Tran Khanh Dang
Josef Küng
Tai M. Chung
Makoto Takizawa (Eds.)

Communications in Computer and Information Science 1500

Future Data and Security Engineering

Big Data, Security and Privacy, Smart City and Industry 4.0 Applications

8th International Conference, FDSE 2021
Virtual Event, November 24–26, 2021
Proceedings

FDSE 2021

Springer

Communications in Computer and Information Science 1500

More information about this series at http://www.springer.com/series/7899

Tran Khanh Dang · Josef Küng · Tai M. Chung ·
Makoto Takizawa (Eds.)

Future Data and Security Engineering

Big Data, Security and Privacy, Smart City and Industry 4.0 Applications

8th International Conference, FDSE 2021
Virtual Event, November 24–26, 2021
Proceedings

Springer

Editors
Tran Khanh Dang
HCMC University of Technology (HCMUT)
Ho Chi Minh City, Vietnam

Josef Küng
Johannes Kepler University of Linz
Linz, Austria

Tai M. Chung
Sungkyunkwan University
Suwon, Korea (Republic of)

Makoto Takizawa
Hosei University
Tokyo, Japan

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-16-8061-8 ISBN 978-981-16-8062-5 (eBook)
https://doi.org/10.1007/978-981-16-8062-5

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

In LNCS volume 13076 and CCIS volume 1500 we present the accepted contributions for the 8th International Conference on Future Data and Security Engineering (FDSE 2021). The conference took place during November 24–26, 2021, in an entirely virtual mode (from Ho Chi Minh City, Vietnam). The proceedings of FDSE have been published in the LNCS and CCIS series by Springer. Besides DBLP and other major indexing systems, the FDSE proceedings have also been indexed by Scopus and listed in the Conference Proceeding Citation Index (CPCI) of Thomson Reuters.

The annual FDSE conference is a premier forum designed for researchers, scientists, and practitioners interested in state-of-the-art and state-of-the-practice activities in data, information, knowledge, and security engineering to explore cutting-edge ideas, to present and exchange their research results and advanced data-intensive applications, and to discuss emerging issues on data, information, knowledge, and security engineering. At FDSE, researchers and practitioners are not only able to share research solutions to problems of today's data and security engineering themes but are also able to identify new issues and directions for future related research and development work.

The two-round call for papers resulted in the submission of 168 papers. A rigorous peer-review process was applied to all of them. This resulted in 24 accepted papers (an acceptance rate of 14.3%) and two keynote speeches for LNCS volume 13076, and 36 accepted papers (including eight short papers, an acceptance rate of 21.4%) for CCIS volume 1500, which were presented online at the conference. Every paper was reviewed by at least three members of the International Program Committee, who were carefully chosen based on their knowledge and competence. This careful process resulted in the high quality of the contributions published in these two volumes. The accepted papers were grouped into the following sessions:

– Advances in Machine Learning for Big Data Analytics (LNCS)
– Big Data Analytics and Distributed Systems (LNCS and CCIS)
– Blockchain and Access Control (CCIS)
– Blockchain and IoT Applications (LNCS)
– Data Analytics and Healthcare Systems (CCIS)
– Machine Learning and Artificial Intelligence for Security and Privacy (LNCS)
– Security and Privacy Engineering (CCIS)
– Industry 4.0 and Smart City: Data Analytics and Security (LNCS and CCIS)
– Emerging Data Management Systems and Applications (LNCS)
– Short Papers: Security and Data Engineering (CCIS)

In addition to the papers selected by the Program Committee, four internationally recognized scholars delivered keynote speeches:

– Artur Andrzejak, Heidelberg University, Germany
– Johann Eder, Alpen-Adria-Universität Klagenfurt, Austria

– Tai M. Chung, Sungkyunkwan University, South Korea
– Thanh Thi Nguyen, Deakin University, Australia

The success of FDSE 2021 was the result of the efforts of many people, to whom we would like to express our gratitude. First, we would like to thank all authors who submitted papers to FDSE 2021, especially the invited speakers for the keynotes. We would also like to thank the members of the committees and additional reviewers for their timely reviewing and lively participation in the subsequent discussion in order to select such high-quality papers published in these two volumes. Last but not least, we thank the Organizing Committee members for their great support of FDSE 2021 even during the COVID-19 pandemic time.

November 2021

Tran Khanh Dang
Josef Küng
Tai M. Chung
Makoto Takizawa

Organization

Honorary Chair

Tomas Benz	Vietnamese-German University, Vietnam

Program Committee Chairs

Tran Khanh Dang	Ho Chi Minh City University of Technology, Vietnam
Josef Küng	Johannes Kepler University Linz, Austria
Tai M. Chung	Sungkyunkwan University, South Korea
Makoto Takizawa	Hosei University, Japan

Steering Committee

Dirk Draheim	Tallinn University of Technology, Estonia
Dinh Nho Hao	Institute of Mathematics, Vietnam Academy of Science and Technology, Vietnam
Fukuda Kensuke	National Institute of Informatics, Japan
Dieter Kranzlmüller	Ludwig Maximilian University of Munich, Germany
Fabio Massacci	University of Trento, Italy
Erich Neuhold	University of Vienna, Austria
Silvio Ranise	Fondazione Bruno Kessler, Italy
A Min Tjoa	Technical University of Vienna, Austria
Manuel Clavel	Vietnamese-German University, Vietnam

Publicity Committee

Nam Ngo-Chan	University of Warsaw, Poland
Tran Minh Quang	Ho Chi Minh City University of Technology, Vietnam
Le Hong Trang	Ho Chi Minh City University of Technology, Vietnam
Tran Tri Dang	RMIT University, Vietnam

Program Committee

Artur Andrzejak	Heidelberg University, Germany
Pham The Bao	Saigon University, Vietnam
Hyunseung Choo	Sungkyunkwan University, South Korea
Manuel Clavel	Vietnamese-German University, Vietnam
H. K. Dai	Oklahoma State University, USA
Vitalian Danciu	Ludwig Maximilian University of Munich, Germany

Quang-Vinh Dang	Industrial University of Ho Chi Minh City, Vietnam
Nguyen Tuan Dang	Saigon University, Vietnam
Tran Tri Dang	RMIT University, Vietnam
Thanh-Nghi Do	Can Tho University, Vietnam
Nguyen Van Doan	Japan Advanced Institute of Science and Technology, Japan
Johann Eder	Alpen-Adria-Universität Klagenfurt, Austria
Jungho Eom	Daejeon University, South Korea
Michael Felderer	University of Innsbruck, Austria
Fukuda Kensuke	National Institute of Informatics, Japan
Alban Gabillon	University of French Polynesia, France
Verena Geist	Software Competence Center Hagenberg, Austria
Osvaldo Gervasi	University of Perugia, Italy
Raju Halder	Indian Institute of Technology, Patna, India
Tran Van Hoai	Ho Chi Minh City University of Technology, Vietnam
Pham Thi Bach Hue	Ho Chi Minh City University of Science, Vietnam
Nguyen Quoc Viet Hung	Griffith University, Australia
Tran Manh Hung	Sungkyunkwan University, South Korea
Trung-Hieu Huynh	Industrial University of Ho Chi Minh City, Vietnam
Kien Huynh	Stony Brook University, USA
Kha-Tu Huynh	International University - VNU-HCM, Vietnam
Tomohiko Igasaki	Kumamoto University, Japan
Koichiro Ishibashi	University of Electro-Communications, Japan
Sungmin Jung	Myongji College, South Korea
M-Tahar Kechadi	University College Dublin, Ireland
Andrea Ko	Corvinus University of Budapest, Hungary
Lam Son Le	Ho Chi Minh City University of Technology, Vietnam
Duc Tai Le	Sungkyunkwan University, South Korea
Nhien-An Le-Khac	University College Dublin, Ireland
Cao Van Loi	Le Quy Don Technical University, Vietnam
Nadia Metoui	Delft University of Technology, The Netherlands
Hoang Duc Minh	National Physical Laboratory, UK
Nguyen Thai-Nghe	Can Tho University, Vietnam
Nam Ngo-Chan	University of Warsaw, Poland
Thanh Binh Nguyen	Ho Chi Minh City University of Technology, Vietnam
Binh Thanh Nguyen	International Institute for Applied Systems Analysis, Austria
Benjamin Nguyen	Institut National des Sciences Appliqués Centre Val de Loire, France
An Khuong Nguyen	Ho Chi Minh City University of Technology, Vietnam
Duy Ngoc Nguyen	Deakin University, Australia
Khoa Nguyen	CSIRO, Australia
Vu Thanh Nguyen	Ho Chi Minh City University of Food Industry, Vietnam
Truong Toan Nguyen	Curtin University, Australia
Trung Viet Nguyen	Can Tho University of Technology, Vietnam
Luong The Nhan	Amadeus IT Group, France

Alex Norta	Tallinn University of Technology, Estonia
Eric Pardede	La Trobe University, Australia
Cong Duc Pham	University of Pau, France
Vinh Pham	Sungkyunkwan University, South Korea
Nhat Hai Phan	New Jersey Institute of Technology, USA
Thanh An Phan	Ho Chi Minh City University of Technology, Vietnam
Nguyen Van Sinh	International University - VNU-HCM, Vietnam
Erik Sonnleitner	Johannes Kepler University Linz, Austria
Ha Mai Tan	National Taiwan University, Taiwan
Nguyen Hoang Thuan	RMIT University, Vietnam
Michel Toulouse	Hanoi University of Science and Technology, Vietnam
Ha-Manh Tran	Ho Chi Minh City University of Foreign Languages and Information Technology, Vietnam
Truong Tuan Phat Tran	Viettel High Technology Industries Corporation, Vietnam
Thien Khai Tran	Ho Chi Minh City University of Foreign Languages and Information Technology, Vietnam
Le Hong Trang	Ho Chi Minh City University of Technology, Vietnam
Tran Minh Triet	Ho Chi Minh City University of Science, Vietnam
Hai Truong	Singapore Management University, Singapore
Takeshi Tsuchiya	Tokyo University of Science, Japan
Le Pham Tuyen	Kyunghee University, South Korea
Le Thi Kim Tuyen	Heidelberg University, Germany
Hoang Huu Viet	Vinh University, Vietnam
Edgar Weippl	SBA Research, Austria
Wolfram Woess	Johannes Kepler University Linz, Austria
Honguk Woo	Sungkyunkwan University, South Korea
Kok-Seng Wong	VinUniversity, Vietnam
Sadok Ben Yahia	Tallinn University of Technology, Estonia
Szabó Zoltán	Corvinus University of Budapest, Hungary

Local Organizing Committee

Tran Khanh Dang	Ho Chi Minh City University of Technology, Vietnam
Josef Küng	Johannes Kepler University Linz, Austria
La Hue Anh	Ho Chi Minh City University of Technology, Vietnam
Nguyen Le Hoang	Ho Chi Minh City University of Technology, Vietnam
Ta Manh Huy	Ho Chi Minh City University of Technology, Vietnam
Nguyen Dinh Thanh	Ho Chi Minh City University of Technology, Vietnam

Additional Reviewers

Phuong Hoang Ai
Xuan Tinh Chu
Vipin Deval
Bhavya Gera
Trung Ha
Pham Nguyen Hoang Nam
Thi Ai Thao Nguyen
Manh-Tuan Nguyen
Le Hoang Nguyen
Chau D. M. Pham
Huy Ta
Cong Tran
Van Hau Tran
Tan Dat Trinh
Chibuzor Udokwu

Contents

Industry 4.0 and Smart City: Data Analytics and Security

Blockchain and Access Control

Data Analytics and Healthcare Systems

Short Papers: Security and Data Engineering

Proposing Recommendation System Using Bag of Word and Multi-label Support Vector Machine Classification

Phat Nguyen Huu[1(✉)], Tuan Nguyen Anh[1], Hieu Nguyen Trong[2], Pha Pham Ngoc[2], and Quang Tran Minh[3,4]

[1] Hanoi University of Science and Technology (HUST), Hanoi, Vietnam
phat.nguyenhuu@hust.edu.vn, tuan.na172900@sis.hust.edu.vn

[2] National Institute of Patent and Technology Exploitation (NIPTECH), Hanoi, Vietnam
{nthieu,pnpha}@most.gov.vn

[3] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet, Dist.10, Ho Chi Minh City, Vietnam
quangtran@hcmut.edu.vn

[4] Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Abstract. Currently, recommendation systems (RS) have attracted a great attention from researchers both in academic and industry. They help extracting useful information quickly to provide right services to users in accordance to there basic requests which are commonly not too details. In the field of higher education and academic research, proposing project/thesis title that has already been developed to the right students/researcher for reference is a new and challenging issue. Therefore, we propose to build a system to suggest title of graduation project in the paper. In proposal system, we use natural language processing (NLP) that is applied artificial neural network (ANN) to solve feature extraction and training problems. The simulation results show that the proposal system achieves an accuracy of 82% with 12 s. The results also show that proposal system is suitable for applying in real environment.

Keywords: Recommendation system · Multi-label classification · Bag of word · Deep learning · Natural language processing

1 Introduction

In recent times, Internet has evolved into platform for large-scale online services. It profoundly changed the way that we communicate, read news, go shopping, and watch movies. Today, large amount of data (movies, news, books, goods, etc.) is transmitted online and recommendation system can help us find them quickly. Therefore, it is a powerful information filtering tool that can promote personalized services and provide an unique experience for each user.

T. K. Dang et al. (Eds.): FDSE 2021, CCIS 1500, pp. 276–289, 2021.
https://doi.org/10.1007/978-981-16-8062-5_19

Nowadays, recommendation systems are widespread that are central component of many online service providers such as Netflix movie, Amazon product, or YouTube video. It helps to reduce searching effort and minimize information overload. However, recommendation system is starting to gain a foothold in e-commerce platforms in recent years when education is still new in Vietnam.

Today, achieving higher education is a trend as it is an essential demand for development in the industry 4.0 society. Students before completing study program at the graduate school need to perform a thesis or project. Consequently, it is necessary for students to obtain many reference topics to make a good thesis or project. Therefore, we propose to build a system to recommend thesis/project title in the paper.

There are two main points of our system based on [21] as follows:

1. First, we create a separating dataset of keywords for each topic name corresponding to cases that help us to input data.
2. Secondly, we combine two methods (bag of word (BOW) [1–3,17] and multi-label classification [5,11,20]) for feature extraction and training.

The rest of the paper is presented as follows. In Sect. 2, we will present related work. In Sect. 3 and 4, we present and evaluate the effectiveness of the proposed model, respectively. Finally, we give conclusion in Sect. 5.

2 Related Work

Recently, there have been a lot of studies relating to recommendation system [6,14,15,21,22,24]. In [22], the authors present book recommendation system based on tagging of entities mention. As a result, they constructed a sequence of entities to reference against each other and rank them based on term frequency - inverse document frequency (TF-IDF) technique [19]. These rankings will be used to find similarities among books.

The recommendation system [21] creates website to use collaborative learning based on information that users provide such as hobbies and books. The system has great advantage of fast processing speed and simplicity. However, its accuracy is only 77%. In [14], a model is built based on aspects of past behavior including items that user has previously purchased as well as rating giving for particular book. The authors [9] use more similarity comparison among books such as cosine, ppc, cpcc and jaccard. The best results of method is to use CPCC comparison method with accuracy of 61%. In addition to research relating to book recommendation system, we also expand search for other studies in [6,15,24].

The authors [6] build restaurant recommendation system based on input data such as restaurant amenities and customer comments with voting rates from 1* to 5*. In [24], the authors built place recommendation system based on input context providing directly from user (hobbies, interests, mood etc.) or environment (time, weather, current location) to solve traveling problem. They then use matrix decomposition techniques to predict outcomes. In [27], the author uses

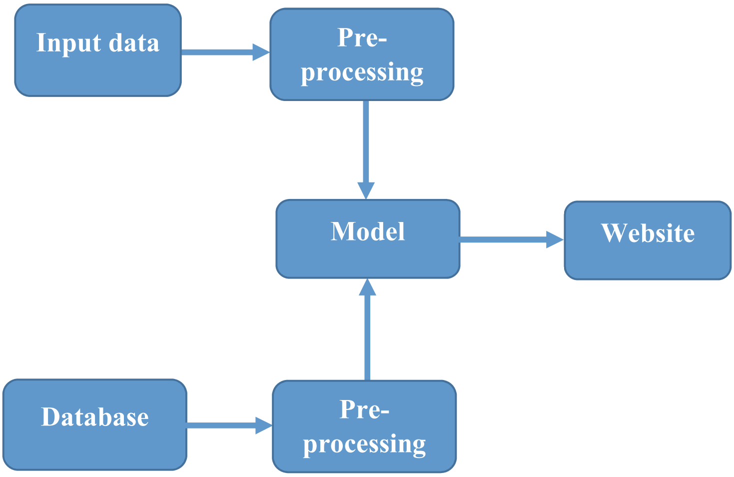

Fig. 1. Overview of proposal system model.

SSVM-based Heuristic method based on SVM to reduce sparsity of user-item matrix. The accuracy of results is 69.7% with a dataset of up to 43000 users and 3500 different movies. In [15], authors use long short-term memory (LSTM) to recommend movies and compare it with other models such as recurrent neural network (RNN) ($dropout = 0.2$) and k-nearest neighbors (KNN). As a result, LSTM gives the best results with accuracy of 70.5%.

As the above analysis, we see that algorithms only filter collaboratively and have not solved the accuracy problem for many cases. The accuracy of algorithms above is only 61 to 78%. Therefore, we propose to BOW algorithm and multi-label support vector machine (SVM) to solve the problems.

3 System Design

3.1 Overview of Proposal System

The system overview is shown Fig. 1. The system will perform as follows:

1. Input data block will receive and transfer them to prepossessing.
2. Database block contains all the subject names that have been entered.
3. Model block performs to convert data from characters to numbers and compares them each other. The results are shown on website.

We propose to change two main points of training and prediction block based on [21]. More detailed of proposal system model is shown in Fig. 2 as follows:

1. First, we use BOW algorithm to generate keywords based on suggested topic names. We then preprocess and compare them with data of topic name and keywords.

Fig. 2. Details of steps of proposal system model.

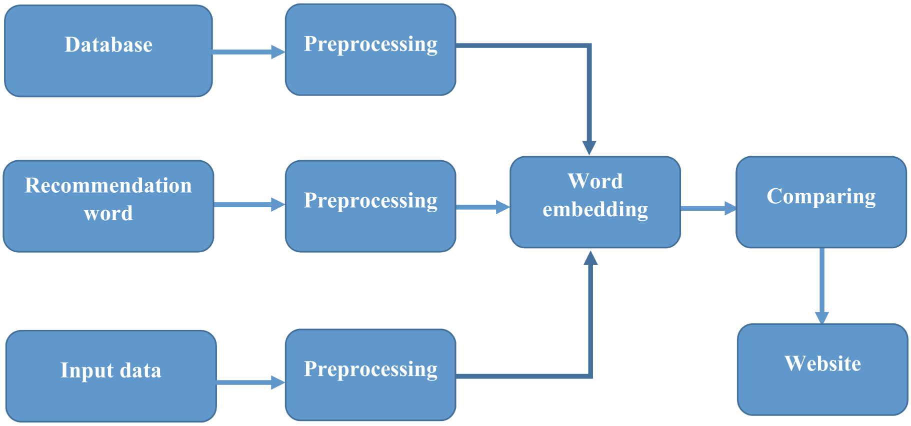

Fig. 3. Developing model on website.

2. Secondly, we will apply multi-label SVM after being trained to suggest topic names based on keywords requesting by users.

Our main contribution is the combination of BOW and multi-Label SVM and construction of dataset for that is suitable for names of graduation projects as shown in Fig. 2. More details of the steps will be presented below.

3.2 Building Model

The block diagram of model is shown in Fig. 3 that consists of following blocks as follows

1. **Database block** performs to store topic names that lecturer has added to class.
2. **Processing block** performs data cleaning steps.
3. **Word embedding block** converts sentences and words into numbers.
4. **Comparing block** compares input data and database. If there is a similarity, it will display the content.
5. **Website block** will display similar topics.

3.3 Training Process

In section, we will present detail of training process to create, preprocess, and train datasets.

Creating Keyword. When there are topic names, we will assign keywords that users can input on website. If we want to find the title of topic "designing smart home system to control by voice", we will input the keyword as smart home, voice, voice interaction, deep learning, or NLP. Table 1 is an example of keywords generating based on available topic names.

Table 1. An example of building data for training process.

No.	Topic name	Keyword
1	Smart home system design interactive	Voice, home, smart
2	Design and build a website using PHP	Website, PHP, design
3	Research on image processing on self-driving cars	Processing, images, cars
4	System design intelligent temperature and humidity	System, intelligent, sensor
5	Designing an incubator system that integrates	IoT, agriculture, animals
6	Design and manufacture anti-snoring pillow	Fabricated, anti-snoring, pillow
7	Design an application model to control by wifi	Application, model, control

Preprocessing Data. Preprocessing data is a very important step to solve any problem of NLP. Most of datasets using for machine learning and language processing need to be processed, cleaned, and transformed before training.

1. Word separation
   Word separation is a processing process that aims to determine boundaries of words of sentence. It is process of identifying single and compound words of sentence. It is necessary to determine grammatical structure of words for language processing. The problem is simple with humans. However, it is a very difficult problem to solve for computers. In the paper, we use tool Vitokenizer() [25] to separate words. The separation process is shown in Fig. 4.
2. Stopword
   Stopwords are not meaning since they are able to be omitted without effecting sentence. They are short and most common words. In case, stopwords can cause problems while finding them. There are two main ways to remove stopwords. First way is using a dictionary. Second way is based on the frequency

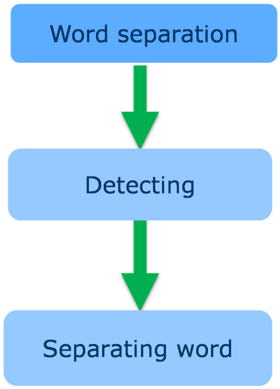

Fig. 4. The steps of word separation process.

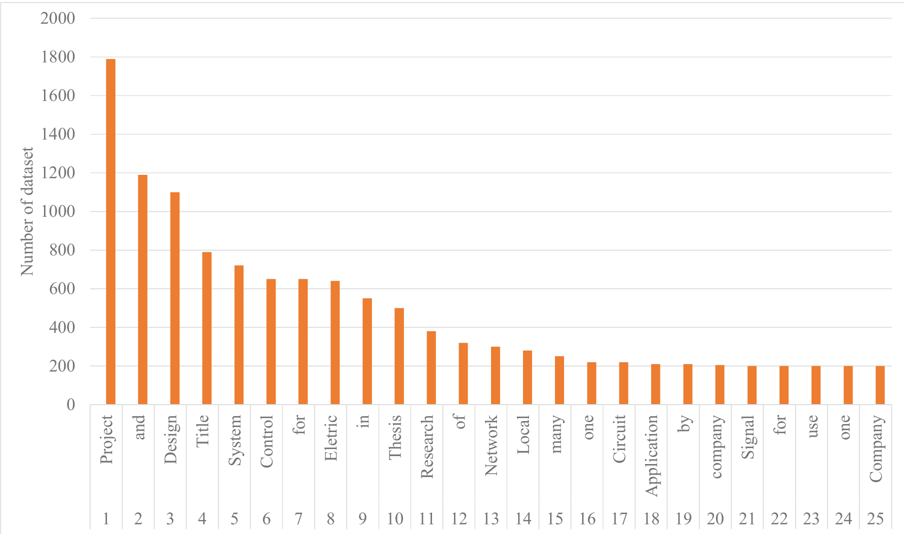

Fig. 5. Result of removing stopword based on word frequency.

of their occurrence. In second way, we count number of their occurrences and then remove the words that appear many times. We recognize that they do not have much meaning.

To perform for stopword algorithm with self-built data, the results as shown in Fig. 5. In Fig. 5, we can see that the words (project, design, and) appear many times. However, they do not mean for machine learning since we are able to remove them.

Currently, there are many ways to find these words. We are able to use Vietnamese-stopword dictionary to remove unnecessary Vietnamese words. Although it is important to remove stopwords, the keywords are usually very short since we do not remove them.

3. Feature extraction
   When we train the machine learning model for NLP, data is in form of text. Therefore, we are not able to feed data to train directly since models only perform on numbers or matrices or vectors.
   In the paper, we use embedding techniques to extract feature. TF-IDF, BOW, and encoder-decoder are used for recurrent neural network (RNN) [8,12,28] or LSTM [18,29]. Besides, we use Word2vec algorithm to learn word embeddings from large datasets.
   TF-IDF is the most widely known statistical method for determining importance of word or dataset. We first need to calculate the frequency of word as

$$tf\,(t,d) = \frac{f_{(t,d)}}{\sum_{t' \in d} f_{t',d}}, \tag{1}$$

   where $f(t,d)$ is number of word occurrences (t) of dataset (d) and $\sum_{t' \in d} f_{t',d}$ is total number of words of dataset (d).
   Next, it is necessary to calculate importance of word according to expression

$$idf\,(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}, \tag{2}$$

   where N is total number of data of dataset $N = |D|$ and $|\{d \in D : t \in d\}|$ is number of data that t appears.
   Finally, we have:

$$tfidf\,(t,d,D) = tf\,(t,d) \times idf\,(t,D)\,. \tag{3}$$

   Words with high TF-IDF values often appear many times of text. This helps to filter out common words and retain high-value.
   Word2vec was one of the first models of word embedding using neural networks. It has ability to be vectorized each word based on set of keyword and context. Word2vec is the mapping of vocabulary to vector space where each of them is represented by n real numbers. Each word corresponds to fixed vector. Their weights are updated continuously after training model using backprobagation algorithm. The model is shown in Fig. 6 [13,16].
   In Fig. 6, input is a one-hot-vector where each word will have form $x_1, x_2, ..x_v$ and v is vocabulary number. It is a vector where each word will have value of "1" equivalent to one in vocabulary and other will be "0".
   The dimension between input and hidden layer is matrix W $(V \times N)$ whose active function is linear and weight between hidden. Output is W' $(N \times V)$ and their activation function is softmax.
   Each row of W is N-dimensional vector representing v_w for input layer. Each row of W is v_w^T calculating as follows

$$h = W^T x = v_w^T. \tag{4}$$

   We calculate the score u_i for each word of vocabulary based on hidden layer to output as $W' = w'_{i,j}$ matrix where v_{wj} is j column vector of W'.

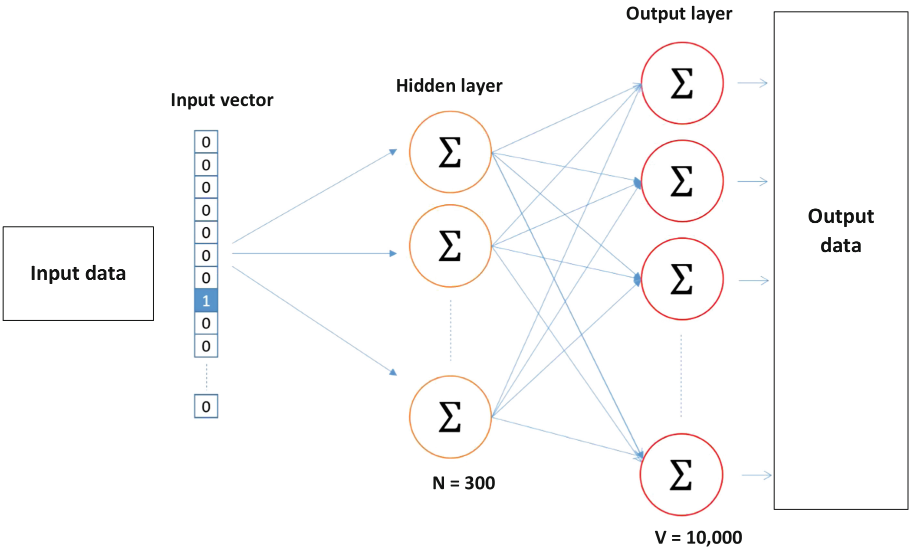

Fig. 6. Describing Word2vec model.

	home	smart	controlled	by	voice	device	through	gestures	in	controll
BoW1	1	1	1	1	1	0	0	0	0	0
BoW2	1	1	0	0	0	1	1	1	1	1

Fig. 7. Vector after word splitting using BOW.

We then use the softmax function as

$$P_{(w_j|w_I)} = y_i = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})}, \tag{5}$$

$$u_j = \nu'^{T}_{wj} v_{w_I}, \tag{6}$$

$$u_{j'} = \nu'^{T}_{wj'} v_{w_I}, \tag{7}$$

where v_w and v'_w are two vectors representing w word from W and W' matrices, respectively.

In the section, we use BOW algorithm since it has a faster calculation speed than other methods. The model is simplified representation using for NLP and information retrieval. In model, a text is represented as multi-set that is not meaning with diversity [7,23]. We see an example of BOW as shown in Fig. 7.

We have two topics, namely smart home controlling by voice and device controlling through gestures in smart home. Based on creating dictionary, we proceed to generate a vector that stores number of word occurrences in dictionary for each sentence. Since the dictionary has 14 words, each vector will have 14 elements as shown in Fig. 7.

If a word of dataset appears N times for sentence, its vector will be N. In BOW1, the word "home" appears once time since its vector will be 1. Otherwise, the word "gestures" does not appear in BOW1, since its vector will be "0".

Training Data. Currently, there are many models for training in typical NLP such as RNN, LSTM. Due to specificity of problem, a keyword has many topic names since we use multi-label classification SVM [4,5]. The description of multi-label is as follows.

We have

$$L = \{\omega_j : j = 1 \ldots q\}, \tag{8}$$

that is used to represent finite number of labels.

We then have:

$$D = \{(x_i; Y_i),\ i = 1 \ldots n\}, \tag{9}$$

to represent training cases where x_i are feature vectors and $Y_i \in L$ is set of labels of i.

The set of labels is defined as binary vector

$$Y_i = \{y_1, y_2, \ldots, y_q\}. \tag{10}$$

The training algorithm is shown in Fig. 8. Input data is the topic names that have relating keywords. They will then be transferred into preprocessing step. These topic names will be converted to vector corresponding to keywords. If the keyword is relevant to topic, it will be 1. Otherwise, it will be 0.

In the step, we continue to process those keywords by word and feature extraction algorithms. We split the dataset into 20% for testing and 80% for training. In next step, we use the multi-label classification SVM model for training. After receiving training results, we will test the model with dataset.

4 Simulation and Result

4.1 Setup

In our model, we use accuracy function to evaluate similarity between two items (Y_i and Z_i) based on [5,10,26] as

$$Accuracy\,(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}, \tag{11}$$

where N is the number of classes.

We use a device configuring with Intel Xeon processor 2 cores, 2.20 GHz and 13 GB of RAM while evaluating accuracy.

4.2 Result

To perform proposal algorithm, we used 157 topics that we collected and built ourselves. When the number of threads increases, processing time will change. We compare the accuracy and training time and prediction of proposal model with others. The results are shown in Table 2 and Fig. 9.

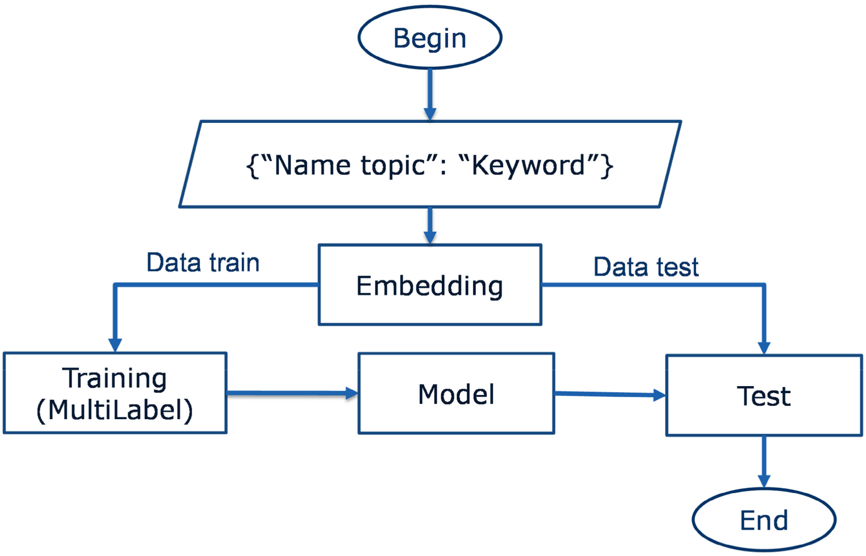

Fig. 8. Flowchart of training algorithm using multi-Label classification SVM.

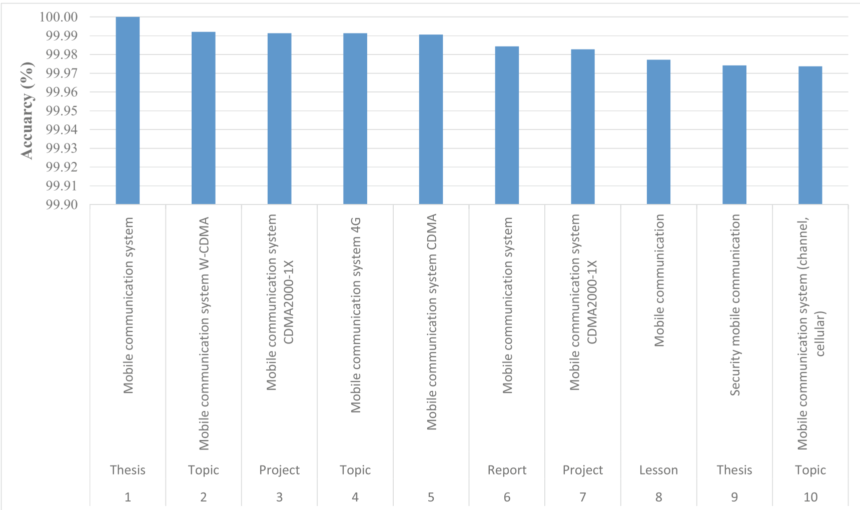

Fig. 9. Accuracy classification results by topic.

Table 2. Comparing accuracy of proposal with other models.

Model	Average accuracy (%)	Training and predicting evaluation
Proposal	82	12 and 0.1 s for training and predicting
Filtering [10]	77	89% for speed of getting recommendations
LSTM [15]	70.5	0.706 (F1-Score@20)

In Table 2, we see that when using proposal method (BOW + multi-label), we improve accuracy up to 82%. The result is higher than 5% comparing with [10] and 10% with LSTM [15].

4.3 Deploying on Website

To test effectiveness of algorithm when running for practice, we deploy on self-designing website. The results are shown in Fig. 10.

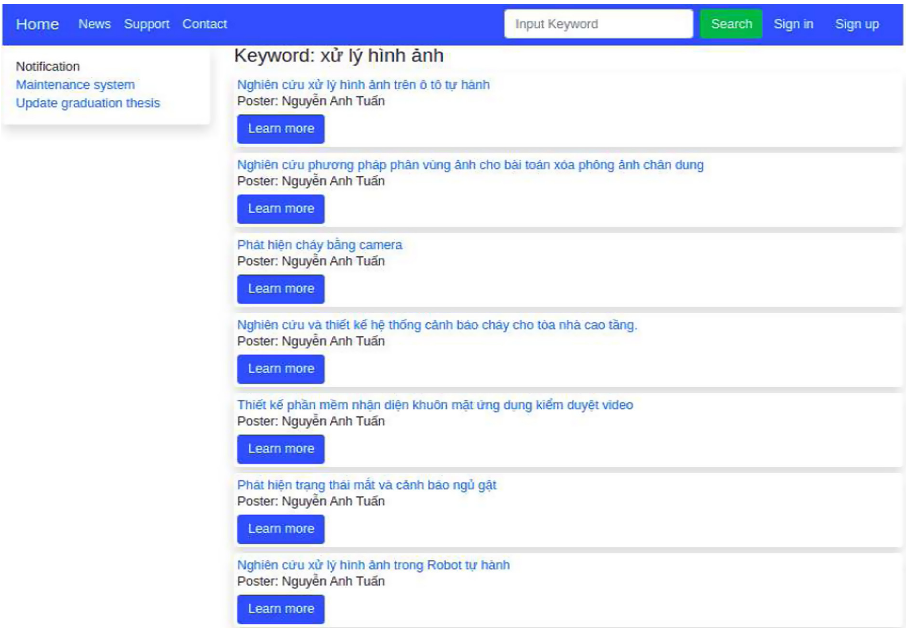

Fig. 10. Result of developing algorithm on website.

Figure 10 shows the results with keyword "image processing". The results of topic names are typically "Fire detection by camera", "design of face recognition software for video censorship", etc. Device configuration using for testing on website platform is Intel core i5-6300 CPU 2.30 GHz and 8 Gb RAM.

5 Conclusion

In the paper, we create dataset for each topic and develop algorithm on website. The results show that accuracy of system is improved up to 82%. However, the system still has limitation that it has not received all keywords for topics.

Therefore, we will continue to improve the dataset by adding more keywords for topics and try with other machine learning models for the next direction.

Acknowledgment. This research is carried out in the framework of the project funded by the Ministry of Science and Technology (MOST), Vietnam under the grant 04.2020M008. The authors would like to thank the MOST for the support.

References

1. Alahmadi, A., Joorabchi, A., Mahdi, A.E.: A new text representation scheme combining bag-of-words and bag-of-concepts approaches for automatic text classification. In: 2013 7th IEEE GCC Conference and Exhibition (GCC), pp. 108–113 (2013). https://doi.org/10.1109/IEEEGCC.2013.6705759
2. Alahmadi, A., Joorabchi, A., Mahdi, A.E.: Combining bag-of-words and bag-of-concepts representations for Arabic text classification. In: 25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), pp. 343–348 (2014). https://doi.org/10.1049/cp.2014.0711
3. Ali, N.M., Jun, S.W., Karis, M.S., Ghazaly, M.M., Aras, M.S.M.: Object classification and recognition using bag-of-words (bow) model. In: 2016 IEEE 12th International Colloquium on Signal Processing Its Applications (CSPA), pp. 216–220 (2016). https://doi.org/10.1109/CSPA.2016.7515834
4. Dolly, C., Vivian, B., María, M.G.: Multi-label classification for recommender systems. Adv. Intell. Syst. Comput. **221**, 181–188 (2013)
5. Fu, D., Zhou, B., Hu, J.: Improving SVM based multi-label classification by using label relationship. In: 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–6 (2015). https://doi.org/10.1109/IJCNN.2015.7280497
6. Gomathi, R., Ajitha, P., Krishna, G.H.S., Pranay, I.H.: Restaurant recommendation system for user preference and services based on rating and amenities. In: 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), pp. 1–6 (2019). https://doi.org/10.1109/ICCIDS.2019.8862048
7. Harri, Z.S.: Distributional struct. Word **10**(2-3), 146–162 (1954). https://doi.org/10.1080/00437956.1954.11659520
8. Kaur, M., Mohta, A.: A review of deep learning with recurrent neural network. In: 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 460–465 (2019). https://doi.org/10.1109/ICSSIT46314.2019.8987837
9. Kommineni, M., Alekhya, P., Vyshnavi, T.M., Aparna, V., Swetha, K., Mounika, V.: Machine learning based efficient recommendation system for book selection using user based collaborative filtering algorithm. In: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), pp. 66–71 (2020). https://doi.org/10.1109/ICISC47916.2020.9171222
10. Kurmashov, N., Latuta, K., Nussipbekov, A.: Online book recommendation system, pp. 1–4 (September 2015). https://doi.org/10.1109/ICECCO.2015.7416895

11. Liu, G., Zhang, X., Zhou, S.: Multi-class classification of support vector machines based on double binary tree. In: 2008 Fourth International Conference on Natural Computation, vol. 2, pp. 102–105 (2008). https://doi.org/10.1109/ICNC.2008.536
12. Liu, T., Wu, T., Wang, M., Fu, M., Kang, J., Zhang, H.: Recurrent neural networks based on lstm for predicting geomagnetic field. In: 2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), pp. 1–5 (2018). https://doi.org/10.1109/ICARES.2018.8547087
13. Ma, L., Zhang, Y.: Using word2vec to process big text data. In: 2015 IEEE International Conference on Big Data (Big Data), pp. 2895–2897 (2015). https://doi.org/10.1109/BigData.2015.7364114
14. Mathew, P., Kuriakose, B., Hegde, V.: Book recommendation system through content based and collaborative filtering method. In: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), pp. 47–52 (2016). https://doi.org/10.1109/SAPIENCE.2016.7684166
15. Nguyen, S., Tran, B.: Long short-term memory based movie recommendation. Sci. Technol. Dev. J. Eng. Technol. **3**(SI1), SI1-SI9 (2020). https://doi.org/10.32508/stdjet.v3iSI1.540
16. Phat, H.N., Anh, N.T.M.: Vietnamese text classification algorithm using long short term memory and word2vec. Inf. Autom. **19**(6), 1255–1279 (2020). https://doi.org/10.15622/ia.2020.19.6.5
17. Pu, W., Liu, N., Yan, S., Yan, J., Xie, K., Chen, Z.: Local word bag model for text categorization. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 625–630 (2007). https://doi.org/10.1109/ICDM.2007.69
18. Pulver, A., Lyu, S.: Lstm with working memory. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 845–851 (2017). https://doi.org/10.1109/IJCNN.2017.7965940
19. Qaiser, S., Ali, R.: Text mining: Use of TF-IDF to examine the relevance of words to documents. Int. J. Comput. Appl. **181**(1), 25–29 (2018). https://doi.org/10.5120/ijca2018917395
20. Qin, Y.P., Wang, X.K.: Study on multi-label text classification based on SVM. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 1, pp. 300–304 (2009). https://doi.org/10.1109/FSKD.2009.207
21. Rana, A., Deeba, K.: Online book recommendation system using collaborative filtering (with Jaccard similarity). J. Phys. Conf. Ser. **1362**, 012130 (2019). https://doi.org/10.1088/1742-6596/1362/1/012130
22. Sariki, T., Kumar, B.: A book recommendation system based on named entities. Ann. Libr. Inf. Stud. **65**, 77–82 (2018)
23. Sivic, J., Zisserman, A.: Efficient visual search of videos cast as text retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **31**(4), 591–606 (2009). https://doi.org/10.1109/TPAMI.2008.111
24. Thien, L.C., Nghe, N.T.: An approach for building context-aware recommender systems (July 2015). https://doi.org/10.15625/vap.2015.000185
25. Tran, V.T.: Pyvi 2021. Accessed 15 May 2021. https://github.com/trungtv/pyvi
26. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Int. J. Data Warehouse. Min. **3**, 1–13 (2009). https://doi.org/10.4018/jdwm.2007070101
27. Xia, Z., Dong, Y., Xing, G.: Support vector machines for collaborative filtering. In: Proceedings of the 44th Annual Southeast Regional Conference. p. 169–174. ACM-SE 44, Association for Computing Machinery, New York (2006). https://doi.org/10.1145/1185448.1185487

28. Yang, T.H., Tseng, T.H., Chen, C.P.: Recurrent neural network-based language models with variation in net topology, language, and granularity. In: 2016 International Conference on Asian Language Processing (IALP), pp. 71–74 (2016). https://doi.org/10.1109/IALP.2016.7875937
29. Yao, L., Guan, Y.: An improved lstm structure for natural language processing. In: 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), pp. 565–569 (2018). https://doi.org/10.1109/IICSPI.2018.8690387

Author Index