Regular Article

An Enhanced Web Document Search Engine using a Semantic Network

Sang Thanh Thi Nguyen[1], Tuan Thanh Nguyen[2]

[1] School of Computer Science and Engineering, International University, Vietnam
[2] School of Computer Science, University of Nottingham, United Kingdom

Correspondence: Sang Thanh Thi Nguyen, nttsang@hcmiu.edu.vn
Communication: received 14 December 2015, revised 15 March 2016, accepted 4 April 2016
Online publication: 10 October 2016, Digital Object Identifier: 10.21553/rev-jec.134
The guest editor coordinating the review of this article and recommending it for publication was Dr. Tran Manh Ha.

Abstract– **With the rapid advancement of ICT technology, the World Wide Web (referred to as the Web) has become the biggest information repository whose volume keeps growing on a daily basis. The challenge is how to find the most wanted information from the Web with a minimum effort. This paper presents a novel ontology-based framework for searching the related web pages to a given term within a few given specific websites. With this framework, a web crawler first learns the content of web pages within the given websites, then the topic modeller finds the relations between web pages and topics via keywords found on the web pages using the Latent Dirichlet Allocation (LDA) technique. After that, the ontology builder establishes an ontology which is a semantic network of web pages based on the topic model. Finally, a reasoner can find the related web pages to a given term by making use of the ontology. The framework and related modelling techniques have been verified using a few test websites and the results convince its superiority over the existing web search tools.**

Keywords– **Semantic network, ontology, topic models, search engine.**

1 Introduction

The World-Wide-Web (WWW) has become a giant information repository with the rapid development of web technology and cloud computing. On one hand, it is appealing that this information repository is getting richer and richer, in other words, there is almost anything that one can think off in the WWW. On the other hand, it has become harder and harder to find relevant information from the WWW due to the huge volume of information available as described as information explosion. An efficient search engine is a long-term research focus aimed at finding better ways to retrieve required information in a timely fashion. There are many groups who have devoted great efforts to research this topic, such as Google, Microsoft and Baidu. Over the last decade, these efforts have improved the performance of existing search engines dramatically in terms of time spent to return the results. However, the key-word based searching strategy is still dominantly used in the existing search engines. This can lead to the situations in which the information retrieved is syntactically relevant but semantically irrelevant. To overcome such a limit, this paper presents a framework of document search engine for web document retrieval from a specific set of websites using a semantic network of web documents. With the semantic network of web documents, a crawler can retrieve documents not only syntactically but also semantically relevant to a given query. A number of experiments conducted using a prototype of this new search engine have shown that performance of this search engine is superior to the existing search tools, including Google search which is considered as the most powerful search engine in the world.

The rest of paper is organised as follows. Section 2 presents related work. Section 3 describes the framework in details, and gives remarks on the framework. Section 4 provides the experimental comparisons of performance of new search engine with Google search and some website-specific local search functions. Section 5 concludes the paper with some future work suggestions.

2 Related Work

Google now dominates web search field by its outstanding characteristics, such as personalized, socialized and semantic-based search. Most of news sites, such as Theguardian, employ Google as their search function within the sites. Other websites, e.g. Wikipedia, or ABC news websites, also have a search function which is very crucial to quickly find information, e.g. articles, topics or any subjects, on the sites. Because of this indispensable function, web search has always been interesting to many researchers.

At the beginning, Google as well as the other early web search engines, e.g. Lexis-Nexis and Yahoo!, retrieve web information by keywords, but this approach takes much time to match keywords with a huge number of web documents. Hence, indexing web documents was studied, particularly in Google and Altavista, to speed up searching web. Indexing a web page is performed by creating a compressed description based

on keywords, key phrases or other vital descriptors. Besides, to determine the most appropriate web pages in the search results of a web search engine, many features need to be used in combination [1], such as cosine similarity and term proximity, together with the PageRank score which is assigned to each web page in the web graph. The idea behind PageRank is that pages visited more often are more important. As a result, we can see Google ranks the most relevant and interesting web pages on the top positions efficiently.

However, the above web searching approaches face some semantics problems, that is, search results contain searched keywords but are irrelevant and are not diverse nor rich. Now users want to have a search engine which can return relevant results while they may enter not the exact keywords but close ones. Also, keywords do not commonly appear in titles, but in content. This leads to indexing failures and demands a long time to match keywords. What they expect is the search engine can help them find relevant documents which contain or are just related to some given keywords. On other words, the search machine should understand the information provided by the user. The search results are expected to be suitable in terms of semantics.

Different from Google, Wikipedia takes a semantic search engine into account, that is, it provides search results based on meaning match, rather than by the popularity of search terms. In this approach of semantic search [2, 3], an ontology language, e.g. Resource Description Framework (RDF)[1] or Web Ontology Language (OWL)[2], is used to map keywords to concepts or to combine uncertainty with logics. Thus, Wikipedia is provided with better capabilities for understanding of the semantics of user queries and categorizing search results [4].

Similarly, the content-based searching approach has been studied to reason relevant cases [5]. This approach has been deployed in the digital repositories by using ontology to define objects, such as resources and services, and relations among them. Experimental results showed that the ontology-based searching approach is better than Google in terms of user satisfactory and number of returned pages. Moreover, its processing time is also better than traditional search engines. By using ontology to represent domain knowledge, we can enrich search results, solve some problems of keyword matching [6] and achieve better search performance in terms of precision and recall.

To improve information retrieval in search engines, besides, some latent semantic models have been used. For example, Latent Semantic Analysis model (LSA) [7], developed from the vector space model, was adopted in search systems. By applying the singular value decomposition (SVD) to document-term matrices, a document can be mapped to a low-dimensional concept vector in order to optimize comparing relations among terms and documents. Extending from LSA, probabilistic topic models, such as probabilistic LSA (PLSA) [8] and Latent Dirichlet Allocation (LDA) [9] have also been proposed for semantic search.

In addition, some variants of latent space models have been conducted to extend the aforementioned latent semantic models [10]. For example, Rosipal and Krämer [11] proposed the use of Partial Least Square (PLS) for learning of latent space model, i.e., two linear projection functions represented as orthonormal matrices are used to match pairs of queries and documents. However, PLS has problems with large data sets because of solving SVD. To address this issue, a sparsity assumption is made for Regularized Mapping to Latent Space (RMLS) [12]. It is verified that the optimization problem of RMLS can be performed row by row and column by column of the matrices, thus the algorithm can be easily parallelized.

Alternatively, Gao *et al.* [13] proposed the use of Bi-Lingual Topic Models (BLTMs), a probabilistic model for query-document matching at the semantic level. The idea of this method is that each query-document pair is assumed to be generated from the same distribution of topics. This topic model is combined with the expectation maximization method to estimate the topic distributions of query words and document words in order to form the "latent space".

Because the semantic relations between a query and a document are quite complex, deep learning techniques have been taken into account, such as, a deep structured semantic model (DSSM) [14] which has been proposed and has achieved better performance for semantic matching. On the other hand, another variant of LDA model, Multivariate Bernoulli LDA, has proposed to discover the hidden user intents of queries for selecting diverse items most relevant to a user in e-commerce sites [15]. This approach would help users find diverse items and achieve more satisfied search results than the eBay production ranker and three other diversified retrieval approaches, i.e, Maximal Marginal Relevance (MMR), Probabilistic Latent Maximal Marginal Relevance (PLMMR), and the category-based approach.

It has been shown that the LDA model is one of the best approaches to semantic searching. Targeting at latent semantic models is a promising trend for modelling documents in web search, but it has not been compared much with the Google search. As a popular and powerful search tool, Google or Wikipedia can return high confident search results, so is often considered as the baseline for comparisons. However, results returned from such the search tools are often too broad because web pages are fetched from many sites, in that, popular sites get higher priority. In addition, sites paying for being listed first in search results can also bias the search engines. In this paper, therefore, we concentrate on searching within specific sites by utilizing the LDA model to propose a new semantic search approach. The proposed LDA-based approach will be examined on how it improves the searching performance in terms of ranking pages relevant to queries, and compared with the popular search tools of Google, Wikipedia, and the ABC News site in some specific domains. The details of the proposed approach

[1]RDF: https://www.w3.org/RDF
[2]OWL: https://www.w3.org/2001/sw/wiki/OWL

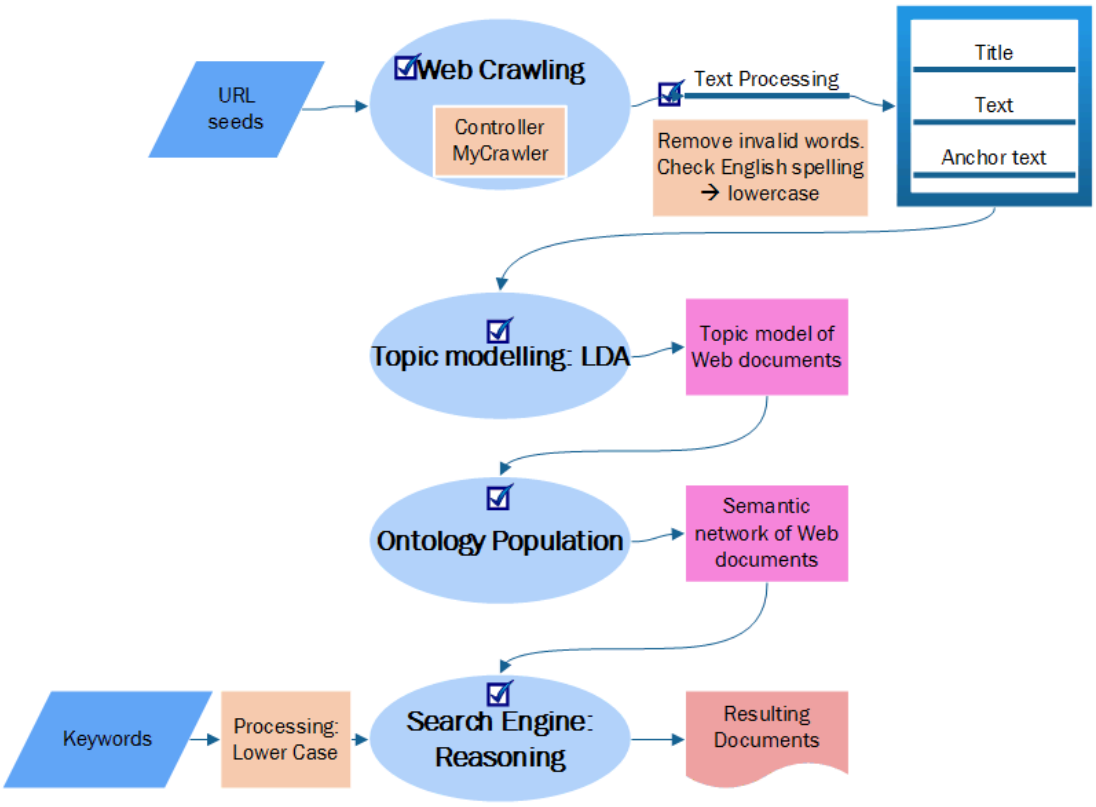

Figure 1. Framework of ontology-based topic model for web search.

are described in the following framework.

3 Framework

This section presents a conceptual framework of web document search engine using a semantic network of documents. The input to the search engine is a search query which consists of a term/keyword. Its output is a list of N web pages ordered from the most relevant to the least relevant to the search query. Before it accepts any search tasks, the engine needs to build a knowledge base of topics in a specific area of interest by taking in one or more seed websites/URLs.

Figure 1 depicts the block diagram of this framework. It consists of four modules, which are web the crawler, the topic modeller, the ontology builder, and the reasoner.

The work flow of this framework can be briefly described as follows: for the given seed URLs, the web crawler will fetch web pages and extract web contents, titles, and anchor text in the fetched web pages. Web contents are then modelled by LDA model [9] to build a topic model. The topic model of web documents is transformed into a semantic network using the ontology techniques for reasoning about keywords and topics in the web pages (or documents). Based on this semantic network, the reasoning unit will find out the web pages/documents related to the search query.

In the rest of this session, each of the four modules will be described.

3.1 Web Crawler

This module is for fetching the web pages that are directly or indirectly linked with the given seed URLs, extracting text from web pages and cleansing the text. The inputs to this model are the URLs of seed web pages (e.g., home page of a company's website). For each web page fetched, the crawler scans through the web page content. The titles, anchor text, and content text in these pages will be extracted. It also processes the extracted text by removing invalid words, such as the stop words, checking English spelling, and/or converting the text into lower case. The output of this module is the cleansed/pre-processed text.

3.2 Topic Modeller

This module is for building a topic model of web pages under consideration. The input to this module is the cleansed text string. The topic model is constructed using Latent Dirichlet Allocation (LDA). In this model, the topics are represented by words extracted from web pages/documents and the web pages are assigned to related topics. The output of this module is the topic model representation of the text.

According to Blei *et al.* [9], a document is represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Formally, the following defines terms in the LDA model:

- A word is defined to be an item from a vocabulary indexed by $1, \ldots, V$. A word is represented by a V-vector w such that $w^v = 1$ and $w^u = 0$ for $u \neq v$ if v is the order of this word in the vocabulary.

- A **document** is a sequence of N words denoted by $w = (w_1, w_2, ..., w_N)$, where w_n is the n-th word in the sequence.
- A **corpus** is a collection of M documents denoted by $D = w_1, w_2, ..., w_M$.
- It is assumed that a fixed number of topics z is known, $z = z_1, ..., z_K$, where z_n is the n-th topic in the set.
- Dirichlet parameters: α and β are corpus-level parameters, sampled once in the process of generating a corpus.
- The variables θ_d are document-level variables, sampled once per document.
- A k-dimensional Dirichlet random variable θ can take values in the $(k-1)$-simplex (a k-vector θ lies in the $(k-1)$-simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1},$$

  where the parameter α is a k-vector with component $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function.
- The probability of a sequence of words and topics is computed as follows:

$$p(\mathbf{w}, \mathbf{z}) = \int p(\theta) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n) d\theta, \quad (1)$$

  where θ is the random parameter of a multinomial over topics, $p(z_n|\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$, $p(w_n|z_n, \beta)$ is a multinomial probability conditioned on the topic z_n, β is a $K \times V$ matrix, where β_{ij} is a fixed quantity that is to be estimated.

Therefore, words are assigned to the corresponding topics with different probabilities computed as (1). Furthermore, topic distribution for document m is computed by the following formula [16]:

$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{j=1}^{K} n_m^{(j)} + \alpha_j}, \quad (2)$$

where $n_m^{(k)}$ is the number of times the word k in document m is assigned to topic k.

Although this model can be used to classify web pages/documents when stored in text files, it is not easy to use this model for searching web documents relevant to a given word. Moreover, the drawback of the LDA model is that it does not present the relationship between topics, so it is proposed to use the ontology language to represent this model so that relations among topics are established, as well as relations among documents, words and topics.

3.3 Ontology Builder

In order to be able to facilitate reasoning about information in the web pages/documents as well as topics, this module is built to automatically transform the topic model in a text file into a semantic network

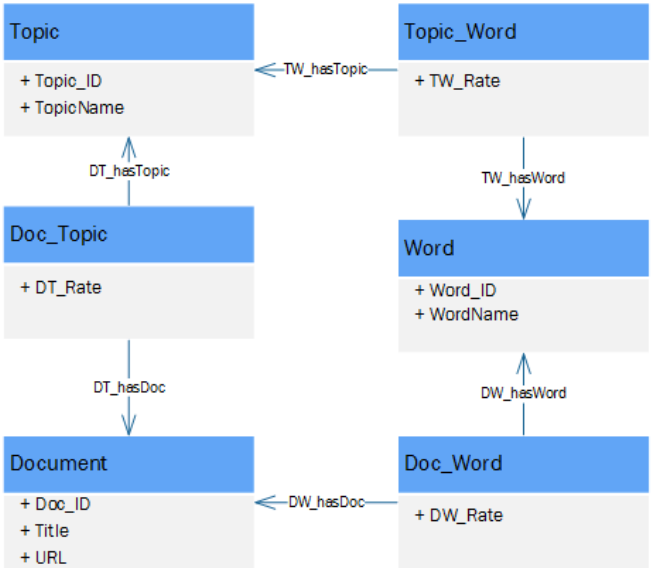

Figure 2. Ontology schema of the proposed semantic network.

of web pages/documents which is implemented using OWL (Web Ontology Language) [17] as a semantic knowledge base. Similar to an ontology, this semantic network is empowered with web sharing and machine-readability. In addition, it can represent complex relations between web pages/documents and topics, and topics and words, as well as some features, such as contribution probabilities between topics and documents, between words and topics, and between documents and words. The ontology schema of this semantic network is depicted in Figure 2.

In this schema, classes Topic, Document, and Word represent the topic, document, and word instances of the topic model, respectively. The relations between topics and words, between documents and words, and between documents and topics are represented by classes Topic_Word, Doc_Word, and Doc_Topic, respectively, along with respective rates which can be obtained from the corresponding distribution probabilities in the topic model.

This semantic network is able to connect web pages/documents of different websites together in accordance with topics, so it enables searching and classifying web pages/documents easily. The coverage of topics depends on the collection of web pages fetched by the web crawler using the given seed URLs.

Through the semantic network, the relationships among topics are able to be reasoned via words and documents.

3.4 Reasoner

This module makes use of the semantic network to reason about related web pages/documents for the query keywords (or a string). This module takes the input search query as keywords and find out the related web pages/documents through reasoning the semantic network and presents the results as a ranked list of web page URLs in the descending order of relevancy to the search query. This reasoning process is described in the following.

Let S be a set of words w_i in the query string, $p(w_i, z_k)$ be the probability of topic z_k associated with word w_i, this probability is computed as (1). Hence, the probability of topic z_k associated with S should be $\sum_{i=1}^{|S|} p(w_i, z_k)$.

Based on the topic model, the occurrence frequency of a word w_i in a document m, namely f_{m,w_i}, is computed by the ratio of the occurrence times of this word in the document and the total of words in the document.

Let Z be a set of topic z_k associated with S, $\vartheta_{m,k}$ be the distribution probability of topic z_k over document m, as computed in (2). Hence, the relevance of document m with the set of topic Z inferred from S can be calculated as follows:

$$\sum_{k=1}^{|Z|} (\vartheta_{m,k} * \sum_{i=1}^{|S|} p(w_i, z_k)) + \sum_{i=1}^{|S|} f_{m,w_i}. \quad (3)$$

By adding $\sum_{i=1}^{|S|} f_{m,w_i}$ into the relevance calculating of a document to the query string, the documents which contain words in the string will be weighted.

The Algorithm 1 describes how to search web pages/documents given a string of words.

Algorithm 1: Searching web pages/documents given a string of words

1) Input: a string of words
2) Output: a list of web pages/documents
3) Process:
   a) Get topics for the given words:
      For each word w_i in the input string,
      i) scan the Topic_Word instances to find out topics associated with word w_i and the corresponding probabilities ($p(w_i, z_k)$).
      ii) the probabilities of the previously found topics associated with word t are accumulated as $\sum_{i=1}^{|S|} p(w_i, z_k)$.
   b) Find documents for the topics
      For each of the found topics, z,
      i) scan the Doc_Topic instances to find out documents associated with topic z,
      ii) and, the relevance between a document and the topics is computed as (3).
   c) Sorting the found documents in descending order of relevance of the documents to the input string.
   d) Return the sorted list of found documents.

3.5 Remarks on the Framework

This framework is mostly suitable for the situations in which the search query returns web pages/documents from a specific set of websites. It is not only better than the search function provided in any specific website but also better than any general-purpose search engine because it makes use of a LDA-based semantic network of web pages/documents constructed based on websites under consideration. It is able to retrieve related web pages/documents to topics expected from certain websites. With this framework, it is unnecessary to search for relevant web pages from each website, neither from the World Wide Web. Furthermore, by employing the LDA model, this framework can be built automatically anytime for many websites, so it would save lots of time to build a search engine tool manually using fixed-code which is only used at a certain period of time.

The limitation of this framework is that it needs a semantic network of web pages/documents. Apart from the extra effort to build this semantic network, it is dependent on the fetched web pages by the web crawler from the given seed URLs. Different seed URLs may result in different topic models which may in term produce different returned web pages/documents. In other words, it may be dependent on seed URLs.

Although the website-specific search functions are widely used for searching within a given website, it is not easy to tell if the results are correct. Furthermore, the existing search engines need to be online to conduct searching while this framework allows searching offline once the semantic network has been constructed.

Google or Wikipedia utilizes hundreds of thousands computers to process trillions of web pages, search through structured and unstructured data sources in order to return fairly well-aimed results. Therefore, their processing time is so fast, we cannot compare the proposed search engine with Google and Wikipedia in terms of processing time because of some limitations of the number of computers used for search, and datasets. There are many other ways of evaluating search engines [18], based on such as retrieval effectiveness tests, results presentation, and search engine user behavior. Testing the retrieval effectiveness of a search engine is often considered, using information queries and the relevance of the first 10 or 20 results. This evaluation method has been used in many studies, such as [5]. Therefore, the following section presents some experiments carried out to evaluate top-ten search results returned from the proposed approach, and compare it with the two popular search engines, i.e., Google and Wikipedia, and search functions of some well-known news sites.

4 Prototype / Demo

This framework has been implemented in Java and Protege (for constructing the ontology schema of the semantic network) and the experiments are run on a Windows Server 2012 PC with Intel Core i7-4770, 3.40 GHz and 8 GB of RAM. In order to show the validity of this framework and the semantic network, two experiments have been conducted using the prototype of the proposed framework. In each experiment, the results of the new framework are compared with a powerful general-purpose search engine, such as Google search, and website-specific search functions.

In the two following experiments, a three point relevance scale, most relevant (1), relevant (0.5) and not

relevant (0), is used to evaluate search results. Query strings are chosen variantly between grammatically correct and incorrect phrases to challenge the search engines. And top-ten search results are taken into account for each query. If a page is returned as expected, it is most relevant. If a page gives some information connected to the query, but not quite to the point, then it is relevant. Pages not containing any information about the query topic are deemed not relevant. Details of each experiments are presented below.

4.1 Experiment 1

In this experiment, the Wikipedia site is considered. Three tested seed URLs are:

- http://en.wikipedia.org/wiki/Technology
- http://en.wikipedia.org/wiki/Science
- http://en.wikipedia.org/

Given these seed URLs, 306 valid web pages are crawled, and assigned to 100 topics to construct a semantic network of Wikipedia pages. For testing, 25 queries are sent to the search engines: the proposed semantic network-based search engine, namely SNSE, and the Wikipedia search tool.

Table I presents the positions of the most relevant pages in search results given the 25 queries. For query strings in correct grammar or meaningful, e.g. queries 2, 3, 5, 6, etc., the two engines are easy to obtain the correct pages at the top position in the search result lists. However, with grammatically incorrect or non-meaningful query strings, e.g. queries 11, 18, 19, and 23, SNSE was still able to find correct pages, while Wikipedia failed to find correct pages. Furthermore, a query string which is not in a page title might cause trouble to the search engine. For example, Wikipedia cannot give a correct page for the given query 25 "agriculture and information technologies" because this search engine is based on keywords mostly in page titles. Fortunately, SNSE can overcome this problem since its search results are based on not only words in the query string but also relevant topics using the built semantic network.

As we can see from the results in Table I, SNSE outperforms Wikipedia on the tested queries. While SNSE found and ranked the correct page at the first position for 80% of all queries, Wikipedia did so 64% of the time, as illustrated in Figure 3.

In addition, to evaluate how many relevant pages the search engine could retrieve, Table II presents the number of the relevant pages including the most relevant ones in the search results. We found that the search results from SNSE is richer than the ones from Wikipedia. SNSE can find 3.96 relevant pages in average, more than Wikipedia which can find 2.24 relevant pages in average per each query.

4.2 Experiment 2

In this experiment, three popular news websites are considered for crawling web documents, so that the built search engine is able to find pages related to a certain topic in these three sites. Table III lists seed URLs for each website.

Table I
Positions of the Most Relevant Pages in Search Results

Query string	SNSE	Wikipedia
1. Technology disambiguation	7	6
2. Horse collar	1	1
3. Horseshoe	1	1
4. Computer science	2	1
5. Scraper archaeology	1	1
6. Space exploration	1	1
7. Technological evolution	1	1
8. Technology management	1	1
9. Post scarcity economy	1	1
10. Roman Empire	1	1
11. Empire Roman	1	0
12. Roman and Greek culture	2	2
13. British academic philologist	1	1
14. Project Socrates	1	1
15. Socrates project	1	1
16. Karl Marx	1	1
17. Nuclear weapon	1	1
18. weapon Nuclear	1	0
19. Motors General	1	0
20. Ecosystem engineer	1	1
21. Engineer ecosystem	1	6
22. Food Technology	4	1
23. Technology food	4	0
24. seeds suicide	1	2
25. agriculture and information technologies	1	0

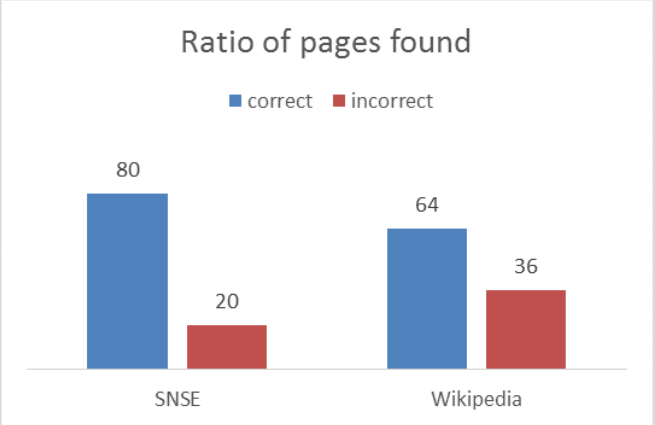

Figure 3. Ratio of relevant vs. irrelevant pages found at the top position.

Given these seed URLs, 302 valid web pages are crawled from the ABC news site, 302 valid web pages are crawled from the ABC7 news site, and 306 valid web pages are crawled from the Theguardian news site. The total 910 pages are assigned to 100 topics to construct a semantic network for the three websites.

For testing, 25 queries are sent to search engines: SNSE, Google, and the search tool of an expected site, e.g. ABC or Theguardian. Since Google usually returns searching results broadly from popular sites, an extra string describing the name of an expected site is added into the query string in order to filter out searching results not belonging to that site. For example, if most

Table II
Number of the (most) Relevant Pages in Search Results

Query string	SNSE	Wikipedia
1. Technology disambiguation	9	5
2. Horse collar	2	1
3. Horseshoe	2	1
4. Computer science	9	1
5. Scraper archaeology	5	4
6. Space exploration	2	1
7. Technological evolution	8	1
8. Technology management	9	1
9. Post scarcity economy	5	1
10. Roman Empire	1	1
11. Empire Roman	2	6
12. Roman and Greek culture	3	6
13. British academic philologist	2	1
14. Project Socrates	1	1
15. Socrates project	1	1
16. Karl Marx	1	1
17. Nuclear weapon	3	1
18. weapon Nuclear	3	5
19. Motors General	3	5
20. Ecosystem engineer	3	1
21. Engineer ecosystem	2	6
22. Food Technology	9	1
23. Technology food	9	3
24. seeds suicide	3	1
25. agriculture and information technologies	2	0

Table III
Seed URLs of News Websites

Query string	SNSE
ABC	http://www.abc.net.au/science http://www.abc.net.au/news/world http://www.abc.net.au/
ABC7	http://www.abc7.com/us-world http://www.abc7.com/health http://www.abc7.com/sports http://www.abc7.com/entertainment/ http://www. abc7.com/
Theguardian	http://www.theguardian.com/uk/technology http://www.theguardian.com/world http://www.theguardian.com/

of resulting pages are expected to come from the ABC News site, then an added extra string would be "ABC News", and the third tested search tool would be the one of ABC News. It is fairer to compare search results with Google and we can evaluate how Google searches a specific site, e.g. ABC News, for results relevant to a query string.

In this experiment, we only focus on investigating how the search engines can lift the most relevant pages up to the top position. Table IV shows the positions of the most relevant pages in the search results of the three tools. It has been found that SNSE could find correct pages for all queries, but Google and the expected site could not do that for four and six queries, respectively. The bottom position where the most relevant pages could be found by SNSE and Google is 7. The specific site is more inferior since the bottom position it could find the most relevant page is 9.

Table IV
Positions of the Most Relevant Pages in Search Results

Query string	SNSE	Google	Expected site
1. All Blacks stay unbeaten win Tonga	1	1	1
2. unbeaten Tonga win	4	1	1
3. EU speed deportation economic migrants	3	1	1
4. asylum seeker crisis	4	4	9
5. seeker crisis asylum	4	1	0
6. Skin cells used grow mini kidneys dish	1	1	1
7. dish kidneys	1	1	3
8. Nobel DNA research	7	4	3
9. outnumber Islamic protesters	3	1	1
10. Islamic protesters Muslim	1	3	9
11. Uber driver attacked	1	1	1
12. Brisbane court Uber driver	2	1	2
13. Violence Indigenous communities	2	0	0
14. Wine trouble apps	4	4	1
15. players cheating sports	2	1	4
16. homes starter for rent	2	2	4
17. Getting children sparkly eyed	3	7	1
18. homes building for average earnings	1	5	0
19. the growth of Uber sharing economy	1	0	0
20. sharing economy taxi and the growth of Uber	1	8	0
21. shoppers say goodbye free plastic bags	1	1	1
22. UK customers protect environment and bags	1	2	6
23. clothing response gaze camera tracking	1	0	0
24. smartphones vulnerable	1	1	1
25. Clinton private email accessed five times	1	0	1

Furthermore, it has been seen that Google and the search tools of specific sites are in trouble with query strings not matching page titles, for examples in queries 18 and 22. Google is biased by hot news, e.g. in queries 15 and 17. Instead of returning the most relevant pages first, Google returned pages many people are interested in first. And Google could be confused in some cases when the query strings are grammatically incorrect, as shown in query 25. On the other hand, the proposed search engine could overcome these problems. As illustrated in Figure 4, the ratio of correct pages found at the top position in SNSE is equal to the one in Google, i.e. 48% correct pages found at the top position in the search results. Their search tools are superior to the website-specific search functions.

In addition to considering the most relevant pages, we also looked into the lists of search results and found that most of results include relevant web pages which may be from one or all of the three tested sites. It shows that a news topic which can be found in different news

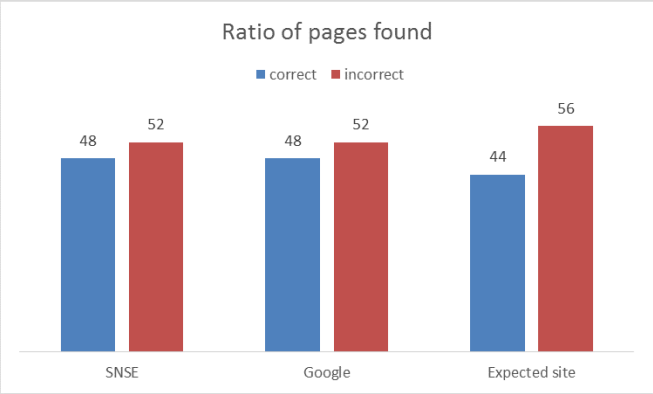

Figure 4. Ratio of relevant vs. irrelevant pages found at the top position.

websites is also returned by SNSE. On the other hand, the semantic network successfully combines web pages from the three sites and constructs the relationships among web pages through topics.

4.3 Remarks

From the experimental results, we can see that the search results of SNSE are diverse and ranked in relation with the words in the query strings. Pages which are directly related to the words are sorted first. The proposed search engine can overcome the challenges of inexact query strings or words not in page titles. Especially, this search engine avoids being biased by hot news or popular sites unlike the existing search engine tools. These results show that the proposed search model is effective and promising.

Whereas, the built semantic networks in the experiments only cover the limited number of crawled web pages for testing, so the number of retrieved relevant documents is small. This limitation can be solved by increasing number of crawled pages. In this paper, indexing pages is not used in SNSE, so the searching time is not examined. However, the response time of the search engine is less than one second for every query, that is acceptable for a search tool. As earlier mentioned, it is not suitable to evaluate the processing time of SNSE in this study. The main point in this study is how to lift the most relevant documents to the top position in search results and the experimental results have proved that the proposed framework could compete with the popular modern search engines, e.g., Google and Wikipedia.

5 Conclusions

This study contributes significantly to the field of ontology-based search model. The proposed framework allows us to search web pages according to topics in certain websites. The search results concentrate on some domains of interest, not on the entire Web. This helps users not to be confused with websites not of interest to them, and to be able to gain what they expect quickly. Moreover, the results are not only related directly to given keywords, but also about related topics within the domain. Users can get more documents related to their interests. The experimental results have proved that the proposed framework of web search is achievable, and can be developed in future semantic search.

The important point is the proposed framework can direct search documents into expected web paths, can be customized based on users demands so that they only need to type simple keywords to get more focused documents.

In the future, more web pages will be crawled, so optimizing the semantic network might be necessary. Words extracted from web pages will be refined carefully to obtain good words for building the topic models. The searching algorithm also needs to be improved to place more relevant pages at first rank.

References

[1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2009.

[2] A. Ramachandran and R. Sujatha, "Semantic search engine: A survey," *International Journal of Computer Technology and Applications*, vol. 2, no. 6, pp. 1806–1811, 2011.

[3] G.Madhu, A.Govardhan, and T.V.Rajinikanth, "Intelligent semantic web search engines: A brief survey," *International journal of Web & Semantic Technology (IJWesT)*, vol. 2, no. 1, 2011.

[4] S. Halling, M. Sigurðsson, J. E. Larsen, S. Knudsen, and L. K. Hansen, "Muzeeker: a domain specific wikipedia-based search engine," in *Proceedings of the First International Workshop on Mobile Multimedia Processing (WMMP2008), Florida (December 2008)*, 2008.

[5] A. Martin, "Intelligent search engine to a semantic knowledge retrieval in the digital repositories," *International Journal on Advances in Intelligent Systems*, vol. 8, no. 1 & 2, pp. 67–76, 2015.

[6] S. Kohli and S. Arora, "Domain oriented semantic web based personalized search engine," pp. 23–28, 2014.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of The American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[8] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[10] H. Li and J. Xu, "Semantic matching in search," *Foundations and Trends in Information Retrieval*, vol. 7, no. 5, pp. 343–469, 2014.

[11] R. Rosipal and N. Krämer, *Overview and Recent Advances in Partial Least Squares*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 3940, ch. 2, pp. 34–51.

[12] W. Wu, Z. Lu, and H. Li, "Learning bilinear model for matching queries and documents," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2519–2548, 2013.

[13] J. Gao, K. Toutanova, and W.-t. Yih, "Clickthrough-based latent semantic models for web search," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 675–684.

[14] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of*

the 22nd ACM international conference on information & knowledge management. ACM, 2013, pp. 2333–2338.

[15] J. Yu, S. Mohan, D. P. Putthividhya, and W.-K. Wong, "Latent dirichlet allocation based diversified retrieval for e-commerce search," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 463–472.

[16] X. H. Phan, L. M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th International World Wide Web Conference (WWW 2008)*, pp. 91–100.

[17] G. Antoniou and F. V. Harmelen, *Web Ontology Language: OWL*. Springer-Verlag Berlin Heidelberg, 2009, pp. 91–110.

[18] D. Lewandowski, *A Framework for Evaluating the Retrieval Effectiveness of Search Engines*. IGI Global, 2012.

Sang Thanh Thi Nguyen has received her PhD degree in Software Engineering from the University of Technology, Sydney (UTS) in 2013. Her PhD thesis is about Semantic-enhanced Web-page Recommender Systems. She was supervised by Dr. Helen Lu and Prof. Jie Lu. She received her Master degree in Computer Engineering from the University of Technology (VNU-HCMC) in 2006. She is working as a lecturer at the School of Computer Science & Engineering at the International University (VNU-HCMC) from 2015. She has eight published research papers in the field of Web mining. Her research interests include Web mining, Semantic Web, knowledge discovery and business intelligence.

Tuan Thanh Nguyen is currently a research fellow at Computer Vision Laboratory from School of Computer Science, University of Nottingham, United Kingdom where he received his PhD degree in Computer Science. His research interests include image processing and analysis, tracking, feature detection, object recognition and machine learning. He is involving a number of Europe and UK projects such as iBit, i4i and iPlant.