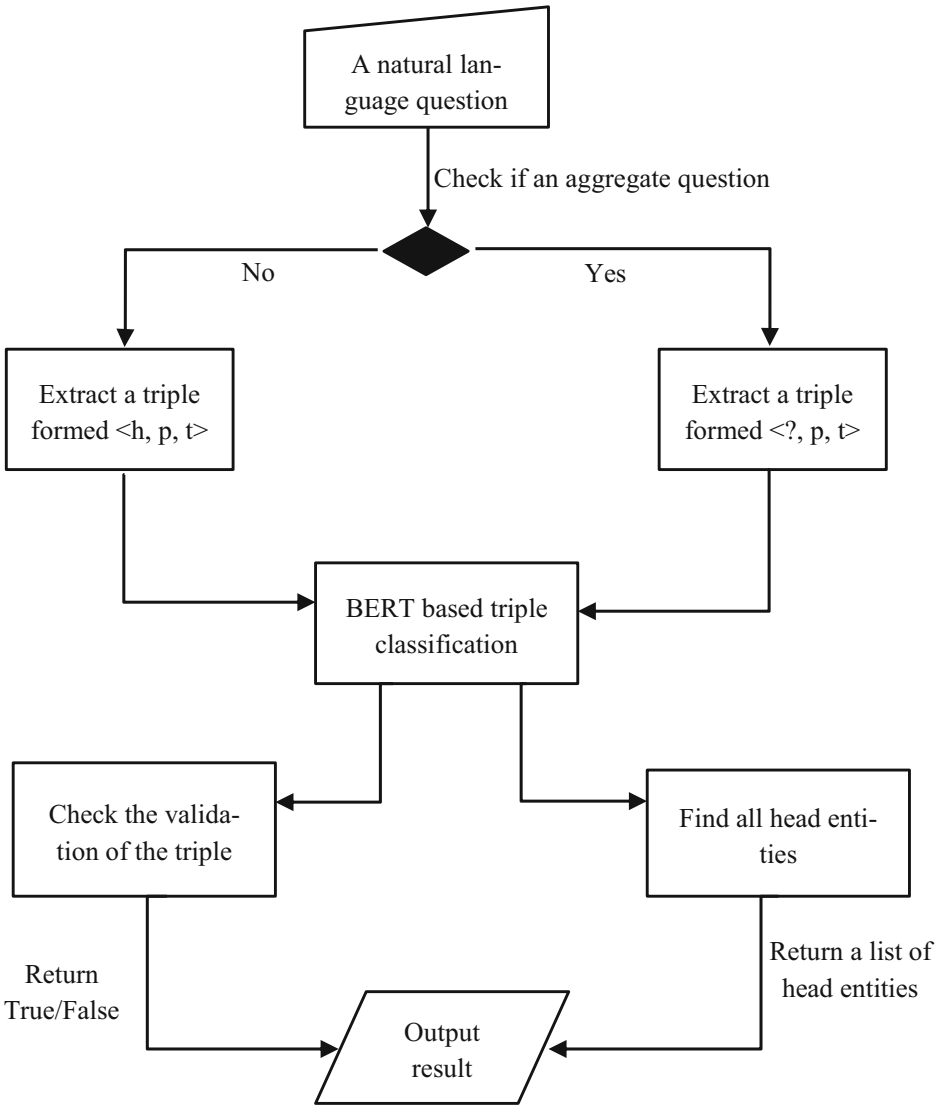

Fig. 4 The process of finding answers for Yes/No and aggregate input questions

In [4], the authors developed a BERT based text classification model, a Softmax Classifier is added to the top of BERT architecture to predict the probability of label c using the following formula:

$$p\,(c|h) = \text{softmax}\,(W_h) \quad (1)$$

Where W_h is the task-specific parameter matrix. The authors used a BERT-based model contains an encoder with 12 Transformer blocks, 12 self-attention heads, and a hidden size of 768. The authors fine-tune all the parameters of the BERT model and W jointly by maximizing the log-probability of the correct label. We leverage this model for our BERT-based triple classification model. The embedding output of the triple description and the triple label will then be fed into the dropout block and Softmax Classifier for triple classification as shown in Fig. 3. The text descriptions of entities are selected carefully from Wikipedia and our knowledge about Vietnam tourism.

Table 5 Question and triple validity

Natural language question	Corresponding Triple
Is Ngu Binh Mountain located at Hue?	<Ngu Binh Mountain, located at, Hue>
Is Chau Van Song the Folk Song of Hue?	<Chau Van Song, Folk Song Of, Hue>
Is Sesame the specialty food of Hue?	<Sesame, SpecialtyOf, Hue>
Is Vong Canh Hill is in the central region of Vietnam?	<Vong Canh Hill, Is In, The Central region of Vietam>

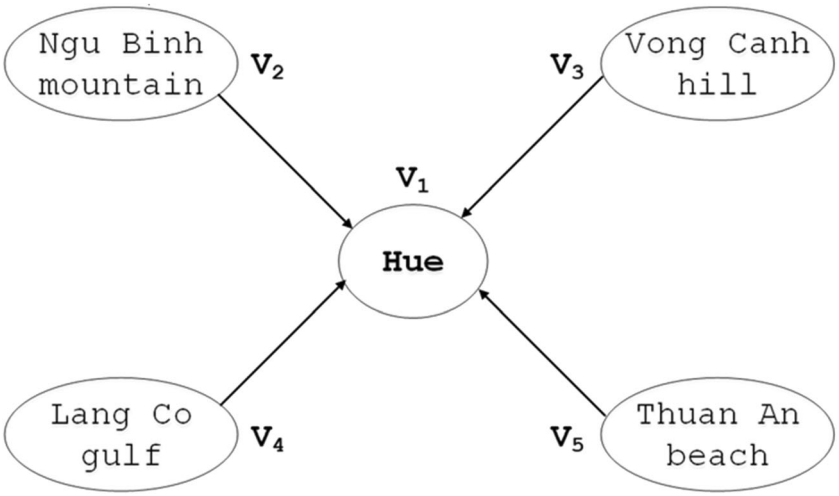

Fig. 5 The answer to an aggregate question

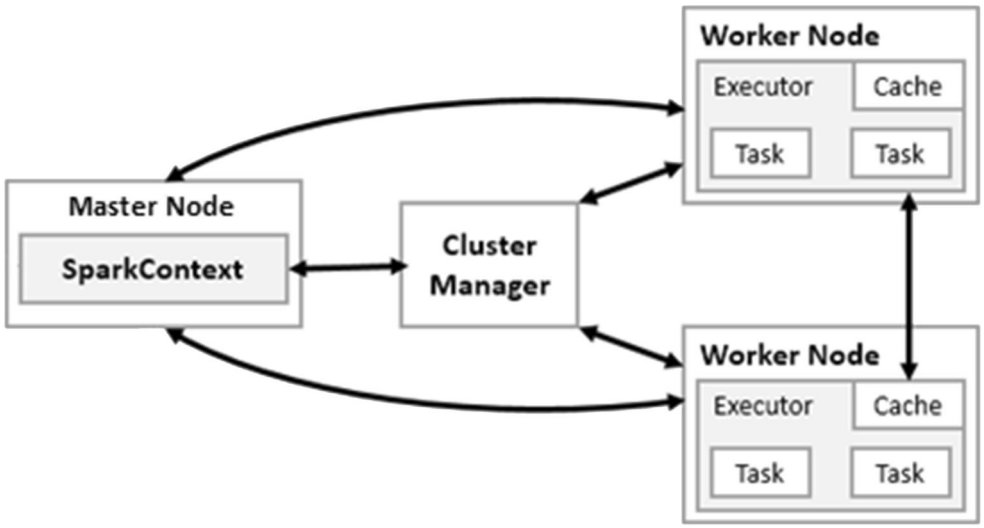

Fig. 6 The architecture of our Spark Cluster

3.3 Enhancing the question answering system using the BERT based triple classification

In Fig. 4, we presented how we applied the BERT-based triple classification to enhance a question answering system.

To answer the questions in Table 5, we need to classify the triple created from the appropriate question. For example, with the question "Is Ngu Binh Mountain located at Hue?", we need to classify the triple <Ngu Binh Mountain, located at, Hue>. If this triple is a non-existent triple, the answer to this question will be "not correct". If this triple is a based triple or derived triple, the answer to this question will be "correct".

To answer the aggregate question such as "What are the beautiful sights located at Hue?". This question generates a triple <h,p,t > where p is "located_at", t is "Hue". We need to find all head entities h in a set of entities of KG, such that triple <h,p,t > is a based triple. The BERT-based Triple classification can be used to classify the triple <h,p,t > .

Table 6 Software is installed in Spark Cluster

Software	Version
Operating System	Ubuntu 18.04
Java	OpenJDK version 1.8.0_222
Scala	Version 2.11.12
Apache Hadoop	Apache Hadoop 2.8.5
Apache Spark	Apache Spark 2.4.4

Table 7 Statistical datasets were used in our experiment

Dataset	#Relation	#Entity	#Edges
WN11	11	38,696	112,581
FB13	13	75,043	316,232
VNTKG	120	4,00,000	5,432,678

Table 8 Entity and the name/description of Entities

Entities	Entity name/description
Ha noi	Ha noi is the capital of Vietnam.Hanoi is the commercial, cultural, and educational centre of Northern Vietnam.
Hue	Hue is the ancient city of Vietnam. There are many beautiful sights in Hue
Da Nang	Danang is a coastal city in central Vietnam, Da Nang is the commercial and educational center of Central Vietnam
Quang Ngai	Quang Ngai is a city in central Vietnam. Tra River is a natural landscape of Quang Ngai
West Lake	West Lake is the biggest freshwater lake of Hanoi, Vietnam, located northwest of the city center.
Ngu Binh Mountain	Ngu Binh mountain is a mountain in Hue. Ngu Binh Mountain is located 30 km from the central of Hue
Ngu Hanh Mountain	Ngu hanh mountain is a mabre mountain in Da Nang. All of the mountains have cave entrances and numerous tunnels

The answer to an aggregate question is a set of elements such as {Ngu Binh Mountain, Vong Canh Hill, Lang Co Gulf, Thuan An Beach} as shown in Fig. 5.

Algorithm 3 is used to collect the facts of KG for answering the aggregate question as follows:

```
Algorithm 3: Find the good head entity in the set of possible entities of KG

Input:tail entity t, predicate p, Set E of possible entities of KG

Output: top K of plausible entities with tail entity t and predicate p

1.  Function FindTailEntity
2.      Answer=[]
3.      for each h in E do
4.          Prob, Class = BERT_Triple_Classification(h, p, t)
5.          if Class==True then
6.              Answer.append(h,Prob)
7.          Endif
8.      Endfor
9.      return Answer
10. EndFunction
```

Function BERT_Triple_Classification(h, p, t) in line 4 is used to classify and calculate the probability of triple <h, p, t > where h is in E of possible entities of KG.

4 Experiment and discussion

4.1 Generating the path instances of meta-path

4.1.1 Spark cluster environment

To conduct experiments of Motif Finding runs on Spark Cluster as shown in Fig. 1, we built a Spark Cluster on the cloud computing system of our university. Our Spark Cluster includes 9 virtual computers. Where, one virtual computer works as both master node and worker node (For simplicity, we refer to this computer as the master node), and eight virtual computers work as worker node only (Similarly, we refer to these computers as the worker nodes). The configuration of the master node as the followings:

- 8 CPUs: Intel(R) Xeon(R) CPU E5–2660 v4 @ 2.00GHz
- Installed memory (RAM): 16.0 GB
- And the configuration of the worker nodes as the followings:
- 4 CPUs: Intel(R) Xeon(R) CPU E5–2660 v4 @ 2.00GHz
- Installed memory (RAM): 8.0 GB

The architecture of our Spark Cluster is shown in Fig. 6. Where:

- Master Node: is a computer running main programs, sending code to the worker nodes to execute in parallel, and collecting the results.
- Worker Node: is a computer participating in processing requests of the master node.
- Cluster Manager: is a component allocating resources across applications.

The software installed on our Spark Cluster as shown in Table 6.

Table 9 Relations and relation names/descriptions

Predicate/Relation	Predicate Relation description
Specialty_Dish	Is the specialty dish of
Born_in	Was born in
Located_At	Is located at
Is_in	Is in
Has_type	Has type
OrganizedAt	Is organized at
Is_Beautiful_Sight_of	Is beautiful sight of

Table 10 Triples of training set

Head entity	Predicate/Relation	Tail Entity
Sesame	Specialty_Dish	Hue
Quang Trung	Born_in	Binh_Dinh
West Lake	Is_Beautiful_Sight_of	Ha Noi
Ngu Binh Mountain	Is_Beautiful_Sight_of	Hue
Vong canh Hill	Is_Beautiful_Sight_of	Hue
Thuan An Beach	Is_Beautiful_Sight_of	Hue
Lang Co Gulf	Is_Beautiful_Sight_of	Hue
Tu Dam Pagoda	Is_Beautiful_Sight_of	Hue

Motif Finding runs on this Spark Cluster for generating the path instances of the meta-paths.

4.1.2 Data set

We experimented on WordNet11 [22, 28], FB13 [25], and a subset of the Vietnam Tourism KG network with 4 million vertices and 5,432,678 edges is used for testing our proposed system. WordNet ([8, 21] is an online lexical database that organizes English nouns, verbs, adjectives, and adverbs into sets of synonyms. WordNet11 includes 11 relations, 38,696 entities, 112,581 edges. FreeBase [14] is a practical graph database for structuring human knowledge graphs. FreeBase contains approximately 125,000,000 tuples, 4000 types, and 7000 properties. However, in this paper, we only extract 13 relations, 75,043 entities, 316,232 links.

Besides the two above benchmark datasets, we also create our dataset about the Vietnam tourism Knowledge graph (VNTKG). This data set is saved in a .csv file and moved to HDFS for processing. Table 7 shows the statistical datasets used in our experiment.

The names/descriptions of entities are shown in Table 8. The names/descriptions of predicates/relations are shown in Table 9. Some triples of the training data set are shown in Table 10. Some triples of the test data set with labels are shown in Tables 11 and 12.

For measuring the performance analysis of Motif Finding in a local and distributed environment, we translate our Knowledge graph from Vietnamese to English to have a large knowledge graph with 240 edge labels (120 edge labels in Vietnamese and 120 edge labels in English), 8,000,000 vertices and 10,865,356 edges. We believe that this large knowledge graph will help us to prove the performance of our proposed method. Moreover, we have a bilingual knowledge graph that we can use two languages for querying.

4.1.3 The performance of motif finding on local and distributed environment

We experiment to measure the execution time of finding all path instances of meta-path between 2 vertices of large HIN on the local environment and Spark Cluster by using algorithm Motif Finding algorithm to find the path instances of meta-path (Beautiful_Sight) $\rightarrow^{\text{Located\_At}}$ (Province) $\rightarrow^{\text{Is\_In}}$ (Region) on an Apache Spark Framework with 1, 2, 4, 6, 8 worker nodes. Figure 7 illustrates the comparison of the execution time of the Motif Finding function on the local environment and Spark Cluster with a different number of workers.

Although the executive time of Motif Finding on the distributed environment does not improve better than this on the local environment because of the small used dataset. However, this experiment proves the performance of the Motif Finding Function on distributed computing on Spark Cluster for processing large-scale graphs from 2 worker nodes to 8 worker nodes. Yet, the distributed environment also took us much computing and communication costs.

4.2 BERT-based triple classification model vs embedding methods

We collected about 537,699 triples about Vietnam tourism. This dataset consists of three parts with 322,621 (60%) for training, 107,539 (20%) for validation, and 107,539 (20%) for testing. Each component has two classes {true, false}, where true represents a triple existing in the description and false indicates a triple that does not appear in the description. The triples collected from the Motif Finding algorithm always give true value, so to make the false triple, we did the

Table 11 Triples of test set

Head entity	Predicate/Relation	Tail Entity	Class
Don	Specialty_Dish	Hue	True
Don	Specialty_Dish	Quang_Ngai	False
Tra River	Is_Beautiful_Sight_of	Quang Ngai	True
Tra River	Is_Beautiful_Sight_of	Quang Nam	False
One column Pagoda	Is_Beautiful_Sight_of	Ha_noi	True
One column Pagoda	Is_Beautiful_Sight_of	Hue	False

Table 12 The label of triple classifier

Triple	Text	Class
<Ngu Binh Mountain, located_at, Hue>,	Ngu Binh Mountain is a mountain in Hue. Ngu Binh Mountain is located at Hue. Hue is the ancient city of Vietnam	Based Triple
<Ngu Binh Mountain, is_in TheCentralRegion>,	Ngu Binh Mountain is a mountain in Hue. Ngu Binh Mountain is located at Hue. Hue is the ancient city of Vietnam. Hue is in the Central Region of Vietnam	Derived Triple
<Ngu Binh Mountain, is_in, TheNorthRegion>,	Ngu Binh Mountain is a mountain in Hue. Ngu Binh Mountain is located at Hue. Hue is the ancient city of Vietnam. Hue is in the North Region of Vietnam	Non-existent Triple

following two ways: head replacement, and predicate replacement from true triples.

For head replacement, given $c \in C$, where $c = (h, p, t)$ is a valid triple in the triple C set. We randomly select a head entity $h' \in E$, where E is the set head and tail of the knowledge graph, to form a triple $c' = (h', p, t)$ such that $h' \neq (h, t)$, $c' \notin C$ and assign $c' = false$. We added the condition $c' \notin C$ because replacing h with h' would result in $c' = (h', p, t) \in C$ and make the value of c' from true becomes false.

With a predicate substitution, given $c \in C$, where $c = (h, p, t)$ is a valid triple in the triple C set. We also randomly selected a predicate $p' \in P$, where P is the predicate set of the knowledge graph, to form a triple $c' = (h, p', t)$ such that $p' \neq p$, $c' \notin C$ and assign $c' = false$.

For each data set, we balance the sample number of triple labeled true and false. The BERT-based classification model was trained on the VN Tourism KG dataset with the following parameters: we use the pre-trained model is BERT-based uncased, the maximum triple length is 200, the training size is 32, the learning rate is 5e-5, the number of epochs is 10. Besides, we also set the Adam algorithm and cross-entropy to measure the effectiveness of our training model [6]. The Adam algorithm is a learning rate optimization algorithm widely used in training deep neural networks. Cross entropy is used to measure the performance of the model by calculating the similarity between the probability of the predictive model and the observed target. For the classification problem, we compared the probability of the true class with the probabilities of each class in the model. The smaller the value of cross-entropy, the more accurate the predictive model is. We are interested in two measures when performing the training of the BERT classification model: accuracy and error rate. In Fig. 8, the accuracy of the training and validation set is nearly equal starting from epoch 8. In Fig. 9, the validation difference rate is always stable below 0.1. The evaluating results of the triple classification process were shown in Table 13.

We collected about 10,865,356 triples about the Vietnam tourism Knowledge graph(VNTKG). This dataset consists of three parts 60% for training, 20% for validation, and 20% for testing. Each component has two classes {true, false}, where true represents a triple existing in the description and false indicates a triple that does not appear in the description. The triples collected from the Motif Finding algorithm always give true value, so to make the false triple, we did the following two ways: head replacement, and predicate replacement from true triples.

We conducted experiments to compare our triple classification model with different embedding methods. We used F-score or F1 to evaluate the accuracy of our BERT-based triple classification model on three training data sets, Vietnam Tourism KG, WN11, and FB13 as shown in Table 14 [19]. Let TP as triples that our method predicted based/derived

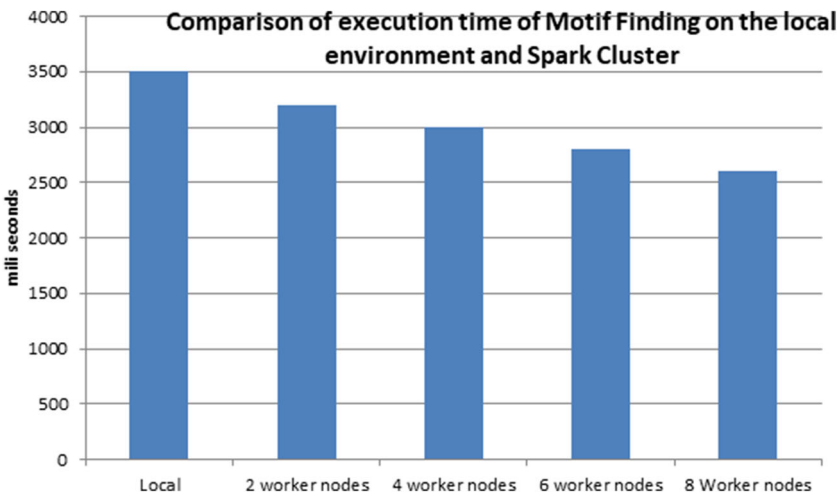

Fig. 7 Comparison of the execution time of Motif Finding on the local environment and Spark Cluster

Fig. 8 The comparison of accuracy between training dataset and validation dataset on Vietnam tourism KG

classes correctly, FP as triples that our method predicted based/derived classes incorrectly, and FN as triples that our method predicted non-existed classes incorrectly. We established the F-score formula as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision*Recall}{Precision + Recall} \tag{4}$$

In Table 14, we compare our solution with other methods introduced in the Related Work section. On FB13, BERT triple classification model outperforms all the baseline methods with an accuracy of 93.2%. On WN11 and Vietnam tourism KG, the accuracy of our model and KG-BERT are equal with 93.6% and 91.3% respectively. Our model still wins the remaining methods.

Table 14 proves the performance of our triple classification model in triple classification. The accuracy of KG-BERT and our Triple Classification Model are fairly equal because two models are developed on the same BERT platform. However, our model is the multiple class triple classifier.

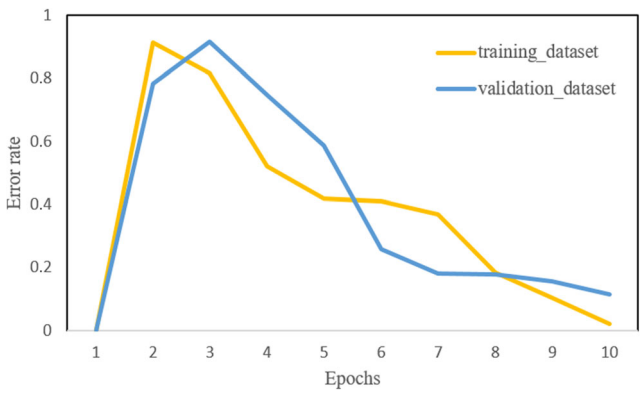

Fig. 9 The comparison of the error rate between the training dataset and validation dataset on Vietnam tourism KG

Table 13 Triple classification

Triple index	Base or derived class	Non-existent class
1	0.45330151	0.17804337
2	0.45283037	0.19121797
3	0.40624392	0.20434187
4	0.42762902	0.18537137
5	0.45153528	0.21761996
6	0.45730424	0.21190324
7	0.41883537	0.23959357
8	0.43212821	0.20390081
9	0.39705953	0.21063472
10	0.42649937	0.20202771

The accuracy of our BERT based triple classification model is also shown in the fact that the number of triples used for training accounts for only 5%, 10%, 15%, 20%, and 30% of the triples of several KG such as Vietnam Tourism KG, WN11, and FB13 data sets, the accuracy is also higher as shown in Fig. 10. The accuracy of the BERT-based Triple Classification Model is not high compared with WN11 and FB15 because we get the text of entity description and relation from Wikipedia. In the future, we will edit these descriptions carefully by a human expert to enhance the accuracy of our model.

4.3 Comparing the executive time of using motif finding with BERT based triple classification model

We implemented another solution using CNN to compare with our BERT-based method. We use CNN to identify the meta-path for each question. For example. Ngu Binh mountain is located in any region. This question will be put into the neural network and the corresponding meta-path detection is Beautiful_Sight $\rightarrow^{\text{LocatedAt}}$ Province$\rightarrow^{\text{IsIn}}$ Region. After extracting all the meta-paths from the knowledge graph, we compiled all the questions for each meta-path and trained the CNN network to identify the meta-paths for the questions. Later, when we encounter a question in natural language, we use CNN to find the meta-path corresponding to the question. Next, we use Motif Finding to find the tuple that satisfies the meta-path. From the triple, we have the fact to answer the question. This solution is time-consuming to identify the meta-path and the time to find triple is based on Spark's Motif Finding. So it is slower than solution 2 using BERT as suggested above.

We made a comparison and got the results shown in Fig. 11 below.

Table 14 The F1-score accurate comparison results of triple classification task (in percentage) between different embedding methods

Method	TourismKG	WN11	FB13	Average
TEKE (Wang and Li 2016)	83.4	86.1	84.2	84.57
TranSparse-S (Ji et al. 2016)	83.5	86.4	88.2	86.03
DistMult (Zhang et al. 2018)	84.3	87.1	86.2	85.87
DistMult-HRS (Zhang et al. 2018)	86.7	88.9	89.1	88.23
AATE (An et al. 2018)	87.1	88.2	87.6	87.63
KG-BERT (2019)	91.3	93.5	92.9	92.23
BERT Triple Classification Model (2020)	91.3	93.6	93.2	92.34

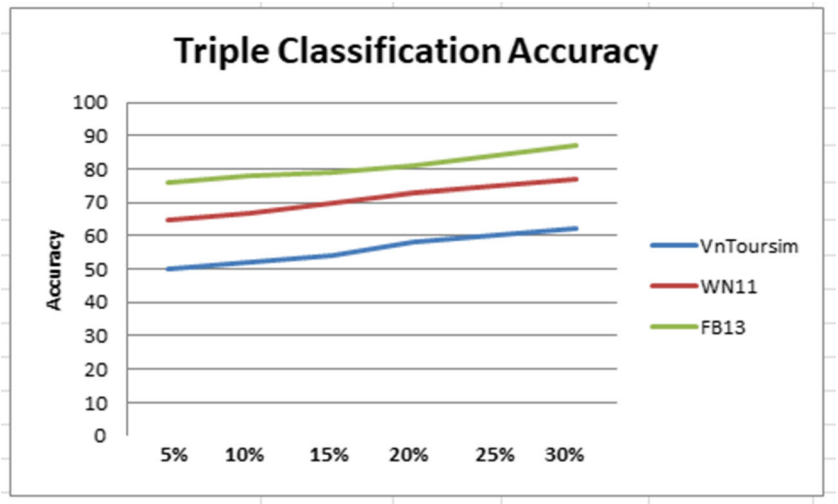

Fig. 10 Test Classification accuracy of our BERT based Triple Classification Model

From the result, as shown in Fig. 10, we hold that a BERT based triple classification model provides a faster response time compared with the solution that uses the Motif Finding algorithm to create a path instance for meta-path determined by the predicate of triple and then creates a triple of these path instances. Especially, our solution achieved most effective when we worked with large-scale HIN and very long meta-path for the derived triple.

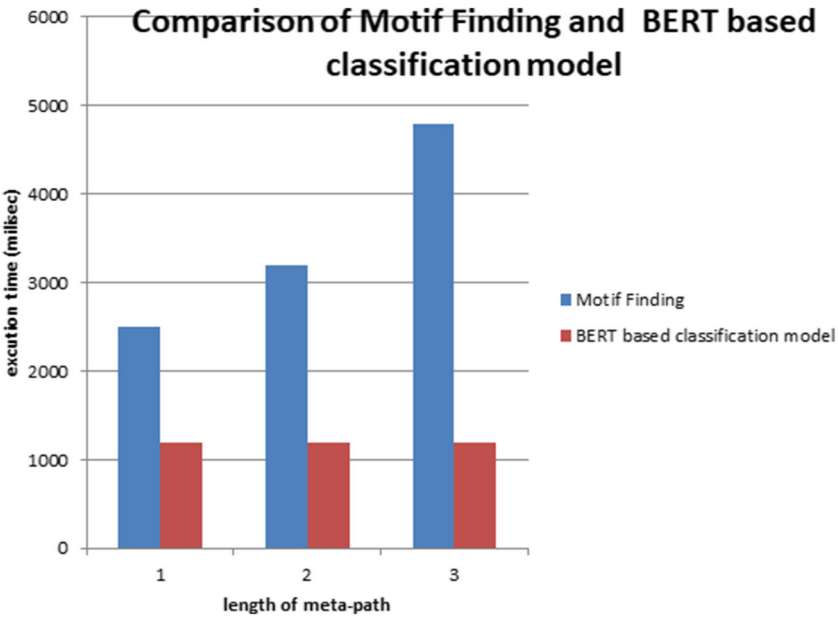

Fig. 11 The comparison of Motif Finding for one meta-path and BERT based triple classification

5 Conclusion

In this paper, we develop a system for triple classification by enhancing the question answering system. The KG contains triples of the form <h, p, t > where h, t are the head and tail entities, p is the predicate indicating the relation between the head and tail entities. We build a large Vietnam Tourism KG with 4 million entities and 6 million links between entities. We consider this KG as a HIN. In HIN, meta-path plays an important role. We use Breadth-First Search to discover all meta-paths of HIN by using network schema. We use the Motif Finding Function of Apache Spark GraphFrames to generate path instances of meta-paths of a large-scale HIN. From these meta-paths, we generate the based triples and derived triples from these path instances. This is the linked information of triple. Each triple <h, p, t > will be converted to text that describes the triple. This text is the description of the entity and predicate of the triple. This is the content base information of triple. We use text-generated form triples to train our BERT-based triple classification model. We use this BERT-based model for triple classification and use triple classification to enhance the capability of the question & answer system. We experimented with Motif Search of Apache Spark GraphFrames and our BERT-based Triple classification model with several KGs such as FP13, WN11, and Vietnam Tourism KG to prove the performance of our proposed algorithm.

6 Limitation and future work

Our proposed solution only reaches the correctness and completeness of the knowledge graph, but we have not regarded heterogeneity and imbalance of relations. Also, we have not applied ontology to make mixed predicates of triples from path instances semantically. In the future, we will improve our method to solve the above limitations on the distributed environment.

Acknowledgments This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCMC) under grant number DS2020-26-01.

References

1. Ankur, D, AJ (2016). GraphFrames: an integrated API for mixing graph and relational Queries. GRADES 2016, June 24 2016, Redwood Shores, CA, USA
2. Binbin Hu, C. S. (2018). Leveraging meta-path based context for Top-N recommendation with a neural co-attention mode KDD 2018
3. Changping, Meng, R. C. (2015). Discovering Meta-paths in large heterogeneous information networks. WWW 2015. Florence, Italy
4. Chi, Sun, X. Q. (2020). How to fine-tune BERT for text classification. Retrieved 2 12, 2020, from arXiv:1905.05583v3 [cs.CL] 5 Feb 2020
5. Chuan S, Y. L. (2017). A survey of heterogeneous information network analysis. IEEE Trans Knowl Data Eng, 29(1), 17–37
6. Diederik P, Kingma JL (2015). ADAM: a method for stochastic optimization. ICLR
7. Do, P. (2019). A System for Natural Language Interaction With the Heterogeneous Information Network . In B. B. Gupta, Handbook of Research on Cloud Computing and Big Data Applications in IoT (pp. 271–301). IGI Global Publishing
8. Fellbaum C (1998) WordNet: an electronic lexical database. MIT Press, Cambridge, MA
9. Galkin M (2020) Knowledge graphs in natural language processing @ ACL:2020 Retrieved June 30, 2020, from https://towardsdatascience.com/knowledge-graphs-in-natural-language-processing-acl-2020-ebb1f0a6e0b1
10. Guller M (2015) Big data analytics with spark. Apress
11. Guoliang Ji, K. L. (2016). Knowledge graph completion with adaptive sparse transfer matrix. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16) (pp. 985-991). Association for the Advancement of artificial
12. Jacob Devlin, M.-WC (2018). BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
13. Jianfei Yu J (2019). Adapting BERT for target-oriented multimodal sentiment classification. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence
14. Kurt Bollacker CE (2008). Freebase: a collaboratively created graph database for structuring human knowledge. Proceedings of the 2008 ACM SIGMOD international conference on Management of data, SIGMOD
15. Liang Yao CM (2019) KG-BERT: BERT for knowledge graph completion. arXiv:190903193v2 [csCL] (11 Sep 2019)
16. Lijun Chang XL (2015) Efficiently computing top-K shortest path join. In: 18th international conference on extending database technology (EDBT). Belgium, Brussels
17. Liu H, Cheqing J (2018) Finding top-k shortest paths withdiversity. TKDE 30(3):488–502
18. Manish Munikar SS (2020, 3 15). Fine-grained sentiment classification using BERT. retrieved from arXiv:1910.03474v1 [cs.CL] 4 Oct 2019
19. Marina Sokolova NJ (2006). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. Advances in Artificial Intelligence
20. Matsuoka KU (2017) Efficient breadth-first search on massively parallel and distributed-memory machines. Data Science and Engineering 2(1):22–35
21. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
22. Muangprathub J (2014) A novel algorithm for building concept lattice. Appl Math Sci 8(11):507–515
23. Ni Lao WW (2010) Fast query execution for retrieval models based on path-constrained random walks. In: KDD'10, Washington. USA, DC
24. Phuc Do PP (2018). DW-PathSim: a distributed computing model for topic-driven weighted meta-path-based similarity measure in a large-scale content-based heterogeneous information network. Journal of Information and Telecommunication, 1-20
25. Richard Socher DC (2013) Reasoning with neural tensor networks for Knowledge Base completion. Advances in Neural Information Processing:926–934
26. Santiago Gonzalez-Carvajal, EC-M (2020). Comparing BERT against traditional machine. Retrieved 5 17, 2020, from arXiv: 2005.13012v1 [cs.CL] 26 May 2020
27. Siva Reddy, D. C. (2019). CoQA: a conversational question answering challenge. arXiv:1808.07042v2

28. Suchanek FM, G. K. (2007). Yago: A core of semantic knowledge Unifying WordNet and Wikipedia. WWW 2007, New York, NY, USA
29. Sun Y, Han J (2011). Path-Sim: Meta path-based top-k similarity search in heterogeneous information networks. VLDB, (pp. 992–1003)
30. Tomasz Drabas DL (2017). Learning PySpark. Packt
31. Weninger BS (2017). ProjE: embedding projection for knowledge graph completion. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)
32. Xiangnan Kong PS (2012). Meta path-based classification in heterogeneous information networks. CIKM'12
33. Yadav R (2015) Spark cookbook. Packt Publishing
34. Zhao Zhang, FZ (2018) Knowledge graph embedding with hierarchical relation structure. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 3198-3207). Association for Computational Linguistics
35. Zhigang W, a. J. (2016). Text-Enhanced representation learning for knowledge graph. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 1293–1300
36. Zhilin Yang ZD (2019) XLNet: generalized autoregressive Pretraining for language understanding. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)
37. Zichen, Z, RC (2018) Evaluating top-k Meta path queries on large heterogeneous Information Networks. IEEE ICDM 2018, Singapore

Phuc Do is currently an Associate Professor at the University of Information Technology (UIT), VNU-HCM, Vietnam. His research interests include data mining, text mining, information network analysis, big data analysis and applications, Question Answering system, knowledge graph.

Truong H. V. Phan is a lecturer in the Information Technology department at Van Lang University, Ho Chi Minh City, Vietnam. He has been teaching computer science since 2009. His research interests are social network analysis, text mining, chatbot advisor, knowledge graph, and question answering system.