AUTOMATIC SUBJECT LABELING IN DOCUMENTS BY USING ONTOLOGY AND GRAPH DATABASES

TẠ DUY CÔNG CHIẾN

Trường Đại học Công nghiệp Thành phố Hồ Chí Minh,
taduycongchien@iuh.edu.vn

Abstract. Ontologies apply to many applications in recent years, such as information retrieval, information extraction, and text document classification. The purpose of domain-specific ontology is to enrich the identification of concept and the interrelationships. In our research, we use ontology to specify a set of generic subjects (concept) that characterizes the domain as well as their definitions and interrelationships. This paper introduces a system for labeling subjects of a text documents based on the differential layers of domain specific ontology, which contains the information and the vocabularies related to the computer domain. A document can contain several subjects such as data science, database, and machine learning. The subjects in text document classification are determined based on the differential layers of the domain specific ontology. We combine the methodologies of Natural Language Processing with domain ontology to determine the subjects in text document. In order to increase performance, we use graph database to store and access ontology. Besides, the paper focuses on evaluating our proposed algorithm with some other methods. Experimental results show that our proposed algorithm yields performance significantly
Keywords. Ontology, Subject labeling, Graph databases.

1 INTRODUCTION

Domain ontology, including of the concepts and the relations among the concepts, is applied in a variety of applications. The automatic subject labeling of a text document is one of the applications to be applied to the domain specific ontology. The labeling of subjects in a text document plays an important role in the science. It helps the scientists to categorize the submitted papers in order to review and arrange the papers into the right sessions in the conferences. Besides, It help us to capture the scientific subjects in a particular document. According to the traditional methods, the labeling of subjects in the text documents uses a keyword distribution form a training corpus to assign label to subjects in a document [1]. However, using only keywords in a training set cannot guarantee accuracy results since authors may use different keywords in the different documents. Previous research shows that the Latent Semantic Index (LSI) method [2] and the n-gram method give good results for Chinese news categorization. However, the indices of LSI and n-grams are less meaningful semantically.

With good domain ontology we can identify the subjects of sentences in a document. Our idea is to use the keywords in a sentence to find out the subject of a sentence. After that we will combine all of the subject of the sentences in a document to point to the main subjects that the document can have. However, building rigorous domain ontology is laborious and time-consuming. But until now, we have already had a domain specific ontology focusing on Computer domain. In this domain, each concept is a subject of application domain.

My key contributions are as follows: (i) I proposes a hierarchical structure of the domain specific ontology and save it in Neo4j graph database, so we can access efficiently the ontology; (ii) I proposes a novel method for obtaining the list of topic keywords from a text document by the Stanford Dependency Parser (SDP) [3]; (iii) the algorithm for mapping the list of topic keyword into domain specific ontology for automatic subject labeling in the text documents. (iv). The performance increases significantly, because the ontology is stored in a graph database.

The rest of this paper is organized as follows: section 2 - related works; section 3 - automatic subject

labeling of text document; section 4 - experimental results and discussion; section 5 - conclusions and future works

2 RELATED WORKS

As outline from Ipeirotis et al [4] , they applied Machine Learning to build the system, which could classify and search Hidden Web text databases. The system is called QProber. QProber categorizes databases without retrieving any document. Instead, QProber uses just the number of matches generated from a small number of topically focused query probes. AlSumait and Domeniconi [5] presented a subspace clustering technique based on a locally adaptive clustering (LAC) algorithm. To improve the subspace clustering of documents and the identification of keywords achieved by LAC, kernel methods and semantic distances are deployed. The basic idea is to define a local kernel for each cluster by which semantic distances between pairs of words are computed to derive the clustering and local term weightings. Their experiments show that semantic LAC is capable of improving the clustering quality. Sebastiani [6] pointed to the challenges of automated text classification. His survey discusses the main approaches to text categorization that fall within the machine learning paradigm. This survey discusses in detail issues pertaining to three different problems, namely, document representation, classifier construction, and classifier evaluation. Salton et al [7] explored the use of Skip-Thought Vectors to create distributed representations that encode features that are predictive with respect to idiom token classification. They show that classifiers using these representations have competitive performance compared with the state of the art in idiom token classification.

Generally, the previous research works focus on the keyword distribution in the documents. In this paper, we don't only focus the keyword distribution or keyword graphs [8-10], but also mention the semantics of the sentences that keywords appear in these sentences

3 AUTOMATIC SUBJECT LABELING OF THE TEXT DOCUMENTS

Definition 1 - subject: a subject is a category or topic of a text document.

Definition 2 - topic keywords: the keywords that can properly summarize a topic would enable users to obtain a brief idea about the topic even without reading its relevant papers [11].

3.1 Overview of the Computer Domain Ontology

Ontology is a formal and explicit specification of a shared conceptualization of a domain of interest. Their classes, relationships, constraints and axioms define a common vocabulary to share knowledge. Conceptualization refers to an abstract model of some phenomenon in the world. Explicitly, it means that the type of concepts used and the limitations of their use are explicitly defined. Formally, it refers to the fact that the ontology should be machine-readable. Shared, it reflects the notion that ontology captures consensual knowledge, that is, it is not private to some individual but accepted by a group.

Formally, ontology can be defined as a tuple [8]:

$$O = (C, I, S, N, H, Y, B, R)$$

Where,

C, is set of classes, i.e., Concepts represent categories of the computer domain (for example, “Artificial Intelligence, hardware devices, NLP” (C)

I, is a set of instances belong to categories. Set I consists of vocabulary of computer (for example, “robotic, Random Access Memory “$\in$ I)

$S = N^S \cup H^H \cup Y^H$ is the set of synonyms, hyponyms and hypernyms of instances of set I.

$N = N^S$ is the set of synonyms of instances of set I.

$H = H^H$ is the set of hyponyms of instances of set I.

Y = YH is the set of hypernyms of instances of set I. (e.g., “ADT”, “data structure”, “ADT is a kind of data structure that is defined by programmer” are synonymous, hyponymous and hypernymous of

"Abstract data type")

B = {belong_to (i, c) | i ∈ I, c ∈ C} is the set of semantic relationships between concepts of set C and instances of set I and are denoted by {belong_to (i, c) | i ∈ I, c ∈ C} mean that i belong to category c. (e.g., belong_to ("robotic", "Artificial Intelligence")

R = {rel (s, i) | s ∈ S, i ∈ I} is the set of relationships between terms of set S and instances of set I and are denoted by hierarchy and are denoted by {rel (s, i) | s ∈ S, i ∈ I} and mean that s has a relationship with i.

The relationships can be synonymous, hyponymous or hypernymous. (e.g., synonym ("ADT", "Abstract data type"), hyponym ("data structure", "Abstract data type"), hypernym ("ADT is a kind of data structure that is defined by programmer", "Abstract data type").

In addition, all concepts, instances of this ontology focus on computer domain; therefore, this ontology is known as Computer Domain Ontology (CDO).

The structure of CDO is separated into four layers:

The first layer is known as the Topic layer. In order to build it, we extract terms from ACM Classification Categories [12]. We obtain over 170 different categories from this site and rearrange them in this layer.

Next layer is known as the Ingredient layer. In this layer, there are many different instances, which are defined as nouns or compound nouns from vocabulary about computer domain, e.g., "robot", "Super vector machine", "Local Area network", "wireless", "UML", etc. In order to setup this layer, we use Wikipedia to focus on English language and computer domain.

The third layer of CDO is known as the Relation layer. To set up this layer, we use the WordNet ontology. Similar to Wikipedia, we only focus on computer domain. This layer includes synonyms, hyponyms, and hypernyms of instances of the Ingredient layer. As we combine two ontologies, which are Wikipedia and WordNet. The instances that belong to many different categories of computer domain. That is an advantage of this ontology.

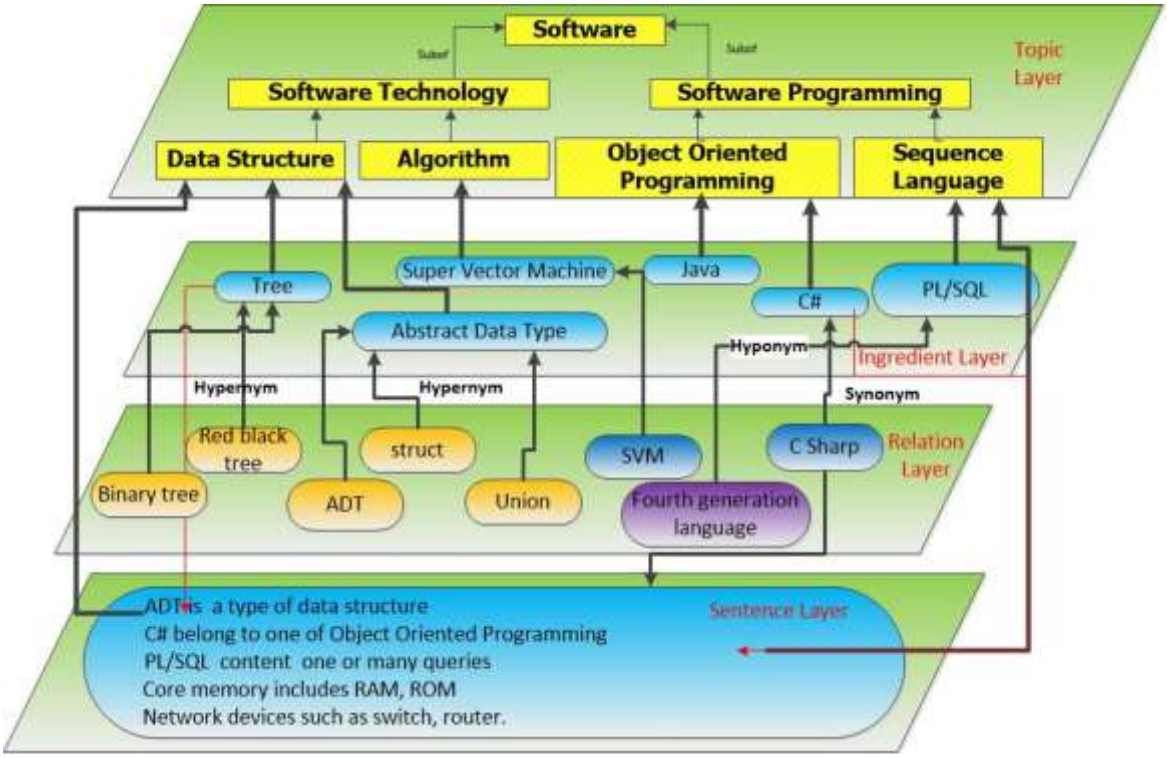

Figure 1: The hierarchy of CDO

The last layer of CDO is known as the Sentence layer. Instances of this layer are sentences that represent syntactic relations extracted from preprocessing stage. Hence, these sentences are linked to one or many terms of the Ingredient layer. This layer also includes sentences that represent semantic relations between terms of Ingredient layer, such as, IS-A, PART-OF, MADE-OF, RESULT-OF, etc. The overall hierarchy of CDO is shown in Fig. 1.

An approach for automated subject labeling of text documents has three steps:

- Splitting Sentence.
- Extracting topic keywords from a sentence based on SDP
- Labelling subject of a document based on CDO.

3.2 Splitting Sentence

There may be one or more sentences in a text document. These sentences are usually separated by symbols, such as dot (“.”), question mark (“?”), exclamation mark (“!”), etc. We use OpenNLP [13] to detect and extract sentences one by one before going through next step

3.3 Extracting topic keywords from a sentence based on SDP

3.3.1 Part-Of-Speech tagging (POS tag)

We use SDP to identify keywords in a sentence. SDP implements POS tagger, it is also called grammatical tagging or word-category disambiguation. SDP is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context—i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. After POS tagging, we use SDP to generate a syntactic tree, for example: sentence “A quick brown fox jumped over the lazy dog”. The sentence’s syntactic tree is shown as Fig. 2.

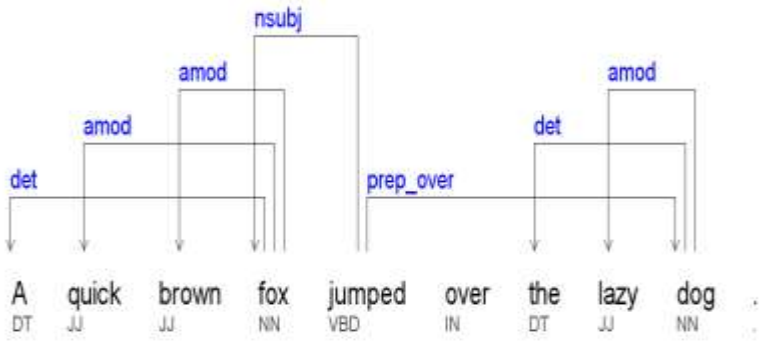

Figure 2: Syntactic tree of the sentence “A quick brown fox jumped over the lazy dog

3.3.2 Keyword Extraction

Keywords are identified and extracted based on syntactic tree. We will extract nouns and compound nouns from the syntactic tree since nouns and compound nouns can properly summarize a topic of a sentence, especially nouns and compound nouns are subject or complement object positions in a sentence. If the sentence has preposition clauses, they will be drop since these clauses usually do not summarize a topic of a sentence.

We propose Algorithm 1 for extracting keywords in a text document by using SDP as follows

Algorithm 1. Keyword Extracting from a document.

Input: A Text document D

Output: List of keywords K

Sentence S;

While (end of D)

S = OpenNLP(Sentence chopping)
List of keywords K ←∅
Dependency graph G ← SDP(S) /*SDP generate dependency graph */
G = Remove(Preposition clause)
For each node of P
If (Existing Subject in G) then /* Stanford dependency representation is **nsubj**) */
keyword ← Subject
Else
If (Existing noun/noun phrase in G) then
keyword ← noun/noun phrase
End if
End if
End For
End While
Return K

According to algorithm 1, we use the SDP tool for collecting the keywords in a sentence. Firstly, we select the subject of a sentence. Next, the nouns, compound nouns, noun phrases or subject modifiers will be considered

3.4 Labeling subjects of a document based on Computing Domain Ontology

Ontology is a kind of graph. Ontology has entities, properties and relationship. Hence, the structure of CDO of new proposed structure is shown as Fig. 3.

- **Conversions on ⟨*T*⟩ & ⟨*I*⟩**: all the topics and subjects in both ⟨*T*⟩ and ⟨*I*⟩ will be considered as data's node with properties such as *<url>, <name> <label>*…
- **Conversions on ⟨*R*⟩**: the relationships between entities will be represented as the graph's edges – each edge (or relationship between nodes). They carry the related parameters to identify which CDO's relation's type it is (IS_A, HAS_PART, RELATED_TO…).
- Each subject nodes also contain keywords related to this subject. We use Neo4j graph database to save our computer domain ontology. The Browser of CDO is shown in Fig. 3.

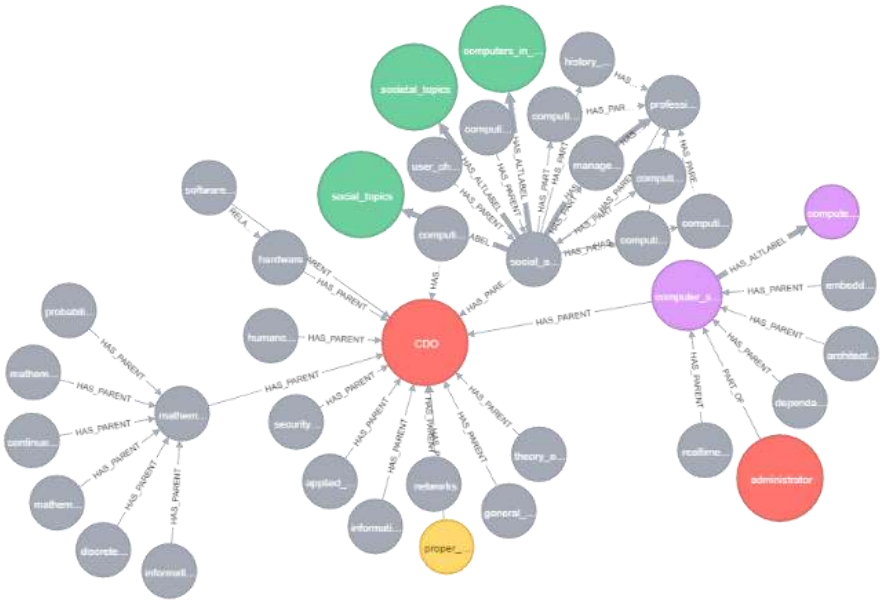

Figure 3: The Browser of CDO in Neo4J graph database

After storing the CDO's data on Neo4J graph database, we can use Cypher query language of Neo4J to process the automatic subject labeling. This is a last step to determine the subjects of a document based on the topic layer of CDO.

Definition 3 - interpretation sentence: An interpretation sentence of the list of keywords query $K = \{k_1, k_2, \ldots, k_n\}$ on a graph database is a graph traversal.

This graph traversal will use DFS algorithm to scan the subject nodes of ontology and compare the keywords in document with the keywords in subject node. We propose Algorithm 2 for labeling subjects in a text document as follows.

Algorithm 2**.** The algorithm for labelling subjects automatically in a text document

Input: *List of keywords K*
Output: *Subjects of a document*
For each *keyword* k_i *in the order list of keywords K*
If (k_i *is abbreviation) then*
Mapping k_i *to synonym concept by graph traversal*
Result ← *instance of synonym table =* k_i
Else
If (k_i *has one or many preposition and i=1) /* k_i *is the first keyword in the order list K */*
Mapping k_i *to sentence relation by graph traversal*
Result ← *instances of sentence table like* k_i
Else
If (i=1) then
Mapping k_i *to ingredient relation by graph traversal*
Result ← *instance of Ingredient table =* k_i
Else
Mapping k_i *to attribute relation by graph traversal*
Result ← *instance of Attribute table =* k_i
End if;
End if ;
End if
End For

According to algorithm 2, list of keywords, which results of algorithm 1, will be mapped into the different layers of CDO in order to identify the subjects in documents. We use graph traversal by Cypher Query Language in Neo4j graph database to generate automatically the subjects in documents.

4 EXPERIMENTAL RESULT AND DISCUSSION

We have already built an application for labeling the subjects of a document. Our System is shown in Fig. 4.

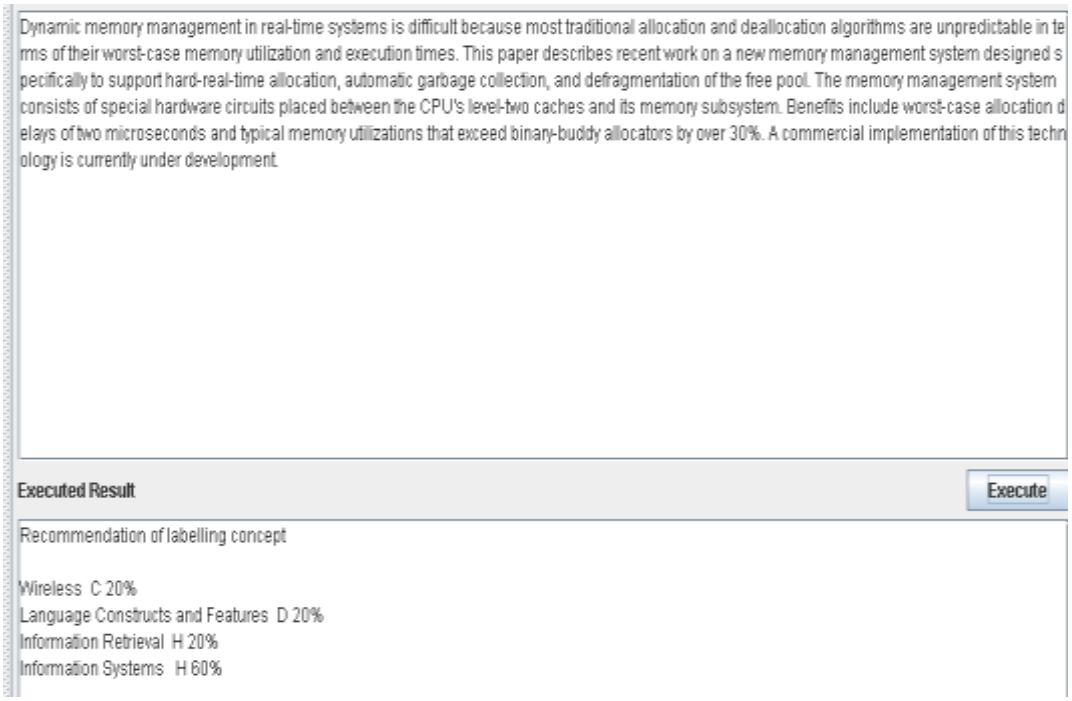

Figure 4: The system of labelling subjects automatically in documents.

In the Fig.4, the Executed Result window shows the all of the labeling concepts of a document. The percent number is calculated based on the total of the words belong to the same category.

4.1 Evaluation based on the Precision measure

The correctness of our proposed approach was calculated by using the precision measure as below:

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

Where C_i represents a subject and correct, wrong represent the number of correct, wrong, respectively.

We implement several experiments for studying the efficiency of our proposed approach. We select two data sets for testing. The first data set contains the header of papers in ACM Digital Library. The second data set are IEEE, Springer papers. The subjects of the papers include Software, Database System, Hardware, Computer Application, Information Systems and Artificial Intelligent. Each data set contains 40 papers. The experimental results of the first data set are shown in Fig. 5.

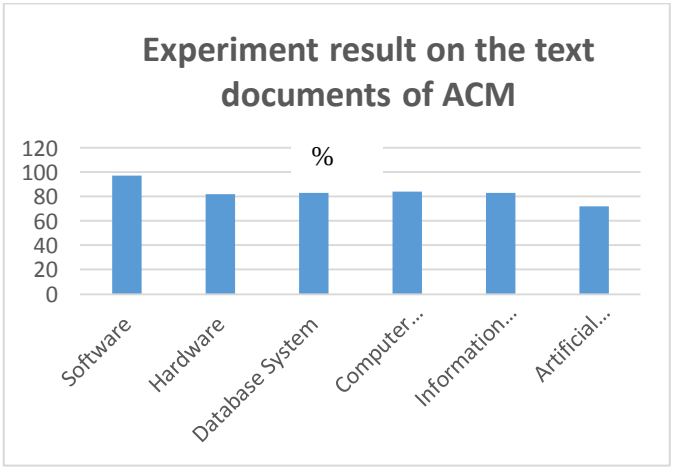

Figure 5: Experimental results of Subject labeling of text documents of ACM Digital Library

The Fig. 5 shows that our proposed system has the best result on the papers with the subject of Information Systems since we get a lot of main keywords when processing the papers in this subject.

With the second dataset, the tests are implemented to the different parts of paper including abstract, introduction, and conclusion to compare the correctness when labelling subjects of papers. The experimental results are shown in Fig. 6.

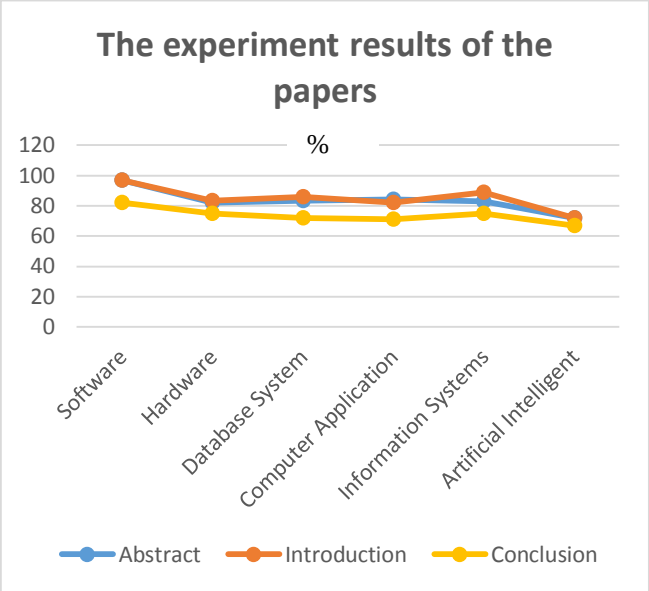

Figure 6: Experiment results on the text documents of IEEE, Springer

According to the Fig. 6, the experimental results are the best on the abstract and the introduction parts of the papers since the main keywords usually belong to both of them. However, the time of implementation of the abstract part is faster than the introduction part since the content of abstract part is shorter than the introduction part.

4.2 Comparative approach

We use Latent Dirichlet Allocation (LDA) algorithm defined by Blei [14] for a comparative approach. We select a collection of documents and use LDA to extract the keywords for identifying subjects from the text documents. In order to compare, we used the same corpus. The comparative results are shown as

Table 1: The Comparative results

Subjects	The proposed Approach	LDA
Software	0.97	0.72
Hardware	0.82	0.67
Database System	0.83	0.52
Computer Application	0.84	0.41
Information Systems	0.83	0.36
Artificial Intelligence	0.72	0.34

The scores in table 1 prove that our proposed system outperforms the keyword selection of LDA algorithm [14]. We believe that the use of Natural Language Processing in our research is a strong point comparing with the bag of words in

5 EXPERIMENTAL RESULT AND DISCUSSION

Our experiment tried to label automatically the subjects in documents. Our proposed approach has three steps: the first step is to construct the domain specific ontology and save to Neo4j Graph database; the second is to extract the keywords from documents; and the last step is to match these keywords to the domain specific ontology to label subjects in documents. We combine the algorithms of Natural Language Processing with domain specific ontology and SDP tool in order to solve a proposed approach. The experiments are implemented to the IEEE, Springer papers and the papers of ACM Digital Library and the. The experimental results are evaluated by the precision measure and are comparative to LDA algorithm. Results generated by such experiments show that the proposed algorithm yields performance respectably. In the future, we will focus on syntactic and semantic LDA to improve the quality of automatic subject labeling.

REFERENCES

[1] R. O. DUDA. Pattern Classification and Science Analysis, New York: Wiley, 1973.
[2] S. H. Wu et al. Assessment on Character-based Chinese News, *Computational Linguistics & Chinese*, vol. 3, no. 2, 1998.
[3] The Stanford Natural Language Processing Group," [Online]. Available: http://nlp.stanford.edu/software/lex-parser.shtml.
[4] P. G. Ipeirotis et al. Classifying and Searching Hidden-Web Text Databases, Columbia University, 2004.
[5] C. D. L. AlSumait. Text Clustering with Local Semantic Kernels, *in Survey of Text Mining II, Clustering, Classification, and Retrieval, Springer*, 2008, pp. 87 - 107.
[6] F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, vol. 34, no. 1, pp. 1 - 47, 2002.
[7] G. D. Salton et al. Idiom Token Classification using Sentential Distributed Semantics, *in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August, 2016.
[8] Tringuyen, PhucDo. Topic Discovery Using Frequent Subgraph Mining Approach, *in Proceedings of the Fourth International, Conference On Computational Science and Technology (ICCST 2017*), Kuala Lumpur, Malaysia, 2017
[9] N. Aletras, M. Stevenson. Labelling Topics using Unsupervised Graph-based Methods, *in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, 2014
[10] R. N. PakpahanI, M. Novitasari. Graceful labeling for some supercaterpillar graphs, *in Proceedings of the 2nd International Symposium on Current Progress in Mathematics and Sciences 2016, (ISCPMS 2016)*, 2016
[11] X. Wang. Understanding Evolution of Research Themes: a Probabilistic Generative Model for Citations, *in The 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Chicago, USA, 2013.
[12] ACM. [Online]. Available: http://www.acm.org/about/class/ccs98-html.
[13] OpenNLP. [Online]. Available: https://opennlp.apache.org/.
[14] D. M. Blei et al. Latent Dirichlet Allocation, *Journal of machine Learning research*, pp. 993 - 1022, 2003.

Ngày nhận bài:13/05/2019
Ngày chấp nhận đăng:10/06/2019