International Review of Research in Open and Distributed Learning

A Semi-Automatic Approach to Construct Vietnamese Ontology from Online Text

Bao-An Nguyen and Don-Lin Yang

Volume 13, Number 5, December 2012

Special Issue: Technology-Enhanced Information Retrieval for Online Learning

URI: https://id.erudit.org/iderudit/1066991ar
DOI: https://doi.org/10.19173/irrodl.v13i5.1250

See table of contents

Publisher(s)

Athabasca University Press (AU Press)

ISSN

1492-3831 (digital)

Explore this journal

Cite this article

Nguyen, B.-A. & Yang, D.-L. (2012). A Semi-Automatic Approach to Construct Vietnamese Ontology from Online Text. *International Review of Research in Open and Distributed Learning*, *13*(5), 148–172. https://doi.org/10.19173/irrodl.v13i5.1250

Article abstract

An ontology is an effective formal representation of knowledge used commonly in artificial intelligence, semantic web, software engineering, and information retrieval. In open and distance learning, ontologies are used as knowledge bases for e-learning supplements, educational recommenders, and question answering systems that support students with much needed resources. In such systems, ontology construction is one of the most important phases. Since there are abundant documents on the Internet, useful learning materials can be acquired openly with the use of an ontology. However, due to the lack of system support for ontology construction, it is difficult to construct self-instructional materials for Vietnamese people. In general, the cost of manual acquisition of ontologies from domain documents and expert knowledge is too high. Therefore, we present a support system for Vietnamese ontology construction using pattern-based mechanisms to discover Vietnamese concepts and conceptual relations from Vietnamese text documents. In this system, we use the combination of statistics-based, data mining, and Vietnamese natural language processing methods to develop concept and conceptual relation extraction algorithms to discover knowledge from Vietnamese text documents. From the experiments, we show that our approach provides a feasible solution to build Vietnamese ontologies used for supporting systems in education.

IRR
ODL

THE INTERNATIONAL
REVIEW OF RESEARCH IN
OPEN AND DISTANCE LEARNING

A Semi-Automatic Approach to Construct Vietnamese Ontology from Online Text

Bao-An Nguyen
Tra Vinh University, Viet Nam
Don-Lin Yang
Feng Chia University, Taiwan

Abstract

An ontology is an effective formal representation of knowledge used commonly in artificial intelligence, semantic web, software engineering, and information retrieval. In open and distance learning, ontologies are used as knowledge bases for e-learning supplements, educational recommenders, and question answering systems that support students with much needed resources. In such systems, ontology construction is one of the most important phases. Since there are abundant documents on the Internet, useful learning materials can be acquired openly with the use of an ontology. However, due to the lack of system support for ontology construction, it is difficult to construct self-instructional materials for Vietnamese people. In general, the cost of manual acquisition of ontologies from domain documents and expert knowledge is too high. Therefore, we present a support system for Vietnamese ontology construction using pattern-based mechanisms to discover Vietnamese concepts and conceptual relations from Vietnamese text documents. In this system, we use the combination of statistics-based, data mining, and Vietnamese natural language processing methods to develop concept and conceptual relation extraction algorithms to discover knowledge from Vietnamese text documents. From the experiments, we show that our approach provides a feasible solution to build Vietnamese ontologies used for supporting systems in education.

Keywords: Ontology; concept discovery; conceptual relation; text mining; lexical pattern; natural language processing

Introduction

An ontology is a formal, explicit specification of a shared conceptualization (Noy & McGuinness, 2001). Ontologies that belong to a specific domain are constructed from knowledge about domain concepts, their properties and instances, and the conceptual relations between them. In recent years, many semantic-based intelligent systems such as searching systems, recommender systems, and question answering systems have used ontologies as their knowledge bases. In education and e-learning, many researchers have built learning support systems that take advantage of ontologies. Li and Rui (2005) proposed a novel way to organize learning content into small "atomic" units called learning objects and systemized them together with their ontology into knowledge bases used for a recommendation mechanism. Ana et al. (2009) built a recommender system in which a domain ontological model is presented as support to Venezuelan students' decision making for study opportunities. Saman et al. (2012) developed a knowledge-based and personalized e-learning recommendation system based on ontology to improve the quality of an e-learning system. These ontologies were constructed manually by using expert knowledge obtained from many resources and documents.

Due to their availability and abundance, text documents are one of the most popular types of knowledge sources for experts to construct their domain ontologies. Many research studies have been done on text mining and ontology construction using concept/entity extraction and conceptual relation discovery. Text mining is a subsection of data mining which could discover useful and hidden patterns or information from text. It has been used widely in many fields such as information retrieval, linguistics, knowledge engineering, and bioinformatics. Among text mining tasks, concept/entity extraction (concept mining) is applied extensively in many applications such as document summarization, question answer systems, taxonomy construction, and ontology construction. Most concept mining methods are based on linguistic rules, statistics, or a combination of both (Zhou & Wang, 2010). Other research studies also use frequent pattern mining and association rule mining for discovering concepts and conceptual relations from text (Maedche & Staab, 2000, 2001; Zhou & Wang, 2010; Chen, Zhang, Li, & Li, 2005).

Ontology construction requires efforts to uncover and organize relevant domain knowledge in a suitable structure according to the purpose of the ontology's usage. This can be done manually or by using automatic or semi-automatic methods, in which learning methods and knowledge engineering are applied to extract concepts and conceptual relations from domain documents.

In manual construction approaches, domain experts play an essential role. Many tasks are done by these experts: covering domain terms (concepts), defining classes and class hierarchies, creating class slots (properties), filling slot values, and generating instances (Noy & McGuinness, 2001). Since every task is executed and verified by humans, the constructed ontologies tend to have a high level of accurate, reasonable, and adequate context. However, it requires a large amount of human effort and time, especially for large-scale domains such as the semantic web.

By contrast, fully automatic ontology construction methods try to learn and extract knowledge from domain documents without human supervision. For instance, Christian and Alfonso introduced an automatic ontology construction using bibliographic information (Blaschke & Valencia, 2002). Maedche and Staab presented an ontology learning framework from the semantic web through ontology import, extraction, pruning, refinement, and evaluation mechanisms (Maedche & Staab, 2001). Lee et al. (2007) presented an episode-based ontology construction mechanism from text documents and used a fuzzy inference mechanism for Chinese text ontology learning. Unfortunately, these methods are usually difficult to implement and limited in specific domains since many domain-specific decisions must be made to adequately specify the domain of interest (Jaimes & Smith, 2003). In addition, learning the knowledge base from unconstructed data is cognitive work that needs many supporting studies, and the concept hierarchy acquisition is one of the largest challenges.

Summarizing the above approaches, a semi-automatic ontology construction method is the most common approach in which information extraction techniques are used under the supervision of humans. Such methods include the learning modules to extract concepts and conceptual relations from domain documents. They require expert knowledge to verify the obtained information and decide which information should be included in the ontology. In English, many frameworks and plugins have been built to help users construct ontologies semi-automatically. For example, TextToOnto proposed by Maedche and Staab (2000) used generalized association rules to find out the co-occurrences between items and relations between them. OntoLT is a Protégé plugin that extracts concepts and relations from annotated documents for ontology construction.

Typically, taxonomy is needed in ontology acquisition tasks to construct the concept hierarchical structure of the ontology. In English, taxonomy-based approaches often use WordNet as a super taxonomy to determine the conceptual relations between concepts. In Chinese, HowNet has been used with the same role. When taxonomies are not available (or for other reasons), a nontaxonomy approach is considered using learning algorithms (e.g., Lee, Kao, Kuo, & Wang, 2007; Maedche & Staab, 2001; Blaschke & Valencia, 2002).

To extract candidate terms, the well-known statistical measurement TF-IDF can be used (Lee, Kao, Kuo, & Wang, 2007; Zheng, Dou, Wu & Li, 2007). Association rules or frequent patterns are mined to discover the co-occurrences and semantic relations between terms (Maedche & Staab, 2000, 2001; Zheng, Dou, Wu, & Li, 2007). Linguistic rules were also used in research (e.g., Zhou & Wang, 2010; Chen, Zhang, Li, & Li, 2005) in which predefined lexical patterns were used to extract candidate concepts by a bootstrapping mechanism. In Vietnamese, Nguyen and Phan (2009) proposed a hybrid approach which combines lexical rule-based and ontology-based methods to extract key terms and phrases from Vietnamese text.

In this research, we propose a semi-automatic approach to extract concepts and conceptual relations from Vietnamese text documents by using a combination of text mining techniques and statistics-based methods. Concepts will be discovered not only based on the TF-

IDF measure but also by applying lexical patterns and association rules mining. The reason to use a combination of various techniques is shown in Table 1. We also aim to compare the performance of various concept discovery algorithms and the combination of them to determine the best extracting approach for Vietnamese text documents.

Table 1

Comparison of Used Techniques

Techniques	Based on	Weaknesses
Statistics-based	Importance of terms – TFIDF Co-occurrences of terms in documents Association rules	Easily affected by noises Does not consider semantic aspect of documents
Lexical rule-based	Predefined linguistic rules	Hard to build a complete rule set covering all language cases
Combination of statistics-based and lexical rule-based	Taking into account both statistics and linguistic characteristics of terms.	

Proposed Method

In this section, we present our proposed system, called Vietnamese Text To Ontology, or ViText2Onto, along with learning techniques to discover concepts and conceptual relations from Vietnamese text documents. Our contribution can be stated as follows: Given a set of Vietnamese text documents in a specific domain, our system can support the user to construct an ontology using a semi-automatic approach. The resulting ontology contains concepts and instances organized in an appropriate hierarchy.

System Architecture

To construct an ontology, domain knowledge must be discovered and organized in a conceptual hierarchical structure. ViText2Onto employs a semi-automatic approach where discovery methods are used in combination with human supervision. From this perspective, an interactive mechanism is established between the system and users where the construction process is iterative and cyclic. After each iteration, the conceptual hierarchy is extended and verified by users such that the users can incrementally discover more concepts and relations based on the assessed concepts. The system architecture is shown in Figure 1, which includes the following components.

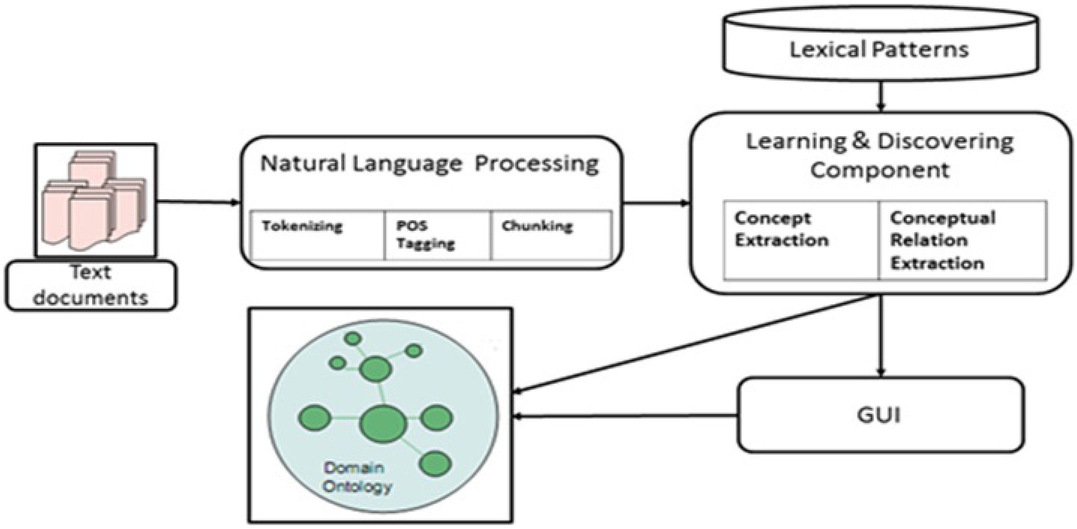


Figure 1. System architecture.

(1) A Vietnamese natural language processing module is used to make the Vietnamese text documents ready for extraction algorithms. It is a set of Vietnamese processing tools to perform tokenizing, part of speech tagging (POSTagging), and chunking. The output of this module is annotated documents being stored in text files. A small convertor is created to read these files and converts them into a compatible format that can be used by GATE.

(2) A learning and discovering component is used for extracting concepts and conceptual relations from annotated documents. We use various learning and discovering algorithms, including pattern-based, statistics-based, and association-based approaches. To implement pattern-based learning, we use JAPE (Java Annotation Pattern Engine) which is an element of the GATE framework. JAPE provides finite state transduction over annotations that let us extract predefined patterns based on rules written in a specific grammar.

(3) Lexical patterns contain lexical rules written using JAPE grammar which are used for pattern-based learning. They are constructed based on Vietnamese syntactic rules. Applying these rules on the corpus using JAPE, we can extract concepts and conceptual relations from the matched patterns.

Vietnamese Language Processing

We use Vietnamese language processing tools provided by the project of Building Basic Resources and Tools for Vietnamese Language and Speech Processing (VLSP) for preprocessing Vietnamese textual corpora. The processing components have the following features.

Vietnamese word tokenization.

Due to the characteristics of Vietnamese, a word might contain only one individual word (one morpheme) or a compound of two or more individual words (many morphemes). This tool identifies words and tokenizes sentences into separate tokens. Resulting tokenized documents are used for further analysis tasks.

Vietnamese part of speech tagging (POS tagging).

As discovered concepts are mostly nouns, proper nouns, and noun phrases, POS tags play an essential role in both syntactic- and semantic-based learning for ontology acquisition. The POS tagger uses tokenized documents as input and assigns a POS label for all tokens.

Vietnamese chunking.

Chunking is used to divide each sentence into frames containing one or more words where each frame has a specific grammatical role in the sentence. Segmenting sentences into chunks helps determine grammatical roles of elements in sentences; hence, it is useful for learning and extracting. In our extraction rule sets, noun phrases and verb phrases are used as majority units of the patterns. Chunking frames are also used in association rule mining where phrases are used as input.

Stop words removing.

There are many words having high frequencies of occurrence in Vietnamese text while they contribute very little to the subject of sentences. To avoid noises caused by these words, we apply stop word removing when computing the TF-IDF of terms.

Learning Algorithms

In this research, the purpose of the ontology learning task is to discover concepts and conceptual relations. We use a combination of lexical pattern-based, frequent sequence-based, and statistics-based methods to overcome some drawbacks in each of the individual methods. Figure 2 shows the model of learning and discovery components.

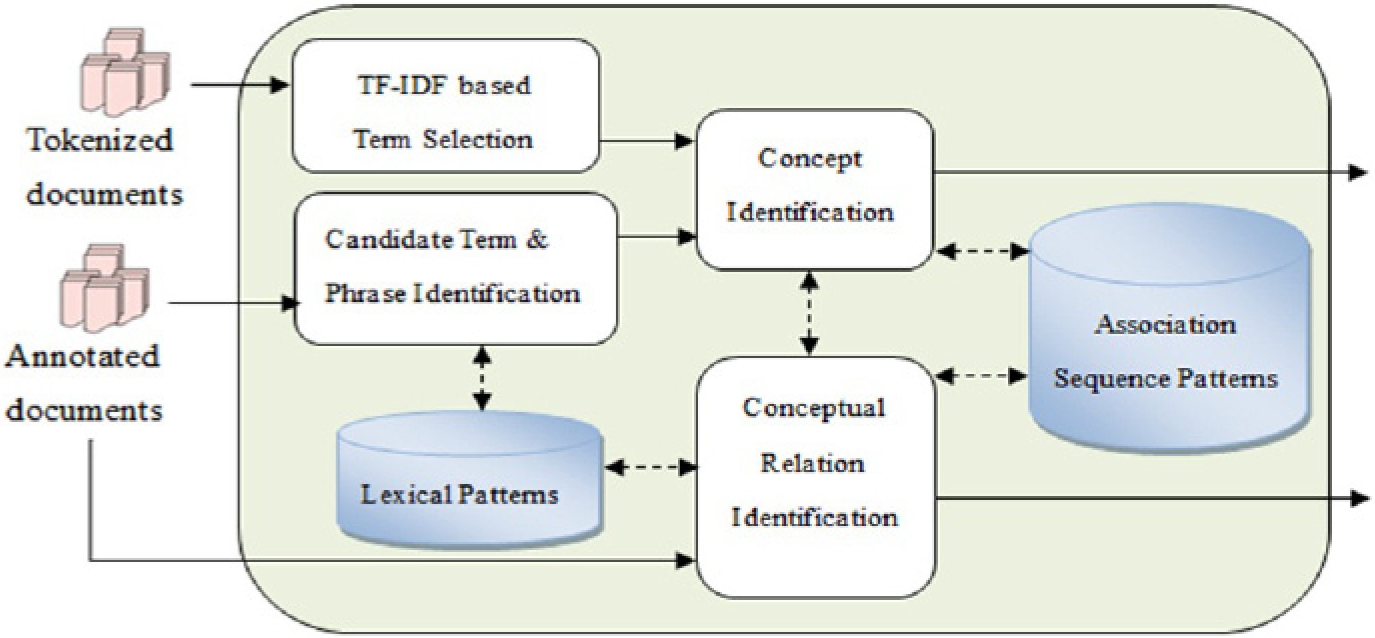

Tokenized documents
Annotated documents
TF-IDF based Term Selection
Candidate Term & Phrase Identification
Concept Identification
Conceptual Relation Identification
Lexical Patterns
Association Sequence Patterns

Figure 2. Learning model.

Overall Construction Process

The overall ontology construction process in the proposed system is illustrated in Figure 3. Initially, a user prepares an input corpus. Then the text files are put into the Vietnamese natural language processing module for tokenizing, POS tagging, and chunking. Processed documents are converted into the specific document format of GATE using our own convertor. These Vietnamese text documents are ready for the learning process.

Firstly, candidate concepts are extracted and presented to our user interface. Note that users can specify the TF-IDF threshold and minimum support to the extraction algorithm. Concepts will be sorted in descending order by the TF-IDF score to help users select important concepts. At this step, users may only have a small number of concepts to select as seed concepts. Then a prefix-based discovery algorithm will be run to generate concept trees. Again, users will select relevant concepts as input to the relation extraction phrase. In this phrase, pattern-based and association rule-based relation extraction methods are executed. Matched patterns and association rules satisfying the minimum support and confidence are sent to the user interface in the form of a relation between a pair of concepts.

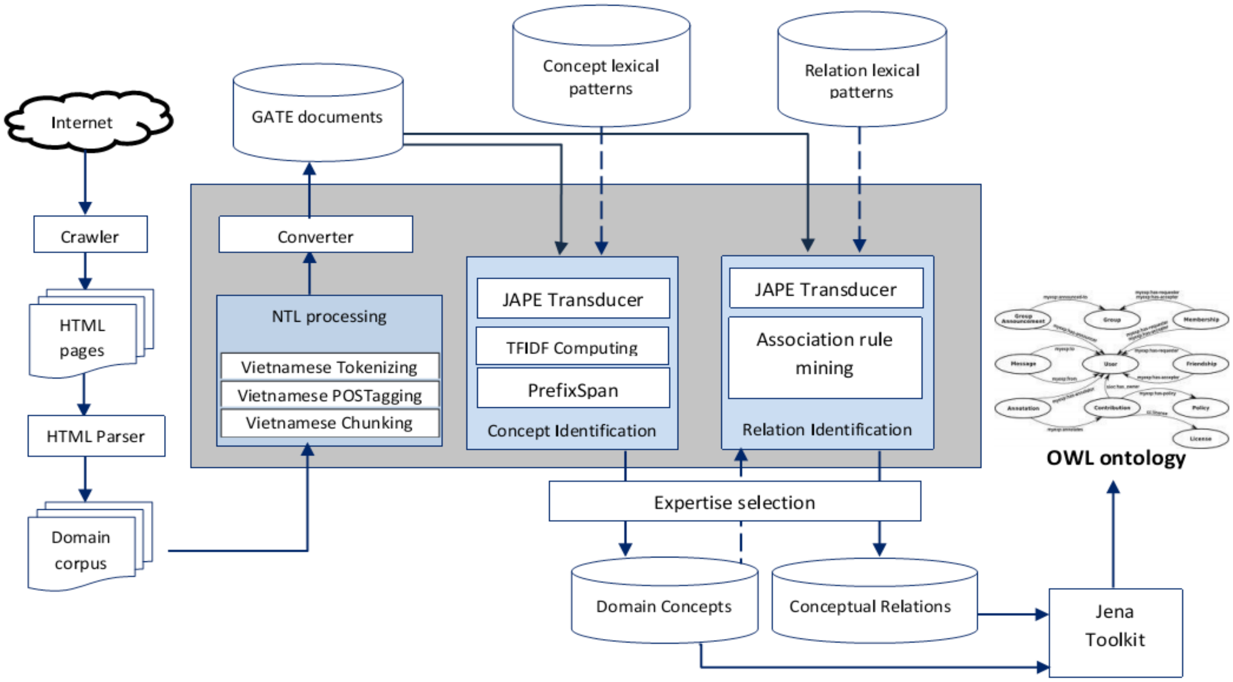


Figure 3. Overall construction process.

Users will make the final decision to select relevant concepts and conceptual relations. The selected result is exported to the ontology in OWL (web ontology language) by using the Jena toolkit.

In this process, expert knowledge is embedded into the ontology in two steps where concepts and relations are presented to users for selection. The resulting ontology contains all selected concepts and relations that can be edited easily using some ontology editing tools such as Protégé to meet user expectations.

Concept Discovery

TF-IDF-based candidate term selection.

The well-known term weight TF-IDF is used to measure the importance of individual terms contributing to documents. Important terms, such as terms having higher TF-IDF scores, will be selected based on a user-defined threshold. TF-IDF of term T_i in a document d_j is computed by the following equation:

$$tfidf(T_i,\ d_j) = tf(T_i, d_j) \times \log(\frac{|D|}{|d: T_i \in d|})$$

where $tf(T_i, d_i) = \frac{n_{ij}}{\sum_k n_{kj}} tf(T_i, d_i) = \frac{n_{ij}}{\sum_k n_{kj}}$ is the term frequency of a term T_i in a document d_j, n_{ij} is the number of occurrences for T_i in d_i, the dominator is the size of d_j, and $\log(\frac{|D|}{|d: T_i \in d|})\log(\frac{|D|}{|d: T_i \in d|})$ is the inverse document frequency of T_i.

We construct a set S_{TFIDF} of candidate terms whose TF-IDF values exceed the threshold δ: $S_{TFIDF} = \{t_i | TFIDF(t_i) \geq \delta\}S_{TFIDF} = \{t_i | TFIDF(t_i) \geq \delta\}$

where t_i refers to one term in the documents.

Lexical pattern-based candidate term and phrase selection.

TF-IDF can be used to select important individual terms. However, a Vietnamese concept often consists of multiple terms. To discover multiple-term concepts, a lexical pattern-based approach is used. We built a set of lexical rules based on JAPE grammar to discover proper nouns and noun phrases. Input sentences are processed by the finite state transducer provided by JAPE, in which matched patterns will be discovered and annotated.

According to Vietnamese grammatical characteristics, the following patterns can be used for noun phrase and proper noun identification:

Noun+ Noun

Noun* (Noun | ProperNoun) (Adj | Noun)*

Noun+ Verb (Adj | Noun)+

Noun* ProperNoun+ Number

where "|" means or, "+" means one or more occurrences, and "*" means zero or more occurrences. The last pattern listed above is used for identifying proper nouns which end with a number, such as Windows Mobile 6.0 and iPhone 4.

Here is an example of a noun phrase: [Công ty]$_{N}$[trách nhiệm]$_{N}$[hữu hạn]$_{Adj}$ [VinaCom]$_{NP}$,(means [Limited]$_{Adj}$ [Company]$_{N}$ [VinaCom]$_{NP}$).

Applying these patterns to the input documents using GATE, we obtain a list of candidate noun phrases and proper nouns S_p.

Sequential pattern mining.

Lexical pattern-based learning is practical and appropriate for deep knowledge discovery, but the competence of its results depends strongly on the completeness of the set of lexical rules. To overcome this weakness, we adopt the advantages of frequent sequence mining in natural language. Based on the assumption that a concept might be a phrase or a part of a phrase in which element words usually appear in fixed orders, we consider a concept as a sequence of ordered words. Concepts might be obtained by mining frequent sequential patterns from the documents where each noun phrase is considered as a transaction.

In our research, we use segmented sentences with chunking labels as input for sequential pattern mining. As the input sentences are segmented into frames which have specific grammatical roles, a concept often belongs to only one frame. We consider each frame as a sequence and each word as an item. By mining frequent sequences we can obtain word sequences that frequently occur together in a frame, and, hence, they can become candidate concepts. For example:

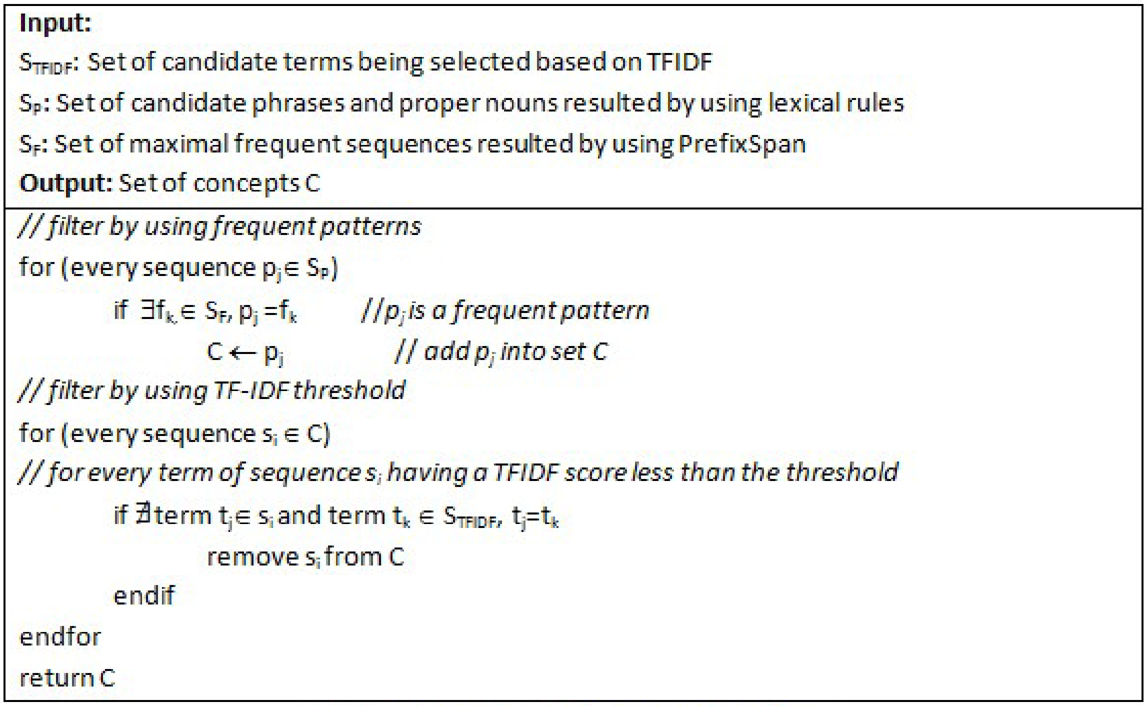
- **Input:** Điện thoại Iphone 4 màu trắng chưa được sản xuất
- **Meaning:** White Iphone 4 has not been produced yet
- **Chunks:** [*Điện thoại Iphone 4 màu trắng*]$_{NP}$ [*chưa được sản xuất*]$_{VP}$

Assuming that one of the frequent sequences is “Iphone 4,” we can see the candidate concept completely belongs to the chunk [Điện thoại Iphone 4 màu trắng]$_{NP}$.

After the mining stage, we only use maximal frequent sequences as candidate phrases. The set of frequent sequences is denoted by S_F.

Concept identification.

By executing the above candidate concept discovery processes, we develop a list of candidate terms and phrases for concept identification. We need a filter mechanism to select relevant concepts for the ontology. The filter algorithm aims to merge three sets of candidates into a unique set of concepts, in which the candidates with lower TF-IDF scores are removed. The steps of concept identification algorithm are shown in Figure 4.



Input:
S_{TFIDF}: Set of candidate terms being selected based on TFIDF
S_P: Set of candidate phrases and proper nouns resulted by using lexical rules
S_F: Set of maximal frequent sequences resulted by using PrefixSpan
Output: Set of concepts C

```
// filter by using frequent patterns
for (every sequence $p_j \in S_P$)
        if $\exists f_k \in S_F, p_j = f_k$        // $p_j$ is a frequent pattern
                $C \leftarrow p_j$               // add $p_j$ into set C
// filter by using TF-IDF threshold
for (every sequence $s_i \in C$)
// for every term of sequence $s_i$ having a TFIDF score less than the threshold
        if $\nexists$ term $t_j \in s_i$ and term $t_k \in S_{TFIDF}$, $t_j = t_k$
                remove $s_i$ from C
        endif
endfor
return C
```

Figure 4. Concept identification algorithm.

Conceptual Relation Discovery

In this phase, we use the combination of pattern-based and association-based learning to discover the relation between concepts. Using lexical rules in conceptual relation discovery has high reliability since lexical rules were predefined by humans based on linguistic rules; however, predefined grammatical rules may not cover all cases of language usage. We use the advantage of association rule mining to overcome this weakness, in which relations between concepts are mined without considering the semantic aspect of sentences.

Lexical pattern-based conceptual relation discovery.

In this process, we take into account the semantic relations between elements in sentences, as used by Nguyen and Phan (2009). According to Vietnamese grammatical characteristics, some rules of relations between nouns or noun phrases are as follows:

- Rule 1: {Noun phrase A} “là một” {Noun phrase B} --> A is a B
- Rule 2: {Noun phrase A} {Proper noun B} --> B is an instance of A
- Rule 3: {Noun phrase A} “có” {Noun phrase B} --> A has a B
- Rule 4: {Noun phrase A} “của” {Noun phrase B} --> B has an A
- Rule 5: {Noun phrase A} “thuộc” {Noun phrase B} --> A is a subclass of B
- Rule 6: {Noun phrase A} “bao gồm” {Noun phrase B},{Noun phrase C} --> B and C belong to A
- Rule 7: {Noun phrase A} (“và” | “hoặc”) {Noun phrase B} --> A (and | or) B

Based on these rules, we build a set of extraction rules using JAPE grammar. When the matching process is invoked, matched concept pairs and the relations between them will be discovered. In this research, we focus on finding subsumption relations and instances of concepts. The set of lexical rules contains many language usage cases that imply isA and hasA relations. Building a complete set of rules is not feasible; however, the rule sets can be enriched in further study.

Heuristic for Concept and Conceptual Relation Discovery

Context implication.

If A is a concept and B appears with A in the context {A} (“và” | “hoặc”) {B}, we can infer that 1) B is also a concept and 2) A and B have the same level of abstraction. For example:

- Điện thoại HTC Hero và Motorola Milestone đều được cài đặt hệ điều hành Android 2.1.
- Both HTC Hero and Motorola Milestone are installed with the operating system Android 2.1.

In this context, if we already know HTC Hero is a concept being recognized as a kind of mobile phone, we can easily infer that Motorola Milestone is also a kind of mobile phone.

Incremental learning.

Obviously, each domain has some cornerstone concepts, said seed concepts, which occur in many documents within the corpus. The learning phase should start from these seed concepts to discover concepts at a lower level of abstraction and repeat the process in an incremental manner.

For example, in the domain of mobile phone, we can start by some commonly used concepts such as mobile phone, keyboard, screen, and operating system. Based on Vietnamese characteristics, a child-concept often begins with its parent-concepts. We define the prefix-based concept and conceptual relation discovery algorithm below:

Given a concept C_{seed} as the seed concept selected by a user, if a concept C_i begins with C_{seed}, C_i might also be a relevant concept that should be selected by the user and C_i is a child-concept of C_{seed} (in the ontology, C_i becomes a subclass of the class C_{seed}).

By executing this algorithm on seed concepts, we can incrementally obtain a tree of concepts. This tree can be used as a part of the concept hierarchy for the ontology containing “isA” relations between child-concepts and its parents. An example of using this approach is shown in Table 2.

Table 2

An Example of Child-Concepts Generalization using Seed Concepts and Prefix-Based Concept Discovery Algorithm

Seed concepts	1_child concepts	2_child concepts	Meaning
màn hình			screen
	màn hình cảm ứng		touch screen
		màn hình cảm ứng điện trở	resistive touch sceen
		màn hình cảm ứng điện dung	capacitive touch screen
bàn phím			keyboard
	bàn phím QWERTY		QWERTY keyboard
	bàn phím cảm ứng		touch keyboard

Learning from instance.

Typically, a class name rarely co-occurs with its subclasses or its properties in a sentence. Instead, instances of that class usually appear together with its related concepts. For example:

- HTC Hero được trang bị màn hình cảm ứng điện dung 4.3 inchesvà bộ nhớ trong 1.5GB.
- HTC Hero is equipped with a capacitive touch screen 4.3 inches and internal memory 1.5GB.

If we already know the class of the instance, we generalize the abstraction of the instance by replacing it with its class and obtain the relation between the class and discovered concepts. In this example, mobile phone will have a "hasA" relation with screen and internal memory.

Association Rule-Based Conceptual Relation Discovery

Frequent sequential pattern mining can help in cases when lexical patterns cannot be applied. We use association rule mining to find hidden (anonymous) relations between concepts by taking into account their co-occurrence in contexts, both on a sentence level and document level.

An association rule reflects an implication between its two sides. Let T = {t_i | i = 1, 2,..., n} denotes a set of transactions, where each transaction is a list of items. I = {i_1, i_2, ..., i_m} denotes a list of items. An association rule of "A *implies* B" states that A associates with B, where A and B belong to I, and the intersection of A and B is empty. A rule "A *implies* B" indicates that the appearance of A is followed by the appearance of B with an acceptable probability. The reliability of a rule is expressed by two measures *support* and *confidence*.

$$support\left(A \Rightarrow B\right) = \frac{|\{t_i | A \cup B \subseteq t_i\}|}{n}$$

$$confidence\left(A \Rightarrow B\right) = \frac{|\{t_i | A \cup B \subseteq t_i\}|}{|\{t_i | A \subseteq t_i\}|}$$

That is, support is the probability to see both A and B appear in the same transaction while confidence is the probability to see the consequence B when the antecedent A appears in a transaction.

At the sentence level, we aim to find concept pairs that often appear together in a sentence. We consider each sentence as a transaction where each term is an item. If a concept is discovered in a sentence, terms that belong to the concept will be merged into one item. The result of the association rule mining stage will be in a pair of concepts (C_A, C_B) with support and confidence exceeding predefined thresholds. Nevertheless, we also want to find the verb that connects these two concepts and the result is in the form of (C_A, C_B, $Verb_{con}$).

At the document level, we consider each sentence as a transaction where items are concepts that appear in it. The assumption under this mining stage is that two concepts occurring in different sentences may have a relation between them. The results are concept pairs of (C_X, C_Y) that may not co-occur in one sentence. Results of association rule mining, both at the sentence level and document level, are presented to users as pairs of concepts. Users will make their own judgment on the final selection.

Experimental Results

To evaluate the performance of our work, we built a real Vietnamese ontology for the mobile phone domain as a base for comparison, which is called reference ontology, O_R. It was manually built by using the source documents from the online technical specifications of various smartphones like iPhone, HTC Hero, Motorola Milestone, Samsung Galaxy, and so on.

Applying our semi-automatic approach to build an ontology, called computer learned ontology O_C, we used Vietnamese news on many kinds of mobile phones from 2009 to 2010 as the corpus. The corpus includes about 500 news articles from Vietnamese Web sites covering such topics as the arrival of new phones, comparisons of various phones, sharing phone usage experiences, symptoms of phone problems, and their advantages and disadvantages. To obtain the input corpus, we used a crawler to download the entire subfolders of three source Web sites. Web pages published before 2009 were not included. Then we manually selected HTML pages that contain well-known phone brands such as iPhone, Nokia, HTC, Samsung, Motorola, Acer, and Sony Ericsson. The final collection of the HTML pages was used as input documents.

Firstly, we used HTMLParser to extract contents from the HTML files to generate text files. The text files were tokenized, POS tagged, and segmented using Vietnamese processing tools: vnToolkit and VLSP tools. To make the Vietnamese text documents suitable for concept extraction with GATE, we developed a convertor to convert them into annotated documents that can be used by GATE. The annotation sets contain POS labels and chunk labels of tokens.

We constructed a set of Vietnamese lexical rules using JAPE grammar for pattern-based discovery. PrefixSpan is used for mining frequent sequential patterns and association rule-based discovery. The pattern concept extraction is executed by GATE transducer based on a set of lexical patterns. Results of this stage are recorded as annotations in the annotation set of the documents to be used as input for conceptual relation extraction. The final results of concept set and conceptual relation set are proposed to the users for manual selection. Selected objects and relations are exported to OWL model by Jena toolkit.

Finally, to evaluate the performance of concept discovery algorithms, we compute the term precision and term recall scores on the comparison between the ontologies O_C and O_R. To evaluate how well relations were learned, we used the measures of taxonomic precision and taxonomic recall. These scores were acquired by comparing the concept hierarchies of the two ontologies based on their position of common concepts.

Evaluation of Concept Extraction

We adopted the most commonly used measurements for information retrieval, term precision and term recall, to measure the performance of concept extraction methods. These measures are computed based on the overlap between the set of concepts in the reference ontology O_R and the computer learned ontology O_C. Let ST be the term set of the reference ontology and T be the term set discovered by our method. We have:

$$Term\ Precision = \frac{|ST \cap T|}{|T|}$$

$$Term\ Recall = \frac{|ST \cap T|}{|ST|}$$

We also use F_b-measure with the same weights of precision and recall to measure the overall performance, where F_b-measure is computed by:

$$F_\beta = \frac{(1+\beta^2).Precision.Recall}{(\beta^2.Precision + Recall)}$$

Here b is recall weight against precision weight. Since the purpose of our system is extracting and presenting as many materials as possible to users, we select F_2 which weights recall twice as much as precision due to the requirement on the completeness of the constructed ontology. This weight means that the more extracted concepts satisfying user requirements, the better result we will get.

$$F_2 = \frac{(1+2^2).Precision.Recall}{(2^2.Precision + Recall)}$$

In order to examine the scalability of the system, we tested the extraction performance with various corpus sizes. We divided the set of documents into five subsets as listed in Table 3. There are one large dataset, two small datasets, and two medium datasets.

Table 3

Description of Test Document Sets

	Number of files	Total size	Number of contained concepts
Subset1	195	1.37 MB	1412
Subset2	28	249 KB	489
Subset3	34	214 KB	340
Subset4	62	768 KB	696
Subset5	60	628 KB	746

In our system, we use two parameters TF-IDF threshold b and minimum support m of frequent sequence mining to drive learning algorithms. Originally, when both two parameters are set as zero, only lexical pattern-based concept extraction algorithm is executed. By increasing the TF-IDF threshold value or minimum support score, extracted concepts will be filtered by the corresponding parameter. These adjustments affect the number of extracted concepts. The higher the parameter values are set, the fewer the number of concepts that will be extracted. According to the size of the constructed ontology, users can adjust these parameters to increase or decrease the size of the set of concepts.

Table 4 presents our results in detail, where δ is the TF-IDF threshold, M is the minimum support, T is the number of extracted concepts, STÇT is the number of relevant extracted concepts, P is the precision value, R is the recall value, and F is the F_2-measure value.

Table 4

The Performance of Concept Extraction

	δ	M	T	STÇT	P	R	F	δ	M	T	STÇT	P	R	F
S1	0	1	4888	1394	0.29	0.99	0.67	0.01	1	4844	1384	0.28	0.98	0.65
S2			1270	489	0.40	1.00	0.77			1042	444	0.43	0.91	0.74
S3			819	340	0.42	1.00	0.78			767	322	0.42	0.95	0.76
S4			2005	694	0.35	1.00	0.73			1438	532	0.37	0.77	0.63
S5			1846	746	0.40	1.00	0.77			1647	680	0.41	0.91	0.73
S1	0	2	2782	932	0.34	0.66	0.56	0.01	2	2781	932	0.34	0.66	0.56
S2			696	311	0.45	0.64	0.59			646	304	0.47	0.62	0.58
S3			480	233	0.49	0.69	0.64			455	220	0.48	0.65	0.61
S4			1256	500	0.40	0.72	0.62			978	412	0.42	0.59	0.55
S5			1846	746	0.40	1.00	0.77			1130	529	0.47	0.71	0.64
S1	0	4	1239	492	0.40	0.35	0.36	0.01	4	1238	492	0.39	0.34	0.35
S2			294	151	0.51	0.31	0.34			274	145	0.53	0.30	0.33
S3			241	137	0.57	0.40	0.43			235	132	0.56	0.39	0.42
S4			597	295	0.49	0.42	0.43			507	259	0.51	0.37	0.39
S5			542	290	0.54	0.39	0.41			525	285	0.54	0.38	0.40
S1	0.005	1	4888	1394	0.29	0.99	0.67	0.02	1	3773	1110	0.30	0.79	0.60
S2			1254	489	0.39	1.00	0.76			465	216	0.46	0.44	0.44
S3			809	337	0.42	0.99	0.78			486	200	0.41	0.59	0.54
S4			1859	658	0.35	0.95	0.71			683	266	0.39	0.38	0.38
S5			1768	722	0.41	0.97	0.76			943	401	0.43	0.54	0.51
S1	0.005	2	2782	932	0.34	0.66	0.56	0.02	2	2481	862	0.34	0.61	0.53
S2			686	311	0.46	0.63	0.59			335	186	0.56	0.38	0.41
S3			478	232	0.49	0.68	0.63			289	144	0.50	0.42	0.43
S4			1197	490	0.41	0.70	0.61			494	222	0.45	0.32	0.34
S5			1209	553	0.46	0.74	0.66			728	348	0.48	0.47	0.47
S1	0.005	3	1524	586	0.39	0.42	0.41	0.02	3	1365	541	0.40	0.38	0.38
S2			363	183	0.51	0.37	0.39			198	116	0.59	0.24	0.27
S3			298	166	0.56	0.49	0.50			188	107	0.57	0.31	0.34
S4			720	336	0.47	0.48	0.48			317	166	0.52	0.24	0.27
S5			652	334	0.51	0.45	0.46			406	212	0.52	0.28	0.31

Results of extraction performance show that the lexical pattern based approach can produce high recall but relatively low precision. By increasing the TF-IDF thresholds and minsup values, precision can be improved. Figure 5 shows the scalability of our system with respect to corpus size when δ = 0.005 and M = 2. In our test cases, the system produced best performance when it was tested with a medium size corpus; increasing corpus size can make the extraction performance go down. Overall, after testing with various sets of input documents, the extraction performance was around 60% when appropriate thresholds were specified.

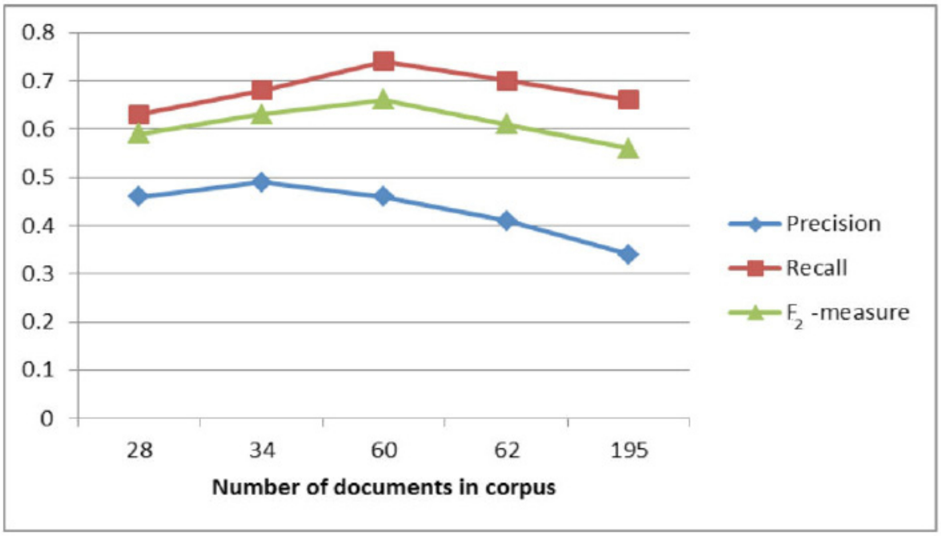


Figure 5. Extraction performance with respect to corpus size.

Evaluation of Conceptual Relation Extraction

As our purpose in the relation extraction step is to find subsumption relations, the lexical rules are built to discover patterns that contain “isA” and “hasA” relations. In the set of relations found by association rule-based extraction, we only used rules that imply “isA” and “hasA” relations. These relations are used to construct the concept hierarchy of the ontology. Some of the top extracted relations are shown in Table 5.

Table 5

Top Extracted Relations

Relations	English meaning
điện_thoại – hasA – màn_hình	phone – hasA – screen
điện_thoại – hasA – bàn_phím	phone – hasA – keyboard
màn_hình – hasA – độ_phân_giải	screen – hasA – resolution
điện thoại– hasA – hệ_điều_hành	phone – hasA – operating_system
Android – isA – hệ_điều_hành	Android – isA – operating_system
màn_hình_cảm_ứng – isA – màn hình	Touch_screen – isA– screen
màn_hình_cảm_ứng_điện_dung – isA – màn_hình_cảm_ứng	Capacitive_touch_screen – isA – touch_screen
bàn_phím_QWERTY – isA –bàn_phím	QWERTY_keyboard – isA – keyboard

Figure 6 is an illustration of some top extracted conceptual relations in the computer learned ontology being translated into English.

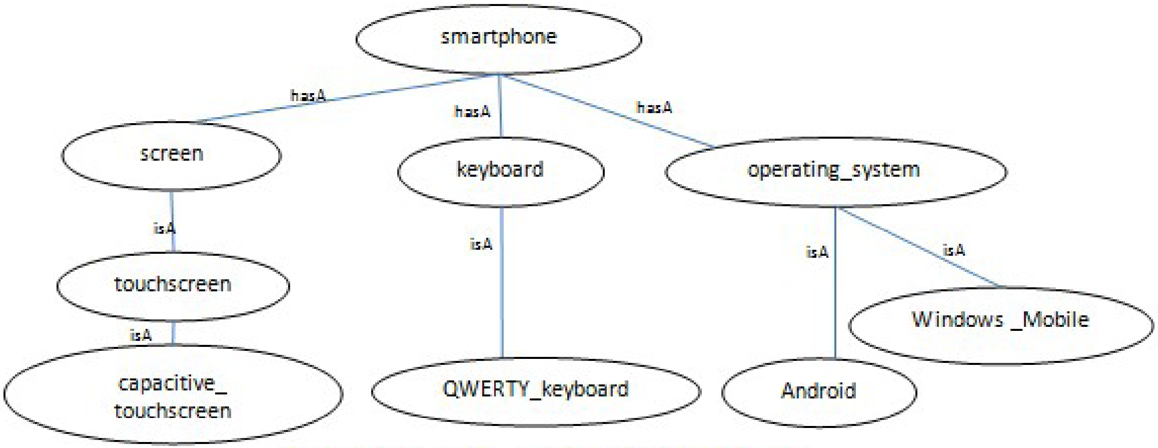


Figure 6. Some learned top level relations.

In order to evaluate how well the concept hierarchy was constructed in the ontology, we used taxonomic precision (TP) and taxonomic recall (TR) as proposed by Dellscaft and Staab (2003), in which the position of a concept in the learned hierarchy is compared with the same concept in the reference hierarchy. TP and TR are computed based on common semantic cotopy (csc) which measures the taxonomic overlap of two ontologies. The common semantic cotopy excludes all concepts which are not also available in the other ontology's concept set. Given a concept c, two ontologies O_1 and O_2, the common semantic cotopy (csc) is defined as follows:

$$\mathrm{csc}(c, O_1, O_2) := \{c_i | c_i \in C_1 \cap C_1 \wedge (c_i <_1 c \vee c <_1 c_i)\}$$

where C_1 and C_2 are two sets of concepts for ontologies O_1 and O_2, respectively.

TP and TR are computed based on common semantic cotopy as follows:

$$TP_{csc}(O_1, O_2) := \frac{1}{|C_1 \cap C_2|} \sum_{c \in C_1 \cap C_2} tp_{csc}(c, c, O_1, O_2)$$

$$TR(O_1, O_2) := TR(O_2, O_1)$$

where $tp_{csc}(c, c, O_1, O_2)$ is a local precision on common semantic cotopy of the concept c and computed by:

$$tp_{csc}(O_1, O_2) = \frac{|\mathrm{csc}(c, O_1, O_2) \cap \mathrm{csc}(c, O_2, O_1)|}{|\mathrm{csc}(c, O_1, O_2)|}$$

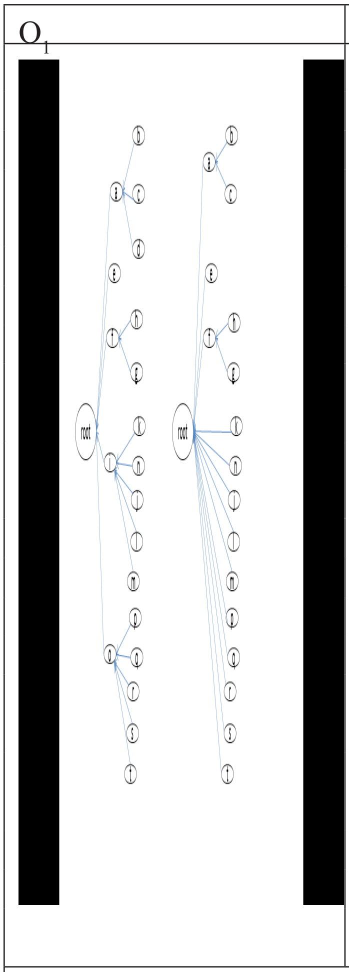
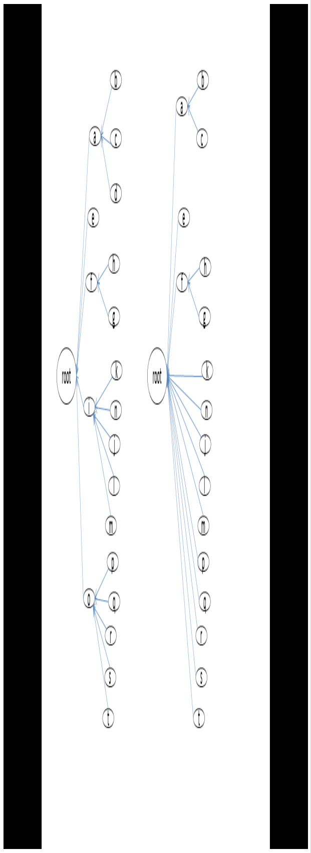
Based on TP and TR, we can compute taxonomic F-measure as:

$$TF(O_1, O_2) = \frac{2.TP(O_1, O_2).TR(O_1, O_2)}{TP(O_1, O_2) + TR(O_1, O_2)}$$

Given two ontologies O_1 and O_2 in which O_1 is the computer learned ontology and O_2 is the reference (or standard) ontology, a part of the evaluation is shown in Table 6. We only take different parts of the two ontologies into consideration, in which a concept c in O_1 has a different position as in O_2. A number of concepts are not considered in this evaluation since they are leaf concepts linked to root nodes (things) in both ontologies. There is no need to find position difference for these concepts.

Table 6

Comparison of the Learned Ontology and Reference Ontology

O_1	O_2	c	csc(c,O_1,O_2)	csc(c,O_2,O_1)
		a	root, b, c	root, b, c, d
		b	root, a	root, a
		c	root, a	root, a
		d	-	root, a
		e	root	root
		f	root, h	root, h
		g	f	f
		h	f	f
		i	-	root, j, k, l, m, n
		j	root	root, i
		k	root	root, i
		k	root	root, i
		l	root	root, i
		m	root	root, i
		m	root	root, i
		o	-	root, p, q, r , s ,t
		p	root	root, o
		q	root	root, o
		r	root	root, o
		s	root	root, o
		t	root	root, o

Description: a: file format; b: music file; c: video file; d: other file; e: smartphone; f: network; g: 2G network; h: 3G network; i: phone software; j: operating system; k: applications; l: music player; m: email; n: call feature; o: phone hardware; p: keyboard; q: memory; r: screen; s: battery; t: earphone

Based on the analysis shown in Table 6, we can compute $TP_{csc}(O_1,O_2) = 100\%$ $TP_{csc}(O_1,O_2) = 100\%$, $TR_{csc}(O_1,O_2) = TP_{csc}(O_2,O_1) = 68.05\%$ $TR_{csc}(O_1,O_2) = TP_{csc}(O_2,O_1) = 68.05\%$, and $TF_{csc}(O_1,O_2) = 0.8$ $TF_{csc}(O_1,O_2) = 0.8$.

Usage of ViText2Onto in the Education Domain

Using ViText2Onto, we built an ontology in the education domain to show its application in real-world projects. The purpose of this project is to build a recommender system of course selection for students of the Information Technology Department at Tra Vinh University. This recommender system takes student profiles as input to output a list of recommended courses. A student profile is created based on the courses taken by the student so far. The knowledge base of this system is an ontology containing information about all courses of the bachelor program in information technology in the school.

The ontology was built based on descriptions of 50 courses in which each course description is stored in one text document. In each document, a course is presented by many units of knowledge that students must learn. Each unit corresponds to a learning objective. Each document includes course name, list of learning objectives, list of knowledge units, list of chapter titles, and the schedule of the course. These documents are available in the school's e-learning system before the beginning of each semester.

We use ViText2Onto to obtain as many concepts as there are courses and learning objective names to construct the ontology. Due to the structure of the source documents, each document only contains a plain list of learning objectives that belong to the corresponding course in which each learning objective is presented by a noun phrase, not a whole paragraph of full sentences like in news or in other types of documents. We did not use the relation extraction feature for this ontology. Consequently, after extracting concepts from documents, we put them in the ontology manually as a semi-automatic approach.

In our ontology, concepts that belong to a course form a concept tree. The concept trees of all courses in the program form the structure of the ontology. Using ViText2Onto, we were able to extract 60% of the concepts used in the ontology. For example, Figure 7 illustrates a concept tree for the course Introduction to C Programming.

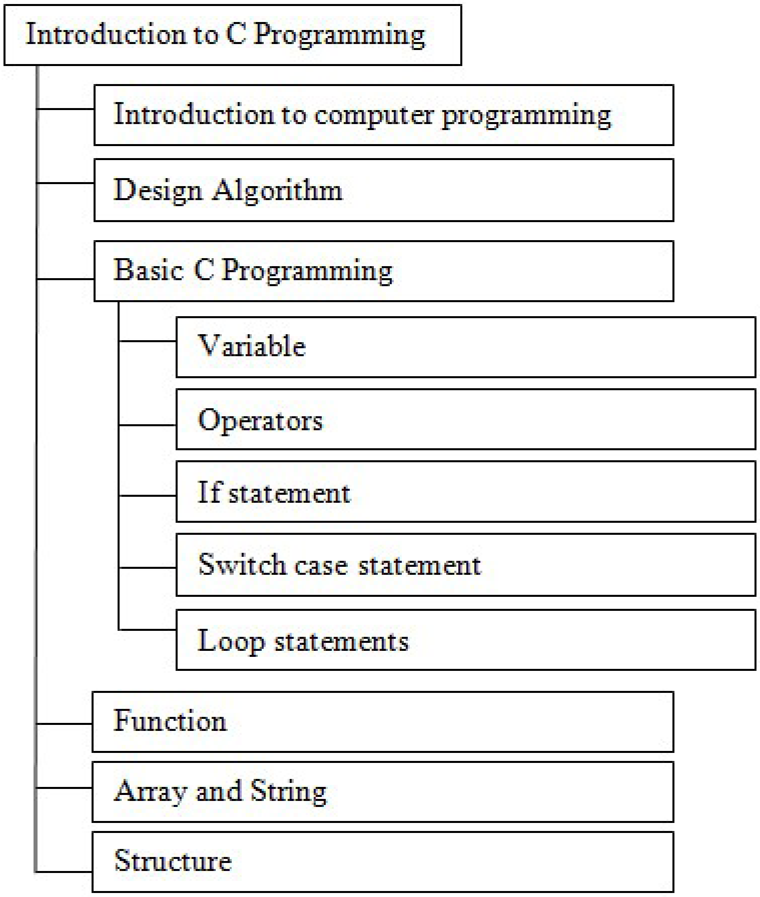


Figure 7. Concept tree for the course of Introduction to C Programming

From a general perspective, the performance of the proposed system is acceptable in supporting users to construct a Vietnamese ontology, in which the labor cost and time consumption are reduced significantly by using the semi-automatic concept extraction method. The accuracy of the system reaches above 50% with our testing datasets. More effort and further studies are on the way to boost the execution of the extraction phase. We believe the overall performance can be improved.

Conclusion and Future Research

In this research, we proposed a support system for Vietnamese ontology construction using the combination of lexical pattern-based, statistics-based, and frequent sequence pattern-based methods. The integrated approach can overcome the weaknesses of each individual method which may lead to missing concepts and relations in the discovery task. We also built a real Vietnamese ontology in the mobile phone domain using our proposed system. Then it was compared with a golden standard of manually constructed ontology. The evaluation shows that our approach has acceptable performance in concept and relation discovery.

In addition, the constructed ontology can be used as a knowledge base in many applications such as a recommendation system, text classification, and information retrieval. Based on our model many knowledge bases can be constructed easily such that more materials are available in open and distance learning.

In the near future, we would like to further automate the ontology construction by automatically learning the taxonomy part of ontology from text documents. Alternative methods of more efficient concept extraction will be considered to take the semantic aspect of documents into account.

References

Ana, M. B., Richard, G., Marla C., Leonardo, C., & Rafael, H. B. (2009). Towards a study opportunities recommender system in ontological principles-based on semantic web environment. *WSEAS Transactions On Computers, 2*(8), 279-291.

Blaschke, C., & Valencia, A. (2002). Automatic ontology construction from the literature. *International Conference on Genome Informatics, 13*, 201-213.

Chen, J., Zhang, Z., Li, Q., & Li, X. (2005). A pattern-based voting approach for concept discovery on the Web. *Lecture Notes in Computer Science, 3399*, 109 -120.

Dellscaft, K., & Staab, S. (2003). On how to perform a gold standard based evaluation of ontology learning. *International Semantic Web Conference* (pp. 228-241.)

Jaimes, A., & Smith, J. R. (2003). *Semi-automatic, data-driven construction of multimedia ontologies*. IEEE International Conference On Multimedia and Expo.

Lee, C. S., Kao, Y. F., Kuo, Y. H., & Wang, M. H. (2007). Automated ontology construction for unstructured text documents. *Data & Knowledge Engineering, 60*, 547-566.

Li, P. S., & Rui, M. S. (2005). Ontology-based learning content recommendation. *International Journal of Continuing Engineering Education and Life Long Learning, 15*(3-6), 308-317.

Maedche, A., & Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems, 16*(2), 72-79.

Maedche, A., & Staab, S. (2000). Semi-automatic engineering of ontology from text. *12th International Conference on Engineering and Knowledge Engineering* (pp. 231-239).

Nguyen, C. Q., & Phan, T. T. (2009). Key phase extraction: A hybrid assignment and extraction approach. *11th International Conference on Information Integration and Web-based Applications & Services* (pp. 276-283).

Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology* (Technical report SMI-2001-0880). Stanford Medical Informatics.

Saman, S., Seyed, Y. B., Nor, A. M. Z., & Shahrul, A. M. N. (2012). Ontological approach in knowledge based recommender system to develop the quality of e-learning system. *Australian Journal of Basic and Applied Sciences, 6*(2), 115-123.

Subhashin, R., & Akilandeswari, J. (2011). A survey on ontology construction methodologies. *International Journal of Enterprise Computing and Business System, 1*(1).

Zheng, Y., Dou, W., Wu, G., & Li, X. (2007). Automated Chinese domain ontology construction from text documents. Bio-inspired computational intelligence and applications. *Lecture Notes in Computer Science, 4688*, 639-648.

Zhou, J., & Wang, S. (2010). Concept mining and inner relationship discovery from text. In J. Zhang (Ed.), *New advances in machine learning*. Retrieved from Intech: http://www.intechopen.com/books/new-advances-in-machine-learning/concept-mining-and-inner-relationship-discovery-from-text

GATE – General Architecture for Text Engineering - http://gate.ac.uk/

Jena: A Semantic Web Framework for Java: http://jena.sourceforge.net/

OntoLT: http://olp.dfki.de/OntoLT/OntoLT.htm

OWL: http://www.w3.org/TR/owl-ref/

TextToOnto: http://sourceforge.net/projects/texttoonto

VLSP Project: http://vlsp.vietlp.org:8080/demo/?page=home

vnToolkit: http://www.loria.fr/~lehong/tools/vnToolkit.php

Data source Web sites:

http://www.mainguyen.vn/tintuc/

http://sohoa.vnexpress.net/sh/dien-thoai/smartphone/

http://thegioididong.com/tin-tuc-dien-thoai,trang-chu-1.aspx