DETECTION OF SEMANTIC RELATIONS BASED ON KNOWLEDGE GRAPH

TA DUY CONG CHIEN
Khoa Công nghệ Thông tin, Trường Đại học Công nghiệp thành phố Hồ Chí Minh
taduycongchien@iuh.edu.vn

Abstract. Semantic relations have been applied to many applications in recent years, especially on Sematic Web, Information Retrieval, Information Extraction, and Question and Answer. Purpose of semantic relations is to get rid of conceptual and terminological confusion. It accomplishes this by specifying a set of generic concepts that characterizes the domain as well as their definitions and interrelationships. This paper describes how to detect semantic relations, including synonym, hyponym and hypernym relations based on WordNet and entities of Knowledge Graph. This Knowledge graph is built from two main resources: Wikipedia and unstructured files from ACM Digital Library. We used Natural Language Processing (NLP) and Deep Learning for processing data before putting into Knowledge Graph. We choose 5 of 245 categories in the ACM Digital Library to evaluate our results. Results generated show that our system yields superior performance.

Keywords. Knowledge graph, Semantic relation, Graph databases.

1 INTRODUCTION

Human knowledge is rich, varied and complex. There are many methods to representative human knowledge. A Knowledge Graph (KG) is one of natural candidates for representing this. NELL [1], Freebase [2], and YAGO [3] are examples of large knowledge graphs that include millions of entities and semantic relations. Semantic relations are represented as triples, each consisting of two entities connected by a binary relation. There are many kinds of semantic relations such as IS-A, Include, Synonym, Hyponym, etc.…

The KG including the semantic relations can be applied in many fields belonging to Computing such as: Search Engine, Information Retrieval, Information Extraction, Question answering. However, there are many challenges in order to build KG related to data, method and tools. Therefore, the KG is built for a long time and focusing on one domain.

The contributions of this paper are shown as follows: (i) we have crawled a large-scale dataset from the Wikipedia and ACM Digital Library by category focus on the computing domain in order to build KG. The KG concept approach tends to focus on the relationships/links of words rather than independently evaluating separated words; (ii) we propose an algorithm for detection many the semantic relations including synonyms, hyponyms and hypernyms based on the KG and WordNet.

The rest of this paper is organized as follows: section 2 - related works; section 3 – detection the semantic relations based on the knowledge graph; section 4 - experimental results and discussion; section 5 - conclusions and future works

2 RELATED WORKS

Information extraction is an important research topic in Natural language Processing (NLP) [4][5]. It tries to find semantic relations, relevant information from the large amount of text documents and on the World Wide Web. Y. Jie et al [6] focused on semantic rules to build an Extraction system from LIDAR (Light Detection and Ranging). F. Gomez et al [7] built semantic interpreter to assign meaning to the grammatical relations of the sentences when they constructed a knowledge base about a given topic. K. Kongkachandra et al [8] proposed semantic based key phrase recovery for domain-independent key phrase extraction. In this method, he added a key phrase recovery function as a post process of the conventional key phrase extractors in order to reconsider the failed key phrases by semantic matching based on sentence meaning. Z.Goudong et al [9] proposed novel tree kernel-based method with rich syntactic and semantic information for the extraction of semantic relations between named entities. A.B. Abacha et al [10] built a platform MeTAE (Medical Texts Annotation and Exploration). This system allows extracting and annotating Medical entities and relationships from Medical text. He relied linguistic pattern to detect semantic relations in medical text files. A.D.S Jayatilaka et al [11] constructed ontology from Web pages. He introduced web

usage patterns as a novel source of semantics in ontology learning. The proposed methodology combines web content mining with web usage mining in the knowledge extraction process. H. Li et al [12] extract semantic relations between Chinese named entities based on semantic features and Vector Space Model
Besides, in recent years, Knowledge graph is interested in the researchers for representing the big data. As outline from Xin Lv et al [13], they proposed a novel knowledge graph embedding model named TransC by differentiating concepts and instances. Specifically, TransC encodes each concept in knowledge graph as a sphere and each instance as a vector in the same semantic space. Besides, their knowledge graph is shown the semantic relations between concepts and instances and the semantic relations between concepts and sub-concepts. Xin Ly's research is just to encode each concept in knowledge graph as a sphere which is a simple model. G. Zhu et al [14] proposed a knowledge graph for exploiting semantic similarity for named entity disambiguation. They also proposed a Category2Vec embedding model based on joint learning of word and category embedding, in order to compute word-category similarity for entity disambiguation. The limit of this research is not to evaluate the performance of similarity methods when they are combined . B. Kotnis and V. Nastase [15] proposed Knowledge graphs, including only positive relation instances, leaving the door open for a variety of methods for selecting negative examples. They also present an empirical study on the impact of negative sampling on the learned embeddings, assessed through the task of link prediction. They used state-of-the-art knowledge graph embedding methods including Rescal , TransE, DistMult and ComplEX. S.S, but their results is based on the subset of Freebase and the subset of WordNet. Dasgupta et al [16] proposed HyTE, a temporally aware knowledge graph embedding method which explicitly incorporates time in the entity-relation space by associating each timestamp with a corresponding hyperplane. HyTE not only performs knowledge graph inference using temporal guidance, but also predicts temporal scopes for relational facts with missing time annotations. X, but this research is only to exploit temporally scoped facts of KG to perform link prediction as well as prediction of time scopes for unannotated temporal facts. B. Ding et al [17] investigated the potential of using very simple constraints to improve knowledge graph embedding, but this research is only focus on two constraints, namely, the non-negativity constraints to learn compact, interpretable entity representations, and the approximate entailment constraints. K. Wang et al [18] proposed a new kind of additional information, called entity neighbors, which contain both semantic and topological features about giving entity. The limit of this research is regardless the semantic of entity neighbors. A. Kutuzov et al [19] proposed path2vec, a new approach for learning graph embeddings that relies on structural measures of pairwise node similarities. In the future, they plan to explore the possibility of training embeddings able to approximate multiple similarity metrics at once.
Generally, there are a lot of methods to have knowledge graph for applied to many different fields. The research can apply approaches related to NLP, Machine Learning, Deep learning or hybrid approaches. In this paper, we use NLP and Deep Learning for data training to build KG focusing computing domain. After that, we detect the semantic relation based on this graph.

3 HETEROGENOUS DOCUMENTS BASED KNOWLEDGE GRAPH EMBEDDING

The approach for detection semantic relations based on Knowledge Graph is shown in Fig.1 including input and output data of each step.

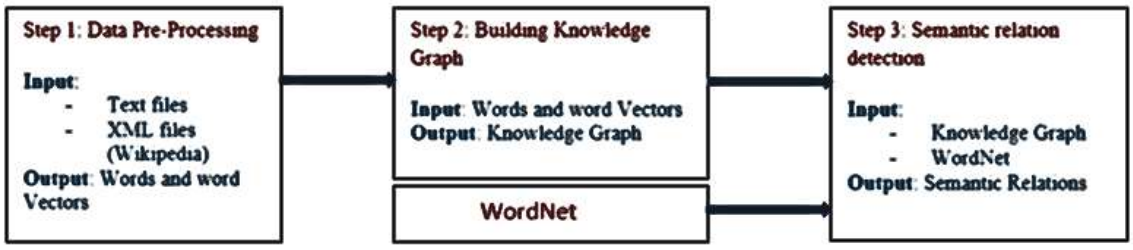

Figure 1. The approach for detection Semantic relations based on Knowledge Graph.

Definition 1. A knowledge graph G includes vertex representing entities, class, subclass, and edges representing relationship among vertexes.

3.1 Building KG from text documents of ACM Digital Library

The process for training text documents of the ACM Digital Library includes 2 steps:

- The first is data pre-processing
- The second is using Keras framework, including a word embedding model of text data.

In the first phrase, all of text files of ACM Digital Library are merged by their category. After merging, each category has only one text file. These text files are as input and it is sent to Tokenizer. The Tokenizer split the sentences into words based on whitespace character. The tokenized words are taken to extractor for converting to lowercase, removing punctuation from each token and filtering out remaining tokens that are not alphabetic as well as filtering out tokens that are stop words. After removing stop words from the text files, these text files are taken to extractor again for stemming process. Stemming refers to the process of reducing each word to its root or base. For example, having, had, has all reduces to the stem have. Some applications, like document classification, may benefit from stemming in order to both reduce the vocabulary and to focus on the sense or sentiment of a document rather than deeper meaning. There are many stemming algorithms, although a popular and long-standing method is the Porter Stemming algorithm. In addition, we use Natural Language Tool Kit (NLTK) [20] for data pre-processing.

In the second phrase, we use Keras [21] framework using Recurrent Neutral Network (RNN) model with word embeddings for training data. The RNN model for training data is shown in Fig 2.

Additionally, in the Figure 1, the word layer includes the words which were processed in the first phrase and the hidden layers includes 4 layers.

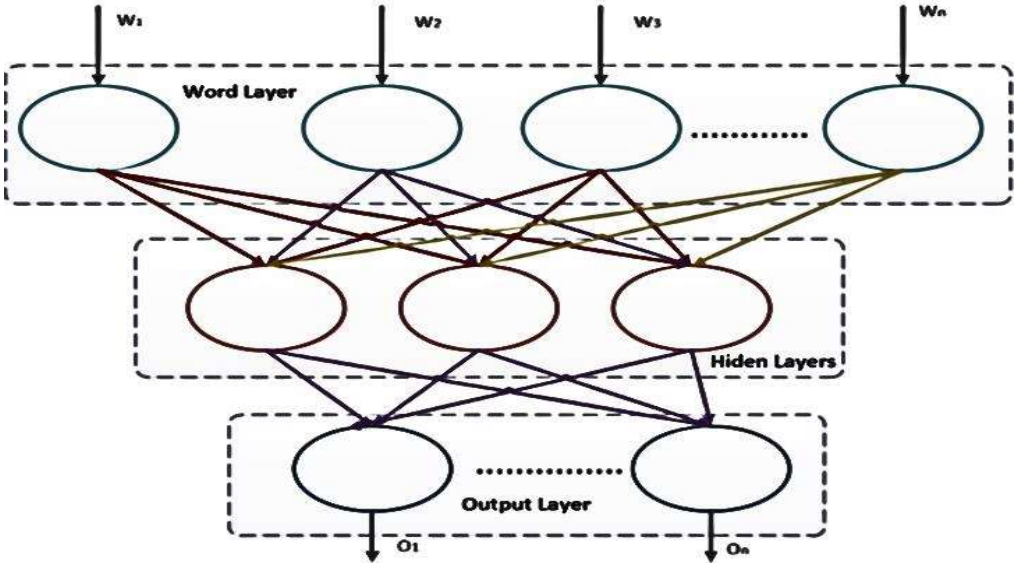

Figure 2. The model uses Keras framework using RNN model with word embedding layer (4 hidden layers, max_feature=40, activation=sigmoid, 64 dimensions, 2000 tokens).

The next step is to build KG. The structure of KG is separated into two layers and Computing domain is a root of KG. The first layer is known as the Subject layer [22]. This layer includes categories which are extracted from ACM Classification Categories [23]. We obtain over 30 different categories from this site. The next layer of KG is known as the Object layer. This layer contains many different word vectors which are output from Word2Vec word embedding model, e.g., “Hardware”, “SQL Server”, “Java”, “CPU”, “Oracle”, “Data Structure”, etc. The KG representing for Computing domain is shown as Fig. 3

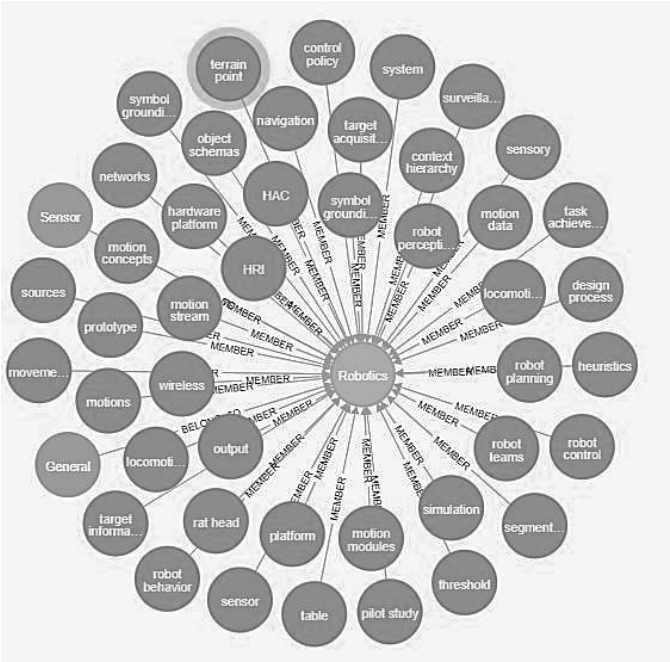

Figure 3: The hierarchy of Knowledge Graph

3.2 Updating KG from XML documents of Wikipedia

The process to update KG by entities extracted from Wikipedia [24] includes three steps:

- The first step is to prepare XML files including entities belong to categories of ACM Digital Libraries
- The second step we take data pre-processing with XML file getting from the first phrase

The thirst step we reuse Keras [21] framework using Recurrent Neutral Network (RNN) model with word embeddings for training data (4 hidden layers, max_feature=40, activation=sigmoid, 64 dimensions, 2000 tokens).

Additionally, in order to access and extract data belong to a category from Wikipedia, we use the API functions which provide by Wikipedia.

3.3 The algorithms for detection the semantic relations based on the knowledge graph

This paper focus on the semantic relations, including synonym, hyponym and hypernym. Those semantic relations play an important role in information retrieval. To find out those semantic relations, we use KG and WordNet. Our proposed algorithm is as follows.

```
Procedure Find_out_SYN_HYPO_HYPE
  While Instance is not null
 Begin
  Instance = root
  Find_out_SYN_HYPO_HYPE(root)
  Root = root.LEFT
  Root = root.RIGHT
  SYN = Select WordNet.SYNONYM where WordNet.Instance = Instance
  HYPO =   Select WordNet.HYPONYM where WordNet.Instance = Instance
  HYPE = Select WordNet.HYPERNYM where WordNet.Instance = Instance
 End
```

After applied the above algorithm, we extracted the semantic relations from WordNet corresponding with entities of KG for example, some of the results are shown in Table 1.

Table 1: Set of Synonym, Hyponym and Hypernym corresponding with entities of KG

Entities of KG	Synonyms	Hyponyms	Hypernyms
NLP	Natural Language Processing		Informatics, information processing
Data structure		Hierarchical structure	Organization, system
Computer Network		Internet, intranet, WAN	Electronic network
RAM	Random Access Memory	Core memory	Volatile storage

From Table 1, we can see some semantic relations between an instance of KG with its synonyms, hyponyms and hypernyms, such as

- NLP is a Natural Language Processing
- NLP such as Informatics, information processing
- Hierarchical structure includes Data structure
- Data structure such as organization, system
- RAM is random access memory
- Core memory includes RAM

4 EXPERIMENTAL RESULT AND DISCUSSION

4.1 Evaluation based on three measures

We implement numerous experiments for studying the efficiency of the proposed approach. We select papers which have only abstract part belong to five categories from ACM Digital Library for testing as following.

- 100 abstracts in Software category.
- 100 abstracts in Process Management category.
- 100 abstracts in Artificial Intelligent category.
- 100 abstracts in Operating system category.
- 100 abstracts in Logic Design category.

We use three measures: Precision (P), Recall (R) and F-measure for experimental evaluation.

$$P(C_i) = \frac{Correct(C_i)}{Correct(C_i) + Wrong(C_i)} \quad (1)$$

$$R(C_i) = \frac{Correct(C_i)}{Correct(C_i) + Missing(C_i)} \quad (2)$$

$$F - measure(C_i) = 2\frac{Precision(C_i) * Recall(C_i)}{Precision(C_i) + Recall(C_i)} \quad (3)$$

Where:

C_i denotes a category in KG; Correct (C_i) denotes a number of the semantic relations which are found in KG and they accurately belong to the category C_i; Wrong (C_i) denotes a number of the semantic relations which are found in KG, but they do not belong to category C_i; Missing (C_i) denotes a number of the semantic relations which are not found in KG. The evaluation results obtained are shown in Tables 2, 3, 4, 5.

Table 2: Evaluation results on instances of KG

Category	**Number of instances**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
Application	3672	79.26	76.51	77.86

Artificial Intelligent	5714	82.94	78.92	80.88
Logic Design	4644	82.18	80.06	81.11
Operating System	6785	84.47	81.37	82.89
Process Management	3056	76.53	72.51	74.47
Software	4249	81.64	79.62	80.62

The result from table 2 reveals that for different number of instances which extracted after pre-processing, the precision along with recall and F-measure will also be different. In all the categories that the experiment consists of, "Operating system" has the highest number of instances, therefore, it results in highest precision and recall among that of other categories. Whereas, "Software" category has the least instances, therefore, its precision and recall are remained the lowest. This experiment shows that the accuracy of semantic relations is found based on KG of a category will be directly proportional to the number instances of that category.

Table 3: Evaluation results on set of synonym relations

Category	**Quality of synonym**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
Application	524	79.26	76.51	77.86
Artificial Intelligent	689	94.41	88.15	91.17
Logic Design	472	92.24	84.27	88.08
Operating System	861	96.18	91.58	93.82
Process Management	517	93.25	86.16	89.56
Software	583	94.26	89.04	91.57

The result from table 3 reveals that for different number of synonym relations detected based on KG, the precision along with recall and F-measure will also be different. In all the categories that the experiment consists of, "Operating system" has the highest number of synonym relations, therefore, it results in highest precision and recall among that of other categories. Whereas "Logic Design" category has the least synonym relations, but its precision and recall are higher the precision and recall of “application” category. This experiment shows that the accuracy of synonym relations is found based on KG of a category will not be directly proportional to the synonym relation number of that category.

Table 4: Evaluation results on set of Hyponym relations

Category	Quality of Hyponym	Precision (%)	Recall (%)	F-Measure (%)
Application	714	89.38	76.51	82.45
Artificial Intelligent	837	96.14	88.29	92.04
Logic Design	718	87.54	84.26	85.86
Operating System	972	96.82	91.42	94.04
Process Management	728	88.31	85.15	86.70
Software	646	85.64	81.04	83.28

Similarly, the result from table 4 reveals that for different number of hyponym relations detected based on KG, the precision along with recall and F-measure will also be different. In all the categories that the experiment consists of, "Operating system" has the highest number of hyponym relations, therefore, it results in highest precision and recall among that of other categories. Whereas "Software" category has the least hyponym relations, but its precision is lower the precision of “application” category, and its recall are higher the recall of “application” category. This experiment shows that the precision and recall of hyponym relations are found based on KG of a category will not be directly proportional to the hyponym relation number of that category

Table 5: Evaluation results on set of Hypernyms

Category	Quality of Hypernym	Precision (%)	Recall (%)	F-Measure (%)
Application	916	79.26	76.51	77.86
Artificial Intelligent	1321	92.41	91.17	91.79
Logic Design	954	84.62	79.37	81.91
Operating System	1413	95.04	96.81	95.92
Process Management	834	82.31	84.55	83.41
Software	893	85.48	80.19	82.75

Similarly, the result from table 5 reveals that for different number of hypernym relations detected based on KG, the precision along with recall and F-measure will also be different. In all the categories that the

experiment consists of, "Operating system" has the highest number of hyponym relations, therefore, it results in highest precision and recall among that of other categories. Whereas "Process Management" category has the least hyponym relations, but its precision and recall are higher the precision and recall of "application" category. This experiment shows that the precision and recall of hypernym relations are found based on KG of a category will not be directly proportional to the hyponym relation number of that category

The number of semantic relations obtained from instances of KG is shown in Fig 3.

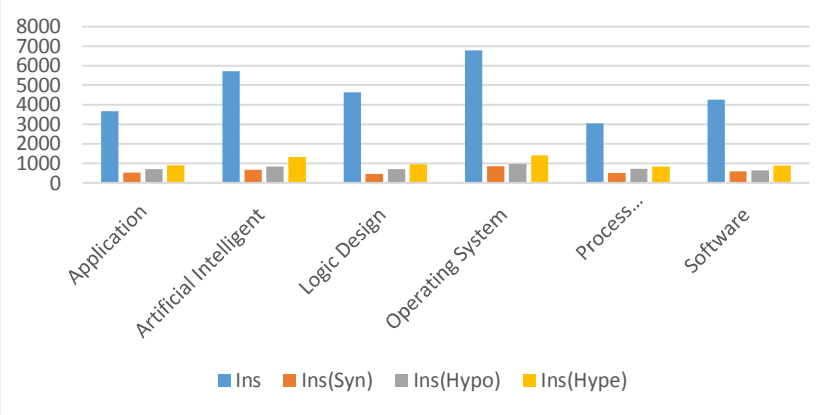

Figure 3. The number of instances of 5 categories and the number of instances of synonym, hyponym and hypernym relations successively.

The result from Fig.3 reveals that for different number of instances which extracted after pre-processing, the number of synonym, hyponym and hypernym relations which detected based on KG will also be different. In all the categories that the experiment consists of, "Operating system" has the highest number of instances, therefore, it results in highest number of synonym, hyponym and hypernym relations among that of other categories.

The comparison between precision percentages of the different categories is shown in Fig 4

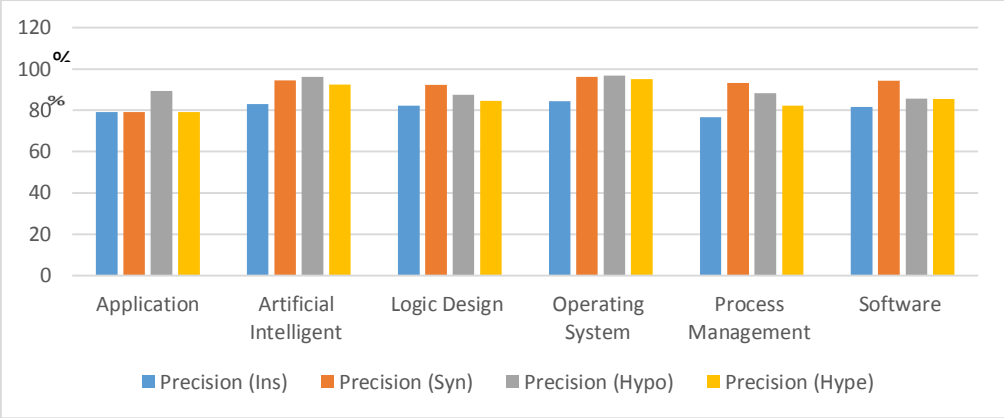

Figure 4. The precision percentages of synonym, hyponym and hypernyms relations successively.

The result from Fig. 4 reveals that for different categories, the precision percentage of synonym, hyponym and hypernym relations which detected based on KG will also be different. In all the categories that the experiment consists of, "Operating system" has the highest precision percentage among that of other categories because it has the highest number of synonym, hyponym and hypernym relations.

The comparison between recall percentages of the different categories is shown in Fig 5

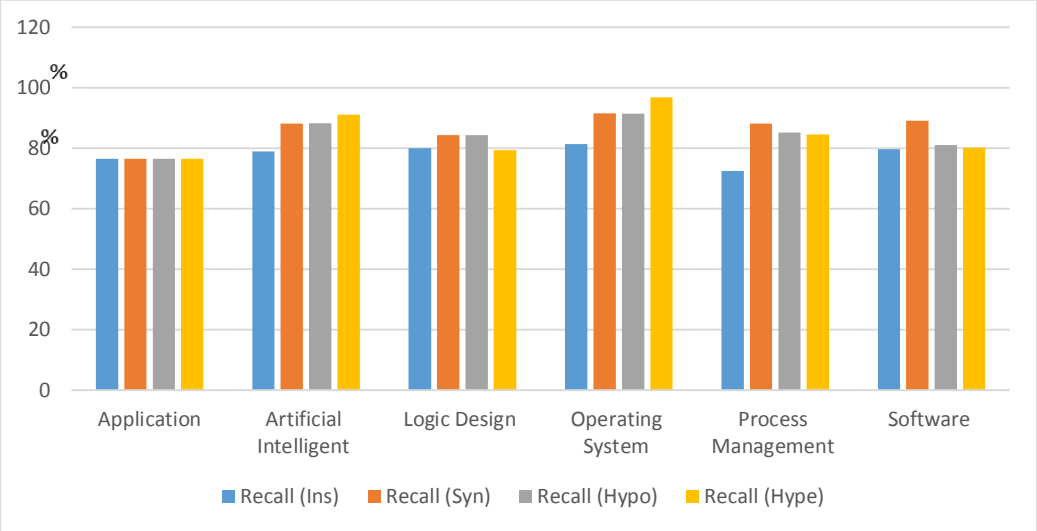

Figure 5. The recall percentages of synonym, hyponym and hypernyms relations successively.

Similarly, the result from Fig. 5 reveals that for different categories, the recall percentage of synonym, hyponym and hypernym relations which detected based on KG will also be different. In all the categories that the experiment consists of, "Operating system" has the highest recall percentage among that of other categories because it has the highest number of synonym, hyponym and hypernym relations.

4.2 Comparative evaluation method

In order to compare the precision and recall of instances which obtain from our model (table 2) We use Stanford CoreNLP [25] for comparative evaluation method. Stanford CoreNLP is a tool for extraction of instances and relations among instances from text documents. Stanford CoreNLP supports the API functions to develop the applications related to NLP. We pick two categories for comparability; the result is shown as below:

Table 6: Comparative evaluation method

Category	Number of instances	Precision (%)	Recall (%)	F-Measure (%)
Our Approach				
Application	3672	79.26	76.51	77.86
Process Management	3056	76.53	72.51	74.47
Stanford CoreNLP approach				
Application	3904	68.46	62.13	65.14
Process Management	3271	62.37	58.75	60.50

The scores reported in table 6 reveals that the number of instances obtained from Stanford CoreNLP tool is greater than the number of instances obtained from Deep learning model, but the precision and recall of our proposed approach are higher than the CoreNLP tool because Deep learning model is interested context when processing the words in text documents. Generally, our proposed method outperforms the Stanford CoreNLP tool. Currently, because we combine the three different corpus including text files, Wikipedia, and WordNet to detect semantic relations, therefore we cannot compare with the other approaches using deep learning for detection semantic relations.

5 CONCLUSIONS AND FUTURE WORKS

Our experiment tried to detect the semantic relations, including synonym, hyponym and hypernym relations based on KG and WordNet. Especially, the KG concept approach tends to focus on the relationships/links of words rather than independently evaluating separated words and the KG is only focus on computing domain. Currently, this KG has 170 categories and one million entities. To solve the problem, we proposed an approach has two steps, including data training for building KG and finding out the semantic relations based on KG and WordNet. We use Keras model on RNN model (four hidden layers) associating with word

embedding layer (2000 tokens, 64-dimensional and sigmoid activation) for data training after pre-processing the data, which are extracted from the ACM Digital Library and Wikipedia. We also use the Neo4J Graph Database for building KG after data training. To detect semantic relations, we propose the search algorithm based on KG and WordNet. We also apply three measures as Precision, Recall and F-Measure for evaluating our approach. In the future, we will combine WordNet ontology into KG for reducing time of query on WordNet Ontology.

REFERENCES

[1]. *Never-Ending Language Learning* - NELL. [Online]. Available: http://rtw.ml.cmu.edu/rtw/

[2]. [Online]. Available: https://developers.google.com/freebase

[3]. [Online]. Available: https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[4]. G. Zhou, Q. Longhua and F. Jianxi, *Tree Kernel-Based Semantic Relation Extraction with Rich Syntactic and Semantic Information*, *Journal of Information Sciences*, 2010, vol. 180, no. 8, pp. 1313 – 1325.

[5]. Q. Wang, Z. Mao and B. Wang, *Knowledge Graph Embedding: A Survey of Approaches and Applications*, in *Proc. Int. Conf. on IEEE Transactions on Knowledge and Data Engineering (TKDE 29)*, 2017, page 2724–2743.

[6]. Y. Jie, Y. Haiquan, T. Ming and Z. Guoning, *Building Extraction from LIDAR based Semantic Analysis*, *Journal of Geo-Spatial Information Science*, 2006, vol. 9, no. 4.

[7]. F. Gomez and C. Segamib, *Semantic interpretation and knowledge extraction*, *Journal of Knowledge-Based Systems*, 2006, vol. 20, no. 1, pp. 51 - 60.

[8]. G. Kongkachandra and K. Chamnongthai, *Abductive Reasoning for Keyword Recovering in Semantic-based Keyword Extraction*, in *Proc. Int. Conf. on The Fifth International Conference on Information Technology: New Generations - IEEE*, 2008, pp. 714 - 719.

[9]. Z. Goudong, Qian, Longhua, Fan and Jianxi, *Tree kernel-based semantic relation extraction with rich syntactic and semantic information*, *Journal of Information Sciences*, 2009, vol. 180, no. 8, pp. 1313 – 1325.

[10]. A.B Abacha and P. Zweigenbaum1, *Automatic Extraction of Semantic Relations between Medical Entities- a rule based approach*, *Journal of Biomedical Semantics*, 2011.

[11]. A.D.S Jayatilaka, *Knowledge Extraction for Semantic Web Using Web Mining*, in *Proc. Int. Conf. on Advances in ICT for Emerging Regions (ICTer 2011) - IEEE*, 2011, pp. 89 - 94.

[12]. H. Li, X. Wu. Z Li and G. Wu, *A Relation Extraction Method of Chinese Named Entities based on Location and Semantic Features*, *Journal of Applied Intelligence*, 2012, vol. 18, no. 1, pp. 1- 14.

[13]. X. Ly, L. Hou, J. Li and Z. Liu, *Differentiating Concepts and Instances for Knowledge Graph Embedding*, in *Proc. Int. Conf. on Empirical Methods in Natural Language Processing*, 2018

[14]. G. Zhu, *Exploiting semantic similarity for named entity disambiguation in knowledge graphs*, *Journal of Expert Systems with Applications*, 2018, vol 101.

[15]. Kotnis and V. Nastase, *Analysis of the impact of negative sampling on link prediction in knowledge graphs*, *the Computing Research Repository (CoRR)*, 2017

[16]. S.S. Dasgupta, S. N. Ray and P. Talukdar, *HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding,* in *Proc. Int. Conf. on Empirical Methods in Natural Language Processing*, 2018, pages 2001–2011.

[17]. B. Ding, Q. Wang, B. Wang and L. Guo, *Improving Knowledge Graph Embedding Using Simple Constraints*, in *Proc. Int. Conf. on the 56th Annual Meeting of the Association for Computational Linguistics*, 2018

[18]. K. Wang, Y. Liu, X. Xu and D. Lin, *Knowledge Graph Embedding with Entity Neighbors and Deep Memory Network*, *the Computing Research Repository (CoRR)*, 2018

[19]. A. Kutuzov, M. Dorgham, O. Oliynyk, C. Biemann and A. Panchenko, *Learning Graph Embeddings from WordNet-based Similarity Measures*, in *Proc. Int. Conf. on the 8th Joint Conference on Lexical and Computational Semantics*, 2019

[20]. *NLTK Project*. [Online]. Available: https://www.nltk.org/news.html

[21]. *Keras Project.* [Online]. Available: https://keras.io/

[22]. Chien. Ta Duy Cong, Tuoi. Phan Thi, *Building Ontology Based-on Heterogeneous Data*, *Journal of Computer Science and Cybernetics*, 2015, vol. 31, no.2, ISSN: 1813-9663.

[23]. *The ACM Computing Classification System.* [Online]. Available: https://www.acm.org/publications/computing-classification-system/1998/ccs98

[24]. *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[25]. *Stanford CoreNLP – a suite of core NLP tools, Stanford University.* [Online]. Available: http://stanfordnlp.github.io/CoreNLP/

PHÁT HIỆN CÁC QUAN HỆ NGỮ NGHĨA DỰA TRÊN ĐỒ THỊ TRI THỨC

Tóm tắt. Trong những năm gần đây các quan hệ ngữ nghĩa được áp dụng trong nhiều ứng dụng, đặc biệt là trong lãnh vực Web ngữ nghĩa, Truy xuất thông tin, Khai thác thông tin và các Hệ thống trả lời câu hỏi. Mục đích của các quan hệ ngữ nghĩa là để loại bỏ sự nhầm lẫn về các khái niệm và thuật ngữ. Các quan hệ ngữ nghĩa thực hiện điều này bằng cách chỉ định một tập hợp các khái niệm chung đặc trưng cho miền cũng như các định nghĩa và mối quan hệ của chúng. Bài báo này nhằm mô tả làm cách nào để phát hiện các mối quan hệ ngữ nghĩa bao gồm các quan hệ đồng nghĩa, hạ tầng và thượng tầng dựa trên WordNet vá các thực thể của đồ thị tri thức. Đồ thị tri thức được xây dựng từ hai nguồn ngữ liệu chính: Wikipedia và các tập tin không có cấu trúc được lấy từ Thư viện số ACM. Chúng tôi đã sử dụng Xử lý ngôn ngữ tự nhiên và Học sâu để xử lý dữ liệu trước khi đưa vào đồ thị tri thức. Chúng tôi đã chọn 5 trong số 245 chủ đề trong Thư viện số ACM để đánh giá kết quả. Kết quả tạo ra cho thấy hệ thống của chúng tôi mang lại hiệu suất vượt trội như mong đợi.

Từ khóa. Đồ thị tri thức; Quan hệ ngữ nghĩa; Cơ sở dữ liệu đồ thị

Received on: 06/05/2020

Accepted on: 15/01/2021