HAL open science

A Vietnamese Question Answering System in Vietnam's Legal Documents

Huu-Thanh Duong, Bao-Quoc Ho

To cite this version:

Huu-Thanh Duong, Bao-Quoc Ho. A Vietnamese Question Answering System in Vietnam's Legal Documents. 13th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Nov 2014, Ho Chi Minh City, Vietnam. pp.186-197, 10.1007/978-3-662-45237-0_19 . hal-01405581

HAL Id: hal-01405581

https://inria.hal.science/hal-01405581

Submitted on 30 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A Vietnamese Question Answering System in Vietnam's Legal Documents

Huu-Thanh Duong(1), Bao-Quoc Ho(2)

Faculty of Information Technology, University of Science, VNU-HCM
(1)dhthanhqa@gmail.com, (2)hbquoc@hcmus.edu.vn

Abstract. In this paper, we develop a Vietnamese Question Answering system to answer simple questions about provisions, processes, procedures, and sanctions in law on enterprises Vietnam. Research to build a Vietnamese Question Answering system is more difficult than English Question Answering system because of lack of Vietnamese processing resources and tools, or their results isn't high. We have utilized available Vietnamese resources, tools and modified some tools (such as Lucene) and algorithms, which worked well in English, applies in Vietnamese. From that, we have proposed a similarity-based model to build a Vietnamese Question Answering system in Vietnam's legal documents, namely vLawyer system. In experimental section, we achieved promising result, about 70% precision in legal documents, it proved that our approach is reasonable in legal document domain.

Keywords: Question Answering, QA, Legal Document, Law on Enterprises, Information Retrieval, IR, Natural Language Processing.

1 Introduction

Today, the development of the information technology has made people overwhelmed in huge documents in the internet. The internet contains lots of useful documents for users, but it has also no less dirty ones. So, to utilize useful documents quickly and effectively, it has to be necessary to develop a system to satisfy those demands. A Question Answering (QA) system is the answer for that. It isn't like the information retrieval (IR) system, only returning a list of documents related to keywords of user's query, QA gives a succinct and short answer for a query as user's natural language. QA becomes a subject which has attracted lots of research groups. It is a system inherited from the information retrieval system, it helps text mining becomes more effectively, quickly and practically. While the English QA achieved lots of remarkable results, the Vietnamese QA has still been under development because there are lots of difficulties in lack of resources and tools to process natural language such as Chunker, Named Entity Recognition (NER), etc. In this paper, we want to utilize some available tools in Vietnamese natural language processing such as VietTokenizer, jvnTagger, etc and other tools such as Lucene which is a very strong tool to support to build information retrieval systems. Besides, we have also utilized ideas of

adfa, p. 1, 2014.
© Springer-Verlag Berlin Heidelberg 2014

some algorithms applied in English effectively such as KEA (Keyphrase Extraction Algorithm) to identify the keyphrases, this algorithm is cross-language. With limited resources and tools, we hope to have a little contribution to build a Vietnamese QA system. In this system, we develop a Vietnamese QA system bases on Vietnam's legal documents, especially as law on enterprises 2005.

The remaining of this paper is organized as followed: in next section, we present some related works for our system. In section 3, we introduce to overview about legal documents. And in section 4, we present our built system, namely vLawyer, including the theoretical basis to build our system and system architecture. Finally, in section 5, we present a case study, give the conclusion and future works.

2 Related Works

While the IR system had more and more exact results, the QA system has not still impressed results yet, especially as Vietnamese. However, developing Vietnamese QA system has had first-step respected results. We will show some achievements in developing Vietnamese QA system as followed:

Vu Mai Tran, Vinh Duc Nguyen, Oanh Thi Tran, Uyen Thu Thi Pham, Thuy-Quang Ha [3] studied to build a Vietnamese QA system in travel domain by incorporating Snowball methodology and relational extraction using search engines. The system, firstly, extracted patterns and tuples into dataset. This system supported to answer simple questions about entities relationship. When it received a user's query, it would extract entities and look for related candidate relations. Then, it calculated similarities between query vector and all patterns of candidate relations to choose the best pattern. Finally, it built a query to database by using entity information (E) and relation (R) to get tuples contained relation R and entity E. According to authors, this system achieved 89.7% precision and 91.4% ability to give the answer when testing on traveling domain.

Mai-Vu Tran, Duc-Trong Le, Xuan-Tu Tran, Tien-Tung Nguyen [4] built a Vietnamese person named entity question answering model. It's a closed QA answering person factoid questions (such as Who, Whose, Whom). The question processing model used CRF mechine learning algorithm and two automatic answering strategies: indexed sentences database-based and Google search engine-based. If answers were not found in database, the question would be pushed into Google search engine. The system filtered candidate answer collection based on their similarities with question and assigned a priotity number to the candidate answers. Finally, the system ranked the answers and sent to user for final validation in order to extract the exact answer. According to authors, the system obtains 74.63% precision and 87.9% ability to give the answer.

Dai Quoc Nguyen, Dat Quoc Nguyen, Son Bao Pham [5] proposed an ontoloy-based Vietnamese question answering system that allowed users to make a query in natural language. It included two components: question processing and answer retrieval. They built a set of relations in the ontology that included only two person relations. According to authors, results were relatively high, the Question Analysis module and the Answer Retrieval module achieve an accuracy of 95% and 70% respectively, 5%

remaining errors are due to the lack of coverage of their Jape grammars in the pattern-matching module. However, the cost for building the database was high.

Dang-Tuan Nguyen and Ha Quy-Tinh Luong [9] built a system to search courses in Vietnam OpenCourseWare program, it allowed users to input question in vietnamese natural language. It, firstly, extracted terms from VOCW pages and stored in ontology-based knowledge. When it received user's query, the system would analysis question parser based on the set of defined syntax rules and build a query in SPAEQL to get data on VOCW ontology.

Dang Truong Son, Dao Tien Dung [11] built a Vietnamese QA system by mapping user's query with questions in a defined dataset as couples of question/answer which were extracted from the internet. With returned answers, authors hope that one of them will satisfy user's demand. According to authors, they extracted 90000 couples of question/answer from Yahoo Answer and also got 50 questions randomly for testing, they obtained higher results compared with answering results from Yahoo.

From these systems, we try to build a question answering system in a relatively popular and useful domain, namely legal documents, based on similarities between documents with using $tf \times idf$ weight and latent semantic indexing.

3 Vietnam's Legal Documents

In this section, we introduce overview of Vietnam's legal documents and a legal structure. According to Law No. 17/2008/QH12: "Legal documents are documents issued or jointly issued by state agencies in accordance with the authority, formats, sequence of steps and procedures prescribed in this Law or the Law on the Promulgation of Legal Documents of People's Councils and People's Committees, which includes common rules of conducts, which has compulsory effectiveness and the implementation of which is guaranteed by the Government to regulate social relations.". The system of legal documents includes (in order of legal priority):

- Constitution, laws and resolutions of the National Assembly.
- Ordinances and resolutions of the Standing Committee of the National Assembly.
- Orders and decisions of the State President.
- Decrees of the Government.
- Decisions of the Prime Minister.
- Resolutions of the Justices' Council of the Supreme People's Court and circulars of the Chief Justice of the Supreme People's Court.
- Circulars of the President of the Supreme People's Procuracy.
- Circulars of Ministers or Heads of Ministry-equivalent Agencies.
- Decisions of the State Auditor General.
- Joint resolutions of the Standing Committee of the National Assembly or the Government and the central offices of socio-political organizations.
- Joint circulars of the Chief Justice of the Supreme People's Court and the President of the Supreme People's Procuracy; those of Ministers or Heads of

Ministry equivalent Agencies and the Chief Justice of the Supreme People's Court, the President of the Supreme People's Procuracy; those of Ministers or Heads of Ministry-equivalent Agencies.
- Legal documents of People's Councils and People's Committees.

A legal document content is divided by levels: chapters, sections, articles, clauses of an article, each clause can contain small sentences. When extracting legal documents from Ministry of Justice's website, the system also gets these documents and bases on this priority to assign a boost value to documents when indexing documents with Lucene. A legal structure includes: assumption, provision and sanction.

- Assumption designates situations (conditions and circumstances) which can occur in practice.
- Provision is subject's behaviors when it faces the situation which is mentioned in assumption section.
- Sanction gives penalities if the subject does not perform rightly what the provision section designated in the situation of the assumption section.

When extracting candidate passages, the system will get an article or a clause of the article or a sentence of the clause of the article, namely a legal.

4 vLawyer System

We named the system as vLawyer (Virtual Lawyer), this is a Vietnamese question answering system to answer simple questions about Vietnam's law on enterprises text. Questions can mention about provisions, processes, procedures and sanctions in law on enterprises. The system allows a question as natural language and the answer is an relevant article or a clause of the article or a sentence of the clause of the article.

4.1 Theoretical basis

Document Preparation.

A legal document must be concise, clear and conform. Thus, it's necessary to choose reliable sources to extract. We choose Vietnam Ministry of Justice's website to extract laws, decrees, circulars, decisions that are still validity or partially validity. All of documents were converted to lowcases and tokenized and tagged Pos by VietTokenizer and jvnTagger [16] respectively.

This corpus will be bigger and bigger during user's searching process. If the system doesn't find the answer in current corpus, it will push query to Google search engine to get more documents from Vietnam Ministry of Justice's website.

At present, we extracted about 113 documents, including laws, decrees, circulars, decisions related to law on enterprise.

Lucene Tool.

Lucene [1] is a free and strong open-source information retrieval software library, developed by Dough Cutting. This library includes basic functions for indexing and searching. Lucene responses results quickly from a big set of files because Lucene searching bases on documents' index, but not simple to search on text directly. Lucene has chosen by lots of researchers because it's simple, effective and cross-language.

In Lucene, the unit of searching and indexing is documents. A document has lots of fields, each field is a couple of name/value. Indexing in Lucene is to create the document with one or lots of fields. To index the text, Lucene extracts the keywords and removes stopwords, keywords are created inverted index and stored into each segment for searching.

After indexing documents, we can search with Lucene based indexed results. Searching in Lucene bases on a list of keywords and logic operators (*AND, OR, NOT*). When searching, users must designate a field to search and make a query, Lucene uses this query to search results for user, based on ranking the similarity scores between documents and the query.

In this system, we use the set of documents extracted from Vietnam Ministry of Justice's website and they are tokenized by VietTokenizer and indexed by Lucene. When the system receives user's query, it will extract keyphrases/keywords and search relevant documents by using Lucene.

Term weight.

Three following weights are used the most in term weights:

- Term Frequency (*tf*): A number of times of a term appear in a document. It gives measure of importance of the term within the particular document.
- Document Frequency (*df*): A number of documents contain a term. It gives measure of distribution of the term in document collection.
- Collection Frequency (*cf*): The sum of term frequency of a term appears in all of documents of the collection.

tf×*idf* weight:

- Inverse Document Frequency (*idf*) is a measure of general importance of the term. We can use the following formular to calculate *idf*:

$$idf = \frac{N}{df_t}$$

where N is the number of documents in collection, df_t is document frequency of term t.

- *tf* × *idf* meanings: If the term appears lots of times within a document (*tf* is high), that term is able to be a keyword in that document. However, if that term also appears lots of times in various documents (*df* is also high), then it may be a common term (less meaningful) such as "và" ("and"), "hoặc" ("or"), ... When *idf* has responsible of descending the score of that term. This weight can be calculated as followed:

$$w(t,d) = \begin{cases} \left(1 + \log\left(tf_{t,d}\right)\right) log \dfrac{N}{df_t}, & if\ tf_{t,d} \geq 1 \\ 0, & if\ tf_{t,d} = 0 \end{cases}$$

Latent Semantic Indexing (LSI).

LSI is an approach to index automatically by mapping documents and terms into *LSI* space. The mapping way bases on a concept, namely Singular Value Decomposition (*SVD*). *SVD* is the way to reduce multi-dimension data, reduce a big data so that the remainder only focuses on necessary data from original data. *LSI* solves two problems in vector space as synonymy and polysemy.

We perform to decompose term-document matrix into smaller matrixes. To do that, we use *SVD*: $C = U.S.V^T$ (where *C* namely a term-document matrix), where:

- *U* matrix is a matrix which each row is a vector of each term.
- V^T matrix is matrix which each column is a vector of each document.
- *S* matrix is a diagonal matrix with descending singular value, it gives importance of each dimension.

Reducing SVD: We only retain *k* singular value, the remainder values are assigned *0 (k < min(M, N))*. When *S* matrix is $k \times k$, *U* matrix is $M \times k$, V^T matrix is $k \times N$, C_k matrix in *LSI* space is $M \times N$. We use *SVD* to calculate C_k matrix. Calculating similarities on C_k matrix will be more reliable result than on *C* matrix.

Noun Phrase and Verb Phrase.

A Vietnamese noun phrase has overview of the structure as followed:

<previous sub-section> <center noun> <post sub-section>

Example: “mái tóc đẹp” (a beautiful hair), noun “tóc” (hair) is a center noun, “mái” is a previous sub-section, “đẹp” is post sub-section.

It notes that a noun phrase is able to be the lack of previous sub-section or post sub-section, but unable to be the lack of the center noun.

Based on this base and refer to [7], we chose the following patterns to extract Vietnamese noun phrases:

- *Noun + Noun*. Example: thủ_tướng chính_phủ (Prime Minister)
- *Noun + Noun + Noun*. Example: thống_đốc ngân_hàng nhà_nước (State Bank Governor)
- *Noun + Adjective*. Example: doanh_nghiệp lớn (a large enterprise)
- *Noun + Noun + Adjective*. Example: doanh_nghiệp tư_nhân lớn (a large private Enterprise)
- *Noun + Verb*. Example: giấy giới_thiệu (letter of recommendation)
- *Noun + Verb + Noun*. Example: dịch_vụ chứng_thực chữ_ký (signature authorization service)

The same as verb phrase, the overview of the structure as followed:

<previous sub-section> <center verb> <post sub-section>

In our using scope, we defined the following patterns to extract verb phrase:

- *Verb* + *Noun*. Example: đăng_ký doanh_nghiệp (enterprise registration)
- *Verb* + *Verb*. Example: thanh_toán chuyển_khoản (transfer payment)
- *Verb* + *Adjective*. Example: xử_lý nhanh (quick process)
- *Verb* + *Verb* + *Noun*. Example: đầu_tư phát_triển cơ_sở_hạ_tầng (invested in infrastructure development)

Besides, when extracting noun phrases and verb phrases, we also use the following rules:

- A phrase cannot start or end by stopwords, also contain special characters.
- A phrase has maximum length as 3.
- Using *idf* assigns score to the phrase to give the importance of the term in collection.

Based on documents were extracted on Vietnam Ministry of Justice's website, we extracted the set of 24771 two-terms phrases (both noun phrases and verb phrases) and 10360 three-terms phrases (both noun phrases and verb phrases).

Other Resources.

In limited period of time, we also made two files manually so that the system gives results better:

- The set of synonymy terms in law text.
- An ontology about lines of business.

4.2 System Architecture

The vLawyer system includes two components:

- Question processing: in this one, the system will convert the question to lowcases and remove stopwords, special words in user query. It also tokenizes, assigns POS tagger to query and extracts keyphrases/keywords to build into Lucene query to search relevant documents.
- Answer selection: Based on documents searched above phase, we extract articles to calculate similarities with user's query and incorporate some heuristics to choose the best answer. If the system cannot give the answer from the current corpus, it will push the query to Google search engine to find more documents to supplement in current corpus.

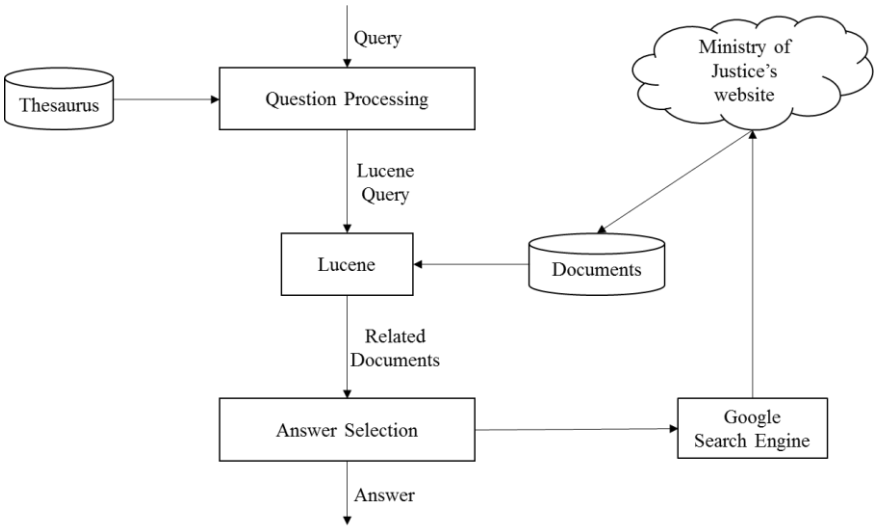

Fig. 1. System Architecture

Question Processing.

The query will be converted to lowcases and tokenized by VietTokenizer, assigned POS Tag by jvnTagger and removed stopwords, special words and question words. Extracting keyphrases, they are noun phrases or verb phrases have:

- The higher is score, the better is keyphrases.
- Priority for two-words phrases. If a three-words phrase contains a two-words phrase, but the two-words phrase has a higher score than the three-words phrase, we choose the two-words phrase is keyphrases.

The remaining is keywords. Afterward, we get synonymy terms of extracted terms. We build into Lucene query to search relevant documents.

Answer Selection.

We choose first top-ten documents which Lucene evaluates the best high score and perform to extract candidate passages in those documents. The candidate passage is able to be an article or a clause of the article or a small sentence of the clause of the article. They are extracted towards the following rules: When reading an article, the system calculates the similarity between the title of the article with the query, if the result is over a defined threshold, then it will extract the whole article as a candidate passage. Otherwise, it continues to read clauses 1, 2, ... of this article and each clause does the same as above to decide to choose that clause for candidate passage or not. If not, it will read smaller sentences of clause as a), b),... the process is the same as above. If it still does not choose any candidate passages, the article will be ignored and

continue to read other articles. The process's performed like that until the end of the document.

If the system cannot find any results from the set of documents returned by Lucene or extract any candidate passages from relevant documents. The query will be pushed to Google search engine to find the more relevant documents. Those documents will be supplemented to the current corpus.

If the system found the set of candidate passages, the system will build a term document matrix of candidate passages and the query, each element of that matrix is *tf*×*idf* weight of a term with respective candidate passage. Then, it calculates similarities between candidate passages and the query by cosin function and orders results by descending similarity scores.

Otherwise, the system reduces the threshold that was used to extract candidate passages and performs to choose candidate passages again. If they are found, the system will map them into *LSI* space and calculate similarities in this space. In this case, similarity calculation is in *LSI* space will be more reliable than normal vector space model.

Incorporating this result and some following heuristics, the system will choose the answer to return to user:

- Choosing the first top-ten results with highest similarity scores. The higher the similarity score is between candidate passage and query, the more it is an answer.
- The more the candidate passage contains the keyphrases/ keywords of the query, the more it is the answer.
- If two candidate passages contain the same as a number of keyphrases/keywords, the system will choose the candidate passage of the shorter length.

From those, we build the following formula to choose answers:

$$Ans(Ca) = \alpha \times sim(Ca, Q) + (1 - \alpha) \times \frac{\sum score(K_i) \times num(Ca, K)}{len(Ca)}$$

Where *Ca*: the candidate passage; *Q*: the user query; *K*: the list of keyphrases or keywords extracted from the user's query; *sim(Ca, Q)*: the similarity score's between the candidate passage and user query; *num(Ca, K)*: A number terms of *K* which *Ca* contains; $\sum score(K_i)$: the sum of scores of terms of *K* which *Ca* contains.

We will choose first top-three results of $Ans(Ca_i)$ score to give answers to user.

5 Experiment, conclusion and future works

5.1 Experiment

We extracted about 113 documents, including laws, decrees, circulars, decisions. All of these documents were indexed by Lucene.

Based on these documents, we built the set of 24771 two-terms phrases (both noun phrases and verb phrases) and 10360 three-terms phrases (both noun phrases and verb phrases). They were used to extracted keyphrases or keywords of user's query.

We also build manually two resources: synonymy terms and ontology-based in law text.

To illuminate for our system model, we gives a following example: it assumes that user wants to query a question as: "Việc đăng ký thay đổi người đại diện theo pháp luật của công ty trách nhiệm hữu hạn, công ty cổ phần được pháp luật quy định như thế nào?" ("How to register the replacement of the legal representative of limited liability companies or joint-stock companies?")

This query will be converted into lowercase to tokenize and tag Pos as followed:

Q = "việc/N đăng_ký/V thay_đổi/V người/N đại_diện/V theo/V pháp_luật/N của/E công_ty_trách_nhiệm_hữu_hạn/N ,/, công_ty_cổ_phần/N được/V pháp_luật/N quy_định/V như_thế_nào/X"

Afterward, it will be removed stopwords, special words, question words to extract keyphrases and keywords. The result as followed:

- Noun phrase: người đại_diện,
- Verb phrase: đăng_ký thay_đổi
- Terms: pháp_luật, công_ty_trách_nhiệm_hữu_hạn, công_ty_cổ_phần

The system continue to get the synonymy terms of above terms and build a query and push to Lucene as followed:
("người đại_diện"~1 OR "chủ_sở_hữu"~) AND "đăng_ký thay_đổi"~1 AND "pháp_luật"~ AND ("công_ty tnhh"~1 OR "công_ty_trách_nhiệm_hữu_hạn"~) AND ("công_ty_cổ_phần"~ OR "công_ty cp"~1)

With the current corpus, Lucene searches 6 relevant documents, including (in order of descending scores):

- Decree No. 43/2010/NĐ-CP: Decree on Enterprise Registration, score = 0.43
- Circular No. 01/2013/TT-BKHĐT: Circular on Business Registration, score = 0.42
- Law No. 60/2005/QH11: Law on Enterprise 2005, score = 0.39
- Decree No. 102/2010/NĐ-CP: Decree Detailing a number of Articles of the Law on Enterprises, score = 0.26
- Decree No. 05/2013/NĐ-CP: Decree on amending and supplementing some Articles in the regulations on Administrative procedures stated in the decree No. 43/2010/NĐ-CP, score = 0.22
- Decree No. 59/2009/NĐ-CP: Decree on organization and operation of commercial Banks, score = 0.11

Then, the system will extract candidate passages from these documents, the system extracts 50 candidate passages. They will be calculated the similarity score with user query, the result as followed (in order of descending score):

- Article 43 of Decree No. 43/2010/NĐ-CP: "Registration of change of owner of single member limited liability company", score = 0.431
- Article 17 of circular No. 01/2013/TT-BKHĐT: "Dossier for change of legal representatives of limited liability companies and shareholding companies", score = 0.2321
- Article 18 of circular No. 01/2013/TT-BKHĐT: "Registration for change of the owner of an LLC1 because of the inheritance", score = 0.2229

- Article 38 of Decree No. 43/2010/NĐ-CP: “Registration of change of legal representative of limited liability company or shareholding company”, score = 0.1991
- etc

The system will calculate other params in (*) and apply (*). As a result, it will give an answer to user as Article 17 of circular No. 01/2013/TT-BKHĐT and two reference answers: Article 18 of circular No. 01/2013/TT-BKHĐT and Article 43 of Decree No. 43/2010/NĐ-CP.

For testing, we got randomly 211 questions about law on enterprises and related law from reliable websites (such as Vietnam Ministry of Justice, vnexpress, ...) to test our system. We performed to match the result returned by our system and the answer of domain experts. We obtained the promising result, approximately 70% precision.

5.2 Conclusion and future works

In this paper, we presented an similarity-based approach for building the vLaywer system to answer questions about provisions, processes, procedures and sanctions in law on enterprises effectively. From [5][9], we also tried to build a ontology about lines of business manually. However, it's still small, so the results of some questions ,which related to establish lines of business, is not high. We will develop this in the future. Compare with [4][11] approaches, they also based on similarity between question and candidates or between question and other defined questions to choose answers for user's query, our obtained result is promising with the set of testing from internet randomly. With obtained results, it proved that our approach is reasonable.

We recognized if the set of synonymy terms and ontology in law text is larger, the result is higher. So, in the future, we will push more semantics into the system by developing ontology-based system and building synonymy corpus in legal documents automatically. Besides, we will broaden other laws and types of question.

References

1. Aatis Gospodnetíc, Erik Hatcher, “Lucene in Action”, Manning Publications Co., 2005.
2. Lê Thị Bích Ngọc, “Bài giảng: Pháp luật đại cương” , Posts and Telecommunications Institute of Technology, HCM city.
3. Vu Mai Tran, Vinh Duc Nguyen, Oanh Thi Tran, Uyen Thu Thi Pham, Thuy-Quang Ha (2009) - College of Technology, VNU Hanoi, VietNam: An Experimental Study of Vietnamese Question Answering System, 2009 International Conference on Asian Language Processing.
4. Mai-Vu Tran, Duc-Trong Le, Xuan-Tu Tran, Tien-Tung Nguyen - KTLab, University of Engineering And Technology: A Model of Vietnamese Person Named Entity Question Answering System, 26th Pacific Asia Conference on Language, Informaton and Computatioin pages 325-332.

5. Dai Quoc Nguyen, Dat Quoc Nguyen, Son Bao Pham (2009) - College of Technology Vietnam National University, HanoI: A Vietnamese Question Answering System, International Conference on Knowledge and Systems Engineering, 2009.
6. Tuoi T.Phan, Thanh C. Nguyen, and Thuy N.T. Huynh, CSE Faculty - HCMC University of Technology: Intelligent Inform and Database Systems, SCI 283, pp 29-40.
7. Đỗ Phúc (2006): "Research on application of frequent sets and association rules to semantic Vietnamese document classification", Science & Technology Development, Vol 9, No.2 – 2006.
8. Dang Truong Son, Dao Tien Dung (2013): Apply A Mapping Question Approach In Building The Question Answering System For Vietnamese Language, Journal of Engineering Technology and Education, The 2013 International Conference on Green Technology and Substainable Development.
9. Dang Tuan NGUYEN, Ha Quy-Tinh LUONG (2009): Document Searching System based on Natural Language Query Processing for Vietnam Open Courseware Library, IJCSI International Journal of Computer Science Issues, Vol. 6, No. 2, 2009.
10. Dat Quoc Nguyen, Dai Quoc Nguyen, Son Bao Pham (2009): A Knowledge-based QA System. In: Proc. Of KSE'09, IEEE CS (2009) 26-32.
11. Dang Truong Son, Dao Tien Dung (2012): Apply Mapping Question Approach in building the uestion answering system for Vietnamese language, Journal of Engineering Technology and Education – The 2012 International Conference on Green Technology and Substainable Development.
12. K.Saruladha, Dr.G.Aghila, Sajina Raj (2010): A New Semantic Similarity Metric for Solving Sparse Data Problem in Ontology based Information Retrieval System, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 11, May 2010.
13. Wen-tau Yih, Ming-Wei Chang, Christopher Meek, Andrzej Pastusiak (2013): Question Answering Using Enhanced Lexical Semantic Models, Microsoft Research, Updated Version – July 29, 2013.
14. Vietnam Ministry of Justice's website [Online] http://www.moj.gov.vn/Pages/home.aspx
15. Question Answering [Online] http://en.wikipedia.org/wiki/Question_answering
16. VLSP Project [Online] http://vlsp.vietlp.org:8080/demo/