THE 2016 IEEE RIVF INTERNATIONAL CONFERENCE ON COMPUTING & COMMUNICATION TECHNOLOGIES
RESEARCH, INNOVATION, AND VISION FOR THE FUTURE (RIVF)
Hanoi, November 7-9, 2016, Thuyloi University, Vietnam

# FOURTH INTERNATIONAL WORKSHOP ON VIETNAMESE LANGUAGE AND SPEECH PROCESSING (VLSP 2016)

## Proceedings of the Workshop

**November 9, 2016**
**Hanoi, Vietnam**

# VLSP 2016 - Fourth International Workshop on Vietnamese Language and Speech Processing

**In conjunction with 12th IEEE - RIVF Conference**

***Time*** 11h00 – 15h45, Nov 9, 2016
***Location*** Thuyloi University
175 Tay Son Street, Dong Da District, Hanoi, Vietnam

### *Workshop Co-chairs*

Le Anh Cuong, Ton Duc Thang University, Ho Chi Minh city, Vietnam
Nguyen Thi Minh Huyen, VNU University of Science, Hanoi, Vietnam

### *Program Committee*

1. Dinh Dien, University of Science, VNU-HCM, Vietnam (co-chair)
2. Luong Chi Mai, Institute of Information Technology, VAST, Vietnam (co-chair)
3. Cao Hoang Tru, University of Technology, VNU-HCM, Vietnam
4. Ho Bao Quoc, University of Science, VNU-HCM, Vietnam
5. Ho Tu Bao, JAIST, Japan
6. Le Anh Cuong, Ton Duc Thang University, Ho Chi Minh city, Vietnam
7. Le Hong Phuong, University of Science, VNU, Hanoi, Vietnam
8. Le Thanh Huong, Hanoi University of Science and Technology, Vietnam
9. Ngo Xuan Bach, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
10. Nguyen Le Minh, JAIST, Japan
11. Nguyen Phuong Thai, University of Engineering and Technology, VNU, Hanoi, Vietnam
12. Nguyen Thi Minh Huyen, University of Science, VNU, Hanoi, Vietnam
13. Nguyen Van Vinh, University of Engineering and Technology, VNU, Hanoi, Vietnam
14. Nguyen Viet Cuong, HPC Systems, Inc., Japan
15. Nguyen Viet Son, Hanoi University of Science and Technology, Vietnam
16. Pham Bao Son, University of Engineering and Technology, VNU, Hanoi, Vietnam
17. Phan Thi Tuoi, University of Technology, VNU-HCM, Vietnam
18. Phan Xuan Hieu, University of Engineering and Technology, VNU, Hanoi, Vietnam
19. Tran Do Dat, Ministry of Science and Technology, Vietnam
20. Vu Hai Quan, University of Science, VNU-HCM, Vietnam
21. Vu Tat Thang, Institute of Information Technology, VAST, Vietnam

***Workshop Program***

10h45-11h00: Registration
11h00-11h05: Opening
11h05-11h20: Report on Named-Entity Recognition Evaluation Campaign: Data and Systems, Nguyễn Thị Minh Huyền, Vũ Xuân Lương
11h20-11h30: Vietnamese Named Entity Recognition using Token Regular Expressions and Bidirectional Inference, Lê Hồng Phương
11h30-11h40: Named Entity Recognition in Vietnamese Text Using Conditional Random Fields, Vũ Anh, Lê Minh Tuấn, Nguyễn Việt Hưng
11h40-11h55: DSKTLAB-NER: Nested Named Entity Recognition in Vietnamese Text, Nguyễn Thị Cẩm Vân, Phạm Thái Sơn, Vương Thị Hồng, Nguyễn Ngọc Vũ and Trần Mai Vũ
11h55-12h10: System Demo: VAIS-TTS: A High Quality Speech Synthesis Service for Vietnamese, Đỗ Quốc Trường, Lương Chi Mai
12h10-12h25: System demo: Instrumentations controlled by Vietnamese voice, Nguyễn Việt Sơn, Nguyễn Việt Tùng, Đỗ Thị Ngọc Diệp, Mạc Đăng Khoa

Lunch

14h00-14h15: Report on Sentiment Analysis Evaluation Campaign: Data and Systems, Lê Anh Cường, Nguyễn Thị Minh Huyền, Nguyễn Việt Hùng
14h15-14h25: DSKTLAB: Vietnamese Sentiment Analysis for Product Reviews, Phạm Thị Quỳnh Trang, Nguyễn Xuân Trường, Trần Văn Hiến, Nguyễn Thị Chăm and Trần Mai Vũ
14h25-14h35: A Simple Supervised Learning Approach to Sentiment Classification at VLSP 2016, Hy Nguyen, Tung Le, Viet-Thang Luong and Dien Dinh
14h35-14h50: System Demo: Technical report for land market information system, Duong Nguyen Thanh, Ha Nguyen and Tu Nguyen
14h50-15h45: Panel Discussion and Closing

# Table of Contents

# VLSP 2016 Shared Task: Named Entity Recognition

Nguyễn Thị Minh Huyền
Faculty of Mathematics, Mechanics and Informatics
VNU University of Science
Hanoi, Vietnam
huyenntm@hus.edu.vn

Vũ Xuân Lương
Vietnam Lexicography Center
Hanoi, Vietnam
vuluong@vietlex.com

***Abstract*—The 4th International Workshop on Vietnamese Language and Speech Processing (VLSP 2016) organizes the shared task of Named Entity Recognition at the first time for Vietnamese language processing. This campaign aims to bring together researchers interested in this topic as well as provides a benchmark dataset for this task. We received six system submissions with various methods and promising results.**

## I. Introduction

Named entities (NE) are phrases that contain the names of persons, organizations, locations, times and quantities, monetary values, percentages, etc. Named Entity Recognition (NER) is the task of recognizing named entities in documents. NER is an important subtask of Information Extraction, which has attracted researchers all over the world since 1990s.

From 1995, the 6th *Message Understanding Conference*– MUC[1] has started evaluating NER systems for English. Besides NER systems for English, NER systems for Dutch and Turkish were also evaluated in CoNLL 2002[2] and CoNLL 2003[3] Shared Tasks. In these evaluation tasks, four named entities were considered, consisting of names of persons, organizations, locations, and names of miscellaneous entities that do not belong to the previous three types. Recently, there have been some competitions about NER organized, e.g. The GermEval 2014 NER Shared Task[4].

For Vietnamese language, so far there is no systematic comparison between the performances of Vietnamese NER systems. The VLSP 2016 campaign, therefore, targets at providing an objective evaluation measurement about the quality of NER tools, and encouraging the development of NER systems with high accuracy.

## II. Task description

### A. Problem

The scope of the campaign this year is to evaluate the ability of recognizing NEs in three types, i.e. names of persons, organizations, and locations. Recognizing of other types of NEs will be covered in next campaigns.

[1] http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html
[2] http://www.clips.uantwerpen.be/conll2002/ner/
[3] http://www.cnts.ua.ac.be/conll2003/ner/
[4] https://sites.google.com/site/germeval2014ner/

### B. Data Preparation

*1) Data collection:* Data are collected from electronic news papers published on webs. Three types of NEs are compatible with their descriptions in the CoNLL Shared Task 2003.

***Locations***

- roads (streets, motorways)
- trajectories
- regions (villages, towns, cities, provinces, countries, continents,dioceses, parishes)
- structures (bridges, ports, dams)
- natural locations (mountains, mountain ranges, woods,rivers, wells, fields, valleys, gardens,nature reserves, allotments, beaches,national parks)
- public places (squares, opera houses, museums, schools,markets, airports, stations, swimming pools,hospitals, sports facilities, youth centers,parks, town halls, theaters, cinemas, galleries,camping grounds, NASA launch pads, clubhouses, universities, libraries, churches,medical centers, parking lots, playgrounds,cemeteries)
- commercial places (chemists, pubs, restaurants, depots,hostels, hotels, industrial parks,nightclubs, music venues)
- assorted buildings (houses, monasteries, creches, mills,army barracks, castles, retirement, homes, towers, halls, rooms, vicarages,courtyards)
- abstract "places"' (e.g. {\it the free world})

***Organizations***

- companies (press agencies, studios, banks, stockmarkets, manufacturers, cooperatives)
- subdivisions of companies (newsrooms)
- brands
- political movements (political parties, terrorist, organisations,
- government bodies (ministries, councils, courts, political unionsof countries (e.g. the {\it U.N.}))
- publications (magazines, newspapers, journals)
- musical companies (bands, choirs, opera companies, orchestras
- public organisations (schools, universities, charities

- other collections of people (sports clubs, sportsteams, associations, theaters companies,religious orders, youth organisations

***Persons***

*first, middle and last names of people, animals and fictional characters, aliases*

**Examples of data**

- Locations: Thành phố Hồ Chí Minh, Núi Bà Đen, Sông Bạch Đằng.
- Organization: Công ty Formosa, Nhà máy thủy điện Hòa Bình.
- Persons:proper name in "ông Lân", "bà Hà".
- An entity can contain another entity, e.g. "Uỷ ban nhân dân Thành phố Hà Nội" is an organization, in which contains a location of "thành phố Hà Nội".

Training data consist of two datasets. In the first dataset, data contain the information of word segmentation. The information of POS tags and word chunks tagscan be also added by utilizing available tools. The second dataset is raw data, which contain only NE tags.

*2) Data format:*

*a) Dataset 1:*

Data have been preprocessed with word segmentation and POS tagging.The data consist of five columns, in which two columns are separated by a single space. Each word has been put on a separate line and there is an empty line after each sentence.

- The first column is the word
- The second column is its POS tag
- The third column is its chunking tag
- The fourth column is its NE label
- The fifth column is its nested NE label

NE labels are annotated using the IOB notation as in the CoNLL Shared Tasks. There are 7 labels: B-PER and I-PER are used for persons, B-ORG and I-ORG are used for organizations, B-LOC and I-LOC are used for locations, and O is used for other elements.

An example for a Vietnamese sentence:

```
Anh           N    B-NP     O          O
Thanh         Np   I-NP     I-PER      O
là            V    B-VP     O          O
cán_bộ        N    B-NP     O          O
Uỷ ban        N    B-NP     B-ORG      O
nhân_dân      N    I-NP     I-ORG      O
Thành_phố     N    I-NP     I-ORG      B-LOC
Hà_Nội        Np   I-NP     I-ORG      I-LOC
.             .    O          O        O
```

where {N, Np, V, E, .}are POS tags and{B-NP, I-NP, B-VP, O} are chunking tags.

**Notes:**

- Because POS tags and chunking tags are determined automatically by public tools, they may contain mistakes.
- For NEs, two main tags, i.e.B-XXX and I-XXX, are used. B-XXX is used for the first word of an NE in type XXX, and I-XXX is used for the other words of that NE. The O label is used for words which do not belong to any NE.

*b) Dataset 2:*

Data contain only NE information.

An example: "*Anh Thanh là cán bộ Uỷ ban nhân dân Thành phố Hà Nội*."

```
Anh <ENAMEX TYPE="PERSON"> Thanh </ENAMEX> là cán bộ <ENAMEX TYPE="ORGANIZATION"> Uỷ ban nhân dân<ENAMEX TYPE="LOCATION">Thành phố Hà Nội </ENAMEX></ENAMEX> .
```

*3) Annotation procedure:* In the framework of this shared task, we choose to make use of the POS tagged dataset published by the VLSP project[5].

Two annotators have worked on the NE labeling with double check.

Table I shows the quantity of NE annotated in the training dataset and the test set.

TABLE I. Statistic of NE in the corpus

| Tag | Training Data | | Test Data | |
|---|---|---|---|---|
| | First level | Nested level | First level | Nested level |
| **PER** | 6230 | 480 | 1294 | 7 |
| **LOC** | 1210 | 1 | 1377 | 100 |
| **ORG** | 7478 | 7 | 274 | 0 |
| **Total** | 14918 | 488 | 2945 | 107 |

*C. Evaluation*

The performance of NER systems is evaluated by the F1 score:

$$F1 = 2 * P * R/(P + R)$$

where P (Precision), and R (Recall) are determined as follows:

$$P = \text{NE-true/NE-sys}$$

$$R = \text{NE-true/NE-ref}$$

where:

- NE-ref: The number of NEs in gold data
- NE-sys: The number of NEs extracted by the system
- NE-true: The number of NEs which is correctly recognized by the system

The results of systems will be evaluated at both levels of NE labels.

[5] http://vlsp.hpda.vn:8080/demo/?&lang=en

## III. SUBMISSIONS AND RESULTS

We have five teams participating to this NER campaign, one of them submits two systems. Each team provided us with their full report ([1], [2], [3] and [4]), excepting one just sent us their short description.

### *A. Methods and Features*

Table II gives an overview of the methods and features applied by the submitted systems for detecting the NEs at first level.

TABLE II. Methods and Features

| Team | Methods | Features |
|---|---|---|
| ner1 | Token regular expression + Bidirectional Inference | Basic features (word, pos tag, chunk tag, 2 previous NE tags)<br>Word shapes<br>Basic joint features<br>Regular expression types |
| ner2 | CRF | word, wordCombination, firstSyllable, lastSyllable, ngrams, initUpcaseWord, allCapWord, letterAndDigitWord, isSpecialCharacter, firstSentenceWord, lastSentenceWord and pos |
| ner3-1 | Bidirectional Long short term memory (LSTM) –CRF | Head word, pos, chunk tag |
| ner3-2 | Stack LSTM | |
| ner4 | CRF/MEM+BS | Current word, pos, word form, context words, is syllable, is in dictionary, regular expression for dates, numbers |
| ner5 | CRF | previous word, current word, next word, pos tag, previous pos tag, next pos tag, chunking tag, previous chunking tag, next chunking tag |

For the nested level, only two teams tried to tackle the problem. Team ner5 uses CRF model with the following features: previous word, current word, next word, pos tag, previous pos tag, next pos tag, chunking tag, ner tag, previous tag.

### *B. Results*

As we mentioned above, among six submitted systems only two systems extracted NE at the nested level. However, as the number of entities at this second level is relatively small, it is the system performance at the first level that decides its final performance. In this section we show the results of each system.

**Result of ner1 system:**

| Tag | P | R | F1 |
|---|---|---|---|
| **PER** | 91.52 | 94.20 | 92.84 |
| **LOC** | 86.50 | 0.9354 | 89.88 |
| **ORG** | 78.95 | 43.80 | 56.34 |
| **Total** | 88.36 | 89.20 | 88.78 |

**Result of ner2 system:**

| Tag | P | R | F1 |
|---|---|---|---|
| **PER** | 92.52 | 74.57 | 82.58 |
| **LOC** | 85.79 | 75.38 | 80.25 |
| **ORG** | 61.69 | 34.67 | 44.39 |
| **Total** | 87.16 | 71.24 | 78.40 |

**Result of ner3 system:**

- Result of ner3-1 system:

| Tag | P | R | F1 |
|---|---|---|---|
| **PER** | 94.06 | 81.99 | 87.61 |
| **LOC** | 86.52 | 84.39 | 85.44 |
| **ORG** | 54.85 | 47.45 | 50.88 |
| **Total** | 86.89 | 79.90 | 83.25 |

- Result of ner3-2 system:

| Tag | P | R | F1 |
|---|---|---|---|
| **PER** | 90.06 | 88.95 | 89.50 |
| **LOC** | 84.82 | 84.82 | 84.82 |
| **ORG** | 55.39 | 41.24 | 47.28 |
| **Total** | 85.06 | 82.58 | 83.80 |

**Result of ner4 system:**

| Tag | P | R | F1 |
|---|---|---|---|
| **PER** | 91.74 | 89.19 | 90.45 |
| **LOC** | 86.3 | 81.35 | 83.75 |
| **ORG** | 61.86 | 43.8 | 51.28 |
| **Total** | 87.06 | 81.30 | 84.08 |

**Result of ner5 system:**

| Tag | P | R | F1 |
|---|---|---|---|
| **PER** | 88.19 | 89.41 | 88.8 |
| **LOC** | 83.01 | 92.23 | 87.38 |
| **ORG** | 96.64 | 52.55 | 68.09 |
| **Total** | 85.96 | 87.30 | 86.62 |

## IV. CONCLUSIONS

In this paper, we have described the VLSP 2016 evaluation campaign on Named Entity Recognition in Vietnamese texts. This shared task is organized for the first time for VLSP and attracted 10 registered teams at the first step. At the final step, five teams have submitted their results, with an F1-score varies from 78,4% to 88,78%.

This challenge has allowed the construction of a first Vietnamese dataset for benchmarking named entity recognizers, as well as an overview on performance of

different machine learning approaches for NER of Vietnamese.

In the next campaigns, we expect to build new datasets containing a richer set of named entity categories.

## *Acknowledgment*

We would like to thank the sponsors for helping financially to the construction of the datasets for the named entity recognition task. And special thanks to all the teams who have participated and contributed to the success of this evaluation campaign.

## *References*

[1] Phuong Le Hong, "Vietnamese Named Entity Recognition using Token Regular Expressions and Bidirectional Inference", in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016

[2] Thanh Huong Le, Thi Thu Trang Nguyen, Trong Huy Do and Xuan Tung Nguyen, "Named Entity Recognition in Vietnamese Text", in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016

[3] Truong Son Nguyen, Le Minh Nguyen and Xuan Chien Tran, "Vietnamese Named Entity Recognition @VLSP 2016 Evaluation Campaign", in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016

[4] Thi Cam Van Nguyen, Thai Son Pham, Thi Hong Vuong, Ngoc Vu Nguyen and Mai Vu Tran, "DSKTLAB-NER: Nested Named Entity Recognition in Vietnamese Text", in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016

# Vietnamese Named Entity Recognition using Token Regular Expressions and Bidirectional Inference

Phuong Le-Hong
College of Science
Vietnam National University, Hanoi, Vietnam
Email: *phuonglh@vnu.edu.vn*

***Abstract*—This paper describes an efficient approach to improve the accuracy of a named entity recognition system for Vietnamese. The approach combines regular expressions over tokens and a bidirectional inference method in a sequence labelling model. The proposed method achieves an overall $F_1$ score of $89.66\%$ on a test set of an evaluation campaign, organized in late 2016 by the Vietnamese Language and Speech Processing (VLSP) community.**

## I. Introduction

Named entity recognition (NER) is a fundamental task in natural language processing and information extraction. It involves identifying noun phrases and classifying each of them into a predefined class. In 1995, the 6th Message Understanding Conference (MUC) started evaluating NER systems for English, and in subsequent shared tasks of CoNLL 2002 and CoNLL 2003 conferences, language independent NER systems were evaluated. In these evaluation tasks, four named entity types were considered, including names of persons, organizations, locations, and names of miscellaneous entities that do not belong to these three types.

More recently, the Vietnamese Language and Speech Processing (VLSP) community has organized an evaluation campaign to systematically compare NER systems for the Vietnamese language. Similar to the CoNLL 2003 share task, four named entity types are evaluated: persons (PER), organizations (ORG), locations (LOC), and miscellaneous entities (MISC). The data are collected from electronic newspapers published on the web.

This paper presents the approach and experimental results of our participating system on this evaluation campaign. In summary, the overall $F_1$ score of our system is $89.66\%$ on a development set extracted from the training dataset provided by the organizing committee of the evaluation campaign. Three important properties of our approach include (1) use of token regular expressions to encode regularities of organization and location names, (2) an algorithm to annotate every token in an input sentence with their token regular expression types, and (3) a bidirectional decoding approach to boost the accuracy of the system.

The remainder of this paper is structured as follows. Section II gives a brief introduction of multinomial logistic regression, the main machine learning model which is used in our system. Section III describes in detail the features used in our model, including common features used in NER and those derived from our newly proposed token regular expressions. This section also presents an algorithm we develop to annotate every token of an input sentence with its regular expression type. Section IV introduces a bidirectional decoding scheme and a method to combine forward and backward models to get a better model. Section V gives experimental results and discussions. Finally, Section VI concludes the paper.

## II. Multinomial Logistic Regression

Multinomial logistic regression (a.k.a maximum entropy model) is a general purpose discriminative learning method for classification and prediction which has been successfully applied to many problems of natural language processing, such as part-of-speech tagging, syntactic parsing and named entity recognition. In contrast to generative classifiers, discriminative classifiers model the posterior $P(y|\,\mathbf{x})$ directly. One of the main advantages of discriminative models is that we can integrate many heterogeneous features for prediction, which are not necessarily independent. Each feature corresponds to a constraint on the model. In this model, the conditional probability of a label $y$ given an observation $\mathbf{x}$ is defined as

$$P(y|\,\mathbf{x}) = \frac{\exp(\theta \cdot \phi(\mathbf{x}, y))}{\sum_{y \in \mathcal{Y}} \exp(\theta \cdot \phi(\mathbf{x}, y))},$$

where $\phi(\mathbf{x}, y) \in \mathbb{R}^D$ is a real-valued feature vector, $\mathcal{Y}$ is the set of labels and $\theta \in \mathbb{R}^D$ is the parameter vector to be estimated from training data. This form of distribution corresponds to the maximum entropy probability distribution satisfying the constraint that the empirical expectation of each feature is equal to its true expectation in the model:

$$\widehat{\mathbb{E}}(\phi_j(h, t)) = \mathbb{E}(\phi_j(h, t)), \qquad \forall j = 1, 2, \ldots, D.$$

The parameter $\theta \in \mathbb{R}^D$ can be estimated using iterative scaling algorithms or some more efficient gradient-based optimization algorithms like conjugate gradient or quasi-Newton methods [1]. In this paper, we use the L-BFGS optimization algorithm [2] and $L_2$-regularization technique to estimate the parameters of the model. This classification model is applied to build a classifier for the dependency parser where each observation $\mathbf{x}$ is a parsing configuration and each label $y$ is a transition type.

## III. Feature Representations

In discriminative statistical classification models in general and the maximum entropy model in particular, features play an important role because they provide the discriminative ability to efficiently disambiguate classes.

In order to facilitate the extraction of various feature types, each lexical token is associated with a surface *word* and an *annotation map* containing different information of the text in the form of key and value pairs. The current annotation map includes values for part-of-speech, chunk, token regular expression type and named entity label.

In the context of named entity recognition, the information about surface word, part-of-speech and chunk tag are given; and in a training phrase, named entity tags are also provided. In the next subsection, we describe the regular expression types which are associated with each token to add some helpful semantic information for named entity disambiguation.

### A. Regular Expressions over Tokens

We use regular expressions at both character level and token level to infer useful features for disambiguating named entities. While character-level regular expressions are used to detect word shape information, which was shown very important in NER, token-level regular expressions are very helpful to detect word sequence information in many long named entities [3].

Common word shape features that our system uses include:

- is lower word, e.g., "*tỉnh*"
- is capitalized word, e.g., "*Tổng_cục*"
- contains all capitalized letters (allcaps), e.g., "*UBND*"
- is mixed case letters, e.g., "*iPhone*"
- is capitalized letter with period, e.g., "*H.*", "*Th.*", "*U.S.*"
- ends in digit, e.g., "*A9*", "*B52*"
- contains hyphen, e.g., "*H-P*"
- is number, e.g., "*100*"
- is date, e.g., "*20-10-1980*", "*10/10*"
- is code, e.g, "*21B*"
- is name, where consecutive syllables are capitalized, e.g., "*Hà_Nội*", "*Buôn_Mê_Thuột*"

Using the word shape features presented above, we then introduce regular expressions over a sequence of words to capture its regularity. Suppose that *fPress(w)* is a boolean function which returns *true* if *w* is in a set of predefined words related to press and newspaper domain, for example *{"báo", "tờ", "tạp_chí", "đài", "thông_tấn_xã"}*, and returns *false* otherwise. And suppose that *fName(w)* is a boolean function which returns *true* if *w* is a name or an allcaps, and returns *false* otherwise. Then, we can define the following token regular expressions to capture the name of a news agency:

*[fPress, fName]*

This sequence pattern matches many different, probably unseen news agency names, such as *"báo Tuổi_Trẻ, thông_tấn_xã Việt_Nam"*, or *"tờ Batam"*.

In a similar way, suppose that we have a function *fProvince* which matches common names of administrative structure at various levels such as *"{tỉnh, thành_phố, quận, huyện, xã,...}"*, we can build a sequence pattern

*[fAllcaps, fProvince, fName]*

which matches many corresponding organization names such as *"UBND thành_phố Đà_Nẵng"*, *"HĐND huyện Mù_Căng_Chải"*, etc.

Note that an elementary token pattern can be reused in multiple sequence patterns. For example, the following sequence pattern

*[fProvince, fName]*

can match provincial names, which are usually of type location, such as *"tỉnh Quảng_Ninh"*, *"thành_phố Hồ_Chí_Minh"*.

By examining the training data, we have manually built a dozen of regular expresions to match common organization names, and six regular expressions to match common location names. These regular expressions over tokens are shown to provide helpful features for classifying candidate named entities, as shown in the experiments.

### B. Regular Expression Type Annotation

Once regular expressions over tokens have been defined, we add a regular expression type for each word of an input sentence by annotating its corresponding annotation map key. Together with word identity, word shape, part-of-speech and chunk tag information, regular expression types provide helpful information for better classifying named entities, as shown in the latter experiments.

We use a greedy algorithm to annotate regular expression type for every word if it has. Basically, the algorithm works as follows. Given a sequence of $T$ tokens (or words) $[w_1, w_2, \ldots, w_T]$ and a map of regular expressions over tokens, each key name defines a pattern sequence: *(patternName, patternRegExp)*, we first search for all positions of the sentence which begins a pattern match, and select the longest match, say, pattern *patternName* which ranges from token $w_i$ to token $w_j$, for $1 \leq i < j \leq T$. Then, all the tokens $w_i, w_{i+1}, \ldots, w_j$ are annotated with the same regular expression type *patternName*. And finally, the algorithm recursively annotates types for tokens in the remaining two halves of the sequence $[w_1, w_2, \ldots, w_{i-1}]$ and $[w_{j+1}, w_{j+2}, \ldots, w_T]$.

Note that this is a greedy method in that we always choose the longest pattern in each run. This is a plausible approach since if there are multiple matches, longer patterns tend to be more correct than shorter ones. For example, there are two matches on the token sequence "*UBND tỉnh Đồng_Nai*", one is an organization name over the entire sequence, and another is a location name over the last two tokens; the longer one is the correct match.

### C. Feature Set

In this subsection, we describe the full feature set that is used in our system to classify a token at a position of a sentence.

- Basic features: current word $w_0$, current part-of-speech $p_0$, current chunk tag $c_0$, previous named entity tags $t_{-1}$ and $t_{-2}$ (or a special padding token "BOS" – begin of sentence);
- Word shape features, as described in the previous subsection;
- Basic joint features: previous word $w_{-1}$ (or "BOS"), joint of current and previous word $w_0 + w_{-1}$, next word $w_{+1}$ (or "EOS" – end of sentence), joint of current and next word $w_0 + w_{+1}$, previous part-of-speech $p_{-1}$, joint of current and previous part-of-speech $p_0 + p_{-1}$, next part-of-speech $p_{+1}$, joint of current and next part-of-speech $p_0 + p_{+1}$, joint of previous and next part-of-speech $p_{-1} + p_{+1}$, joint of current word and previous named entity tag $w_0 + t_{-1}$;
- Regular expression types: current regular expression (regexp) type $r_0$ (or "NA" – not available), previous regexp type $r_{-1}$ (or "NA"/"BOS"), joint feature $r_0 + r_{-1}$, next regexp type $r_{+1}$ (or "NA"/"EOS"), joint feature $r_0 + r_{+1}$, joint features between current word and regexp types $w_0 + r_0$, $w_0 + r_{-1}$, $w_0 + r_{+1}$, and lastly, joint features between current part-of-speech and regexp types $p_0 + r_0$, $p_0 + r_{-1}$, and $p_0 + r_{+1}$.

## IV. Bidirectional Decoding

The standard decoding algorithm for sequence labelling is the Viterbi algorithm, which is a dynamic programming algorithm for finding the most likely sequence of tags given a sequence of observations. In this work, we also use the Viterbi algorithm to find the best tag sequence for a given word sequence. However, we found a significant improvement of tagging accuracy when combining two decoding directions, both forward decoding and backward decoding. In this section, we describe our bidirectional decoding approach.

Given a sequence of $T$ words $[w_1, w_2, \ldots, w_T]$, for each word $w_j$, a pre-trained multinomial logistic regression model computes a conditional probability distribution over possible tags $y_j \in \mathcal{Y}$:

$$P(y_j|c_j) = \frac{\exp(\theta \cdot \phi(c_j, y_j))}{\sum_{y_j \in \mathcal{Y}} \exp(\theta \cdot \phi(c_j, y_j))}, \quad \forall j = 1, 2, \ldots, T,$$

where $\phi(c_j, y_j)$ is the feature function which extract features from context $c_j$ containing known information up to position $j$. As described in the previous section, $c_j$ encodes useful features for predicting $y_j$, including those extracted from a local word window $w_{j-2}, \ldots, w_{j+2}$, two previous tags $y_{j-1}, y_{j-2}$, and joint features between them.

The probability of a tag sequence given a word sequence is approximated by using the Markov property. In a log scale, we have

$$\log P(y_1, \ldots, y_T|w_1, \ldots, w_T) \approx \sum_{j=1}^{T} \log P(y_j|c_j).$$

The Viterbi algorithm is then used to find the best tag sequence $\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_T$ corresponding to the max-probability path on a lattice of size $K \times T$ where $K = |\mathcal{Y}|$ is the size of the tag set.

Note that in the second-order Markov model as above, each context $c_j$ uses the two tags $y_{j-2}$ and $y_{j-1}$ which have been infered in the previous steps. That said, this is a left-to-right inference scheme. In the experiments, we use a greedy update at each position $j$ where the tag $y_j$ is chosen as the best tag of each local probabilty distribution computed by the maximum entropy model.

A reversed inference scheme does the same decoding procedure but in a right-to-left fashion, where two tags $y_{j+2}, y_{j+1}$ are infered before decoding $y_j$. In essence, when performing backward decoding, we can use the same Viterbi decoding procedure as in the forward counterpart, but now using a backward maximum entropy model to compute the probability of a tag given its following tags. It turns out that both the training and decoding procedure for this model can be reused simply by reversing the word and tag sequences at both training and test stages.

An important finding in our experiments is that the backward model is much better than the forward model in recognizing location names while it is much worse in recognizing person names. We therefore propose a method to combine the strength of the two models to boost the accuracy of the final model. The combination method will be presented in detail in the experiments.

## V. Experiments

### A. Datasets

We evaluate our system on the training dataset provided by the VLSP NER campaign.[1] This dataset contains $16,858$ tagged sentences, totaling $386,520$ words. The dataset contains four different types of named entities: person (PER), location (LOC), organization (ORG), and miscellaneous (MISC). Since the real test set has not been released, we divide this training set into two parts, one for training and another for development. The training dataset has $306,512$ tokens (79.3% of the corpus), and the development dataset has $80,007$ tokens (20.7% of the corpus).

### B. Parameter Settings

The multinomial logistic regression models used in our system are trained by the L-BFGS optimization algorithm using the $L_2$-regularization method with regularization parameter fixed at $10^{-6}$.[2] The convergence tolerance of objective function is also fixed at $10^{-6}$. The maximum number of iterations of the optimization algorithm is fixed at 300. That is, the training terminates either when the function value converges or when the number of iterations is over 300. We use the feature hashing technique as a fast and space-efficient method of vectorizing features. The number of features for our models are fixed at $262,144$ (that is, $2^{18}$).

These parameters values are chosen according to a series of experiments, for example, using a smaller number of features

[1] http://vlsp.org.vn/evaluation_campaign_NER

[2] Using a larger regularization parameter underfits the model.

(say, $2^{17}$) reduces slightly the performance of the models, while using a larger number does not result in an improvement of accuracy but increase the training time.

### C. Main Results

We train our proposed models on the training set and test them on the development set as described in the previous subsection. The performance of our system is evaluated on the development set by running the automatic evaluation script of the CoNLL 2003 shared task[3]. The main results are shown in Table I.

Table I: Performance of our system

| Type | Precision | Recall | $F_1$ |
|---|---|---|---|
| All | 89.56% | 89.75% | 89.66 |
| LOC | 84.97% | 94.13% | 89.32 |
| MISC | 93.02% | 81.63% | 86.96 |
| ORG | 79.75% | 52.72% | 63.48 |
| PER | 94.82% | 92.75% | 93.77 |

Our system achieves an $F_1$ score of 89.66% overall. Organization names are the most difficult entity type for the system, whose $F_1$ is the lowest of 63.48%. Person names are the easiest type for the system whose both precision and recall ratios are high and the $F_1$ score of this type is 93.77%.

### D. Effect of Bidirectional Inference

In this subsection we report and discuss the results using unidirectional inference, either forward and backward. The performance of the forward model is shown in Table II and that of the backward model is shown in Table III.

Table II: Performance of the forward model

| Type | Precision | Recall | $F_1$ |
|---|---|---|---|
| All | 88.08% | 87.10% | 87.59 |
| LOC | 81.61% | 86.54% | 84.00 |
| MISC | 97.67% | 85.71% | 91.30 |
| ORG | 79.75% | 52.72% | 63.48 |
| PER | 94.38% | 93.45% | 93.91 |

Table III: Performance of the backward model

| Type | Precision | Recall | $F_1$ |
|---|---|---|---|
| All | 88.03% | 87.94% | 87.98 |
| LOC | 85.60% | 91.80% | 88.59 |
| MISC | 100.00% | 83.67% | 91.11 |
| ORG | 66.45% | 43.10% | 52.28 |
| PER | 92.15% | 92.54% | 92.34 |

We see that the backward model is better than the forward model by 4.6 point of $F_1$ score in recognizing location names. This is surprising since the only difference between the two models is a reverse of input sentences. One possible explanation of this effect is that when recognizing location names of a token sequence $w_1, w_2, \ldots, w_n$, if we already know about the type of $w_n$ it is easier to predict its previous token $w_{n-1}$ and so on. We conjecture that this is due to the natural structure of Vietnamese location names.

However, the backward model underperforms the forward model in recognizing the organization names by a large margin. Its $F_1$ score of this type is only 52.28%, while that of the forward model is 63.48%. This is understandable because our token regular expressions are designed to capture regularities in many organization names, as described in the subsection III-A, but these expressions do not work anymore if an input token sequence is reversed.

Either of the two unidirectional models achieves an overall $F_1$ score of 88.00% but when they are combined, our system achieves an overal score of 89.66%, as presented in the previous subsection. The combined model has both the strong ability of recognizing location names of the backward model and is good at recognizing organization names of the forward model.

### E. Effect of Token Regular Expressions

In this subsection, we report the effectiveness of token regular expressions to our model. We observe that using token regular expressions significantly improves the performance of the system.

If the token regular expressions for ORG type are not used, its $F_1$ score of the forward model is 62.94%. Adding token regular experessions for this type help boost this score to 65.01%. Similarly, when token regular expressions for LOC are not used, its score of the forward model is 82.19%. Adding six token regular expressions for this type improves its score to 83.07%. However, we observe that when all the regular expressions for this two named entity types are used together, they interact with each other and make their scores slightly different, as shown in the Table II.

### F. Software

The named entity recognition system developed in this work has been integrated into the Vitk toolkit, which includes some fundamental tools for processing Vietnamese texts. The toolkit is developed in Java and Scala programming languages, which is open source and freely downloadable for research purpose.[4] An interesting property of this toolkit is that it is an Apache Spark application, which is a fast and general engine for large scale data processing. As a result, Vitk is a very fast and scalable toolkit for processing big text data.

## VI. Conclusion

We have introduced our approach and its experimental result in named entity recognition for Vietnamese text. We have shown the effectiveness of using token regular expressions, of bidirectional decoding method in a conditional Markov model for sequence labelling, and of combining the backward and forward models. Our system achieves the overall $F_1$ score of 89.66% on a test corpus.

[3]http://www.cnts.ua.ac.be/conll2003/ner/

[4]https://github.com/phuonglh/vn.vitk

## ACKNOWLEDGEMENTS

This research is partly financially supported by Alt Inc.[5], and in particular we thank Dr. Nguyen Tuan Duc, the head of Alt Hanoi office. We thank the developers of the Apache Spark software.


## REFERENCES


[1] G. Andrew and J. Gao, "Scalable training of $l_1$-regularized log-linear models," in *Proceedings of ICML*, Oregon State University, Corvallis, USA, 2007, pp. 33–40.

[2] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[3] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of CoNLL-2003*, W. Daelemans and M. Osborne, Eds. Edmonton, Canada, 2003, pp. 142–147.


[5]http://alt.ai/corporate

# Named Entity Recognition in Vietnamese Text

Thanh Huong LE, Thi Thu Trang NGUYEN
Hanoi University of Science and Technology
1 Dai Co Viet, Hanoi, Vietnam
{huonglt,trangntt}@soict.hust.edu.vn

Trong Huy DO, Xuan Tung NGUYEN
Hanoi University of Science and Technology
1 Dai Co Viet, Hanoi, Vietnam
{tronghuy2807, tungxbk}@gmail.com

***Abstract*—This paper introduces our approach to Named Entity Recognition for Vietnamese text, using Conditional Random Fields. To find the optimal feature set for the training process, a hill climbing algorithm is used. The F-scores of the NER system when using the given training set and test set are 94.67% and 78.38%, respectively. Future work is to use a semi supervised method to boost the result when the training data set is small.**

***Keywords— Named Entity Recognition, Conditional Random Field, hill climbing***

## I. INTRODUCTION

Named Entity Recognition (NER) is the task of locating and classifying atomic elements in text into predefined categories (e.g., people, organizations, locations, expressions of time, quantities, etc.). NER is useful in many applications such as search engine, question-answering system, etc.

Most NER's systems concentrate on machine learning approaches, whose performance relies on features being used. Therefore, selecting a subset of good features is an optimization problem of multiple objectives, such as number of features and accuracy. Using a small feature subset requires less computational time than using a larger one. However, the system's accuracy is lower than using a larger, suitable set of features. Another problem is, using more features increases the system complexity, but does not always increase the system's accuracy. A bad feature may even degrade the performance of the system. In this paper, we introduce our method to select the optimal feature set that can produce high accuracy for the NER system. Section III will represents our method to solve this task.

Another problem is, there are several ambiguous cases in recognizing named entity labels from Vietnamese text. Because of that, we propose a post process to deal with these cases. This process will be introduced in Section IV.

There are a number of models for the NER problem such as: Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM) and Conditional Random Field (CRF). Since Conditional Random Field is suitable model for NER task, it will be used as the machine learning algorithm in our system. CRF will be introduced in the next section.

## II. CONDITIONAL RANDOM FIELD

### *1) Definition of Conditional Random Fields.*

The theory of Conditional Random Fields is proposed by Lafferty et al. (2001). Conditional Random Fields are undirected graphical models trained to maximize a conditional probability.

A linear-chain CRF with parameters $\Delta = \{\lambda, ...\}$ defines a conditional probability for a state (or label) sequence $y = y_1 \dots y_T$ given an input sequence $x = x_1 \dots x_T$ (in which T is the length of sequence) to be

$$P_{\lambda}(y \mid x) = \frac{1}{z_x} \exp\left( \sum_{t=1}^{T} \sum_{k} \lambda_k f_k(y_{t-1}, y_t, x, t) \right) \quad \text{(Eq. 1)}$$

where:

- x, y: the observing data string and the state respectively;
- $Z_x$ is the normalization constant that makes the probability of all state sequences sum to one;
- $f_k(y_{t-1}, y_t, x, t)$ is a feature function which is often binary-valued, but can be real-valued;
- $\lambda_k$ is a learned weight associated with feature $f_k$. Large positive values for $\lambda_k$ indicate a preference for such an event, while large negative values make the event unlikely.

*Training CRF*

Training process is actually seeking set parameters $\lambda_k$ of the model. The method for evaluating the model is to maximize the likelihood measure between distribution models and empirical distribution.

$$l(\lambda) = \sum_{x,y} \hat{p}(x,y) \sum_{i=1}^{n+1} \lambda_i f_i - \sum_{x} \hat{p}(x) \log Z \quad \text{(Eq. 2)}$$

It proved to be the log-likelihood for the model CRF is concave function and continuous in the whole space of the parameters. So we can find the maximum log-likelihood function by means of derivatives (Eq. 2) with respect to $\lambda_k$, as in Equation 3.

$$\frac{\partial l(\lambda)}{\partial \lambda_k} = E_{\hat{p}(x,y)}[f_k] - E_{p(x,y)}[f_k] \quad \text{(Eq. 3)}$$

The establishment of the Equation 3 equal to 0 is equivalent to making binding model: the expected value of the

feature function $f_k$ to the distribution model with the expected value of the feature function $f_k$ for empirical distribution.

Currently, there are many methods for solving log-likelihood function maximum, such as: iterative methods (IIS and GIS), Conjugate Gradient method, L-BFGS method. As assessed by Malouf (2002), the method is considered more effective is L-BFGS.

Inference in CRF

Inference in CRF is seeking sequence state y* that maximizes the probability distribution p(y|x)between sequence y and observing data x (Eq. 4).

$$y^* = \text{argmax}\, y^* \, p(y \mid x) \quad \text{(Eq. 4)}$$

where y* sequence was determined by the Viterbi algorithm (Kristjansson et al., 1999).

To our knowledge, CRF is one of the best methods of machine learning for NER. CRF is limited by the high complexity of learning algorithms. To recognize the entity for the EC, we have used Stanford Named Entity Recognizer (SNER, 2012) written by Finkel and his team. SNER is a CRF toolkit written by Java, which is suitable for processing Vietnamese text.

## III. FEATURES USING IN NER TASK

### *III.1. Feature set*

The NER task considered in this research is to recognize Person, Organization, Location name entities from Vietnamese text. An important task in using machine learning algorithm is to use an appropriate feature set. To solve this task, basing on the studies of Vietnamese features for CRF (Tran et al, 2007; Le Thanh et al., 2015) and features used in SNER (2012), the following feature types are proposed to use in our NER system: *word, wordCombination, firstSyllable, lastSyllable, ngrams, initUpcaseWord, allCapWord, letterAndDigitWord, isSpecialCharacter, firstSentenceWord, lastSentenceWord* and *pos.* Vietnamese language is monosyllable. Its vocabulary has many compound words such as "*bến tàu*$_{\text{pier}}$", "*bến xe*$_{\text{car park}}$" in which the first syllable (e.g., "*bến*$_{\text{parking place}}$" ) is a large concept, whereas the second syllable is a specific kind of that concept (e.g., "*xe*$_{\text{car}}$"). Another type of compound words (e.g., "*tuyến đường* $_{\text{route}}$", "*mặt đường*$_{\text{road}}$") has the last syllable is a concept (e.g., "*đường*$_{\text{road}}$"), whereas the first syllable is an attribute of that concept (e.g., "*tuyến*$_{\text{line}}$", "*mặt*$_{\text{face}}$"). Therefore, we propose to use firstSyllable and lastSyllable in our NER system to detect words belonging to the same group.

In general, each feature type is considered in the window size of five (two words before the current word, the current word, and two words after the current word). However, since some feature types at certain positions have no meaning in the NER task, we restrict the feature's positions as shown in Table I.

TABLE I. FEATURES FOR NAMED ENTITY RECOGNITION

| Index | Feature group | Feature type | Position |
|---|---|---|---|
| 1 | Context | word | 0, ±1, ±2 |
| 2 | | wordCombination | (-1 0), (0 1), (-1 1), (-2 -1 0), (-1 0 +1), (0 +1 +2) |
| 3 | | firstSyllable (e.g., *bến tàu*$_{\text{pier}}$ , *bến* xe$_{\text{car park}}$) | 0, -1, -2 |
| 4 | | lastSyllable (e.g., tuyến *đường* $_{\text{route}}$, mặt *đường*$_{\text{road}}$) | 0, -1, -2 |
| 5 | NGram | NGram characters of a word by 5 lengths | 0, ±1, ±2 |
| 6 | | NGram but does not include the characters start and end of the first words | 0, ±1, ±2 |
| 7 | | NGram from previous word and current word | -1, 0 |
| 8 | Orthography | initUpcaseWord | 0, ±1, ±2 |
| 9 | | allCapWord | 0, ±1, ±2 |
| 10 | | letterAndDigitWord | 0, ±1, ±2 |
| 11 | | isSpecialCharacter (e.g., . ? ! ( ) " ) | ±1, ±2 |
| 12 | | firstSentenceWord | 0, -1, -2 |
| 13 | | lastSentenceWord | 0, +1, +2 |
| 14 | Part-of-Speech | pos | 0, ±1, ±2 |

In Table 1, position 0 means the current word; positions ±1, ±2 mean the word at the position ±1, ±2 compared to the current word. Each feature is a combination of its feature type and its position. For example, there are five features with the feature type "word", including prev2Word (the word before the previous word), prevWord (the previous word), curWord (the current word), nextWord (the next word), next2Word (the word after the next word), corresponding to positions -2, -1, 0, 1, and 2, respectively.

### *III.2. Optimazing feature set*

To optimize feature set, the NER annotated corpus is divided into two parts: 70% of the corpus for training, 20% for optimizing feature set, and 10% for evaluating the system. A hill climbing algorithm is used to get an optimal feature set for NER task. The optimizing feature process includes the following steps:

1. Selecting features from each feature groups and combining them into a feature set.
2. Training the NER system with the chosen feature set.
3. Computing Precision, Recall and F-score of the NER system.

Loop:

4. Adding a feature which has not been used in the feature set.

5. Training the NER system with the new feature set.
6. Computing Precision, Recall and F-score of the NER system.
7. If the new F-score increases compared with the previous F-score, keep the new feature in the feature set;

   Otherwise, remove it from the feature set.

Table II shows the optimal features after the selection process is done.

TABLE II. OPTIMAL FEATURES

| | Group | Features |
|---|---|---|
| 1 | Context | Words at positions 0, ±1, ±2 |
| 2 | | wordCombination at positions (-1 0), (0 1), (-1 1) |
| 3 | | wordCombination (-1 0 1) with low case |
| 12 | NGram | 5 characters of a word |
| 13 | | NGram but does not include first and last characters of a word |
| 14 | | NGram for previous word and current word |

The POS is not a good feature with the given training data since the F-score of the system decreases 6% when using it.

## IV. EXPERIMENTAL RESULTS

The data used in our system was provided by VLSP workshop (2016), including a training set and a test set. The training set includes of 267 documents which already annotated with named entity labels. The test set consists of 45 documents without named entity labels. Each file has 50 sentences in average. The training documents were annotated manually with tags B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, and O (other). The test set does not have these tags and was used in the campaign for evaluating the NER system. Our task is to automatically annotated the test set with the above NER tags.

Since the given test set does not include NER tags, 70% of the training set was used for training and the remaining part was used for evaluating the system. Traditional measures in machine learning tasks, Precision, Recall, F-score were used for this purpose.

Experimental results with the training set are shown in Table III below. The results were calculated by an evaluation tool provided by the VLSP workshop.

TABLE III. EXPERIMENTAL RESULTS USING THE TRAINING SET

---------- Ner tag lv1 (column 4) ----------

| Tag | TruePos | FalsePos | FalseNeg | P | R | F1 |
|---|---|---|---|---|---|---|
| LOC | 1448 | 75 | 153 | 0.9508 | 0.9044 | 0.9270 |
| ORG | 235 | 44 | 66 | 0.8423 | 0.7807 | 0.8103 |
| PER | 2492 | 44 | 89 | 0.9826 | 0.9655 | 0.9740 |
| MIS | 49 | 0 | 5 | 1.0000 | 0.9074 | 0.9515 |
| Total | 4224 | 163 | 313 | 0.9628 | 0.9310 | 0.9467 |

---------- Ner tag lv1+lv2 (column: 4, 5) ----------

| Tag | TruePos | FalsePos | FalseNeg | P | R | F1 |
|---|---|---|---|---|---|---|
| LOC | 1439 | 84 | 162 | 0.9448 | 0.8988 | 0.9213 |
| ORG | 147 | 132 | 154 | 0.5269 | 0.4884 | 0.5069 |
| PER | 2491 | 45 | 90 | 0.9823 | 0.9651 | 0.9736 |
| MISC | 49 | 0 | 5 | 1.0000 | 0.9074 | 0.9515 |
| Total | 4126 | 261 | 411 | 0.9405 | 0.9094 | 0.9247 |

The accuracies of NER tags lv1+lv2 are lower than that of NER tag lv1 since we did not filled values on column lv2.

Our system output for the given test set was evaluated by VLSP's organization. The results are shown in Table IV below.

TABLE IV. EXPERIMENTAL RESULTS USING THE TEST SET

---------- Ner tag lv1 (column 4) ----------

| Tag | TruePos | FalsePos | FalseNeg | P | R | F1 |
|---|---|---|---|---|---|---|
| PER | 965 | 78 | 329 | 0.9252 | 0.7457 | 0.8258 |
| LOC | 1038 | 172 | 339 | 0.8579 | 0.7538 | 0.8025 |
| ORG | 95 | 59 | 179 | 0.6169 | 0.3467 | 0.4439 |
| MISC | 32 | 2 | 17 | 0.9412 | 0.6531 | 0.7711 |
| Total | 2130 | 311 | 864 | 0.8726 | 0.7114 | 0.7838 |

---------- Ner tag lv1+lv2 (column: 4, 5) ----------

| Tag | TruePos | FalsePos | FalseNeg | P | R | F1 |
|---|---|---|---|---|---|---|
| PER | 965 | 78 | 329 | 0.9252 | 0.7457 | 0.8258 |
| LOC | 1038 | 172 | 339 | 0.8579 | 0.7538 | 0.8025 |
| ORG | 37 | 117 | 237 | 0.2403 | 0.1350 | 0.1729 |
| MISC | 32 | 2 | 17 | 0.9412 | 0.6531 | 0.7711 |
| Total | 2072 | 369 | 922 | 0.8488 | 0.6921 | 0.7625 |

The F1 scores of the system when using the given test set is much lower than that when using the given training set. This is because the training set is quite small. The F1 scores of the ORG tags are lowest since the training data for this tags is smallest.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have reported the method of implementing our NER system for Vietnamese text. Our system uses the Stanford Named Entity Recognizer for NER task and a hill climbing algorithm to optimize the feature set. The F-scores of the NER system when using the given training set and test set are 94.67% and 78.38%, respectively. Future

work is to use a semi supervised method to boost the result when the training data set is small.

## *References*


[1] Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In Proceedings ofSixth Workshop on Computational Language Learning(CoNLL-2002)

[2] Lafferty, J., McCallum, A. and Pereira, F. (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In Proceedings ofI CML, pages 282-290.

[3] Huong Le Thanh, Luan Tran Van, Hoai Nguyen Xuan, Hien Nguyen Thi. 2015. Optimizing Genetic Algorithm in Feature Selection for Named Entity Recognition. SoICT 2015.

[4] Trausti Kristjansson, Aron Culotta, Paul Viola, Andrew McCallum. Interactive Information Extraction with Constrained Conditional Random Fields. In Proceedings of the National Conference on Artificial Intelligence, London, 1999.

[5] SNER (2012). Stanford Named Entity Recognizer.http://nlp.stanford.edu/software/CRF-NER.shtml. (lastvisited Aug. 2016)

[6] VLSP workshop (2016) http://rivf2016.tlu.edu.vn/tabid/394/catid/940/workshops.aspx

[7] Tran Thi Q, Thao Pham T.X, Hung Ngo Q, Dien Dinh, Nigel Coller: Named entity recognition in Vietnamese documents. Progress in Informatics, No.4, pp.5–13, (2007)

# Vietnamese Named Entity Recognition at VLSP 2016 Evaluation campaign

Nguyen Truong Son
School of Information Science
JAIST
1-8 Asahidai, Nomi, Ishikawa, Japan
and
University of Science
VNU-HCMC, Ho Chi Minh city, Vietnam
Email: nguyen.son@jaist.ac.jp

Tran Xuan Chien
School of Information Science
JAIST
1-8 Asahidai, Nomi, Ishikawa, Japan
Email: nguyen.son@jaist.ac.jp

Nguyen Le Minh
School of Information Science
JAIST
1-8 Asahidai, Nomi, Ishikawa, Japan
Email: nguyen.son@jaist.ac.jp

***Abstract*—This paper adapted some state-of-the-art approaches for Vietnamese Named Entity Recognition task at the VLSP 2016 evaluation campaign. We exploited two types of models that are invented for sequence labeling task including Conditional Random Fields (CRFs) and Neural architectures. In the neural architectures, we investigated several models based on Long Short Term Memory (LSTM) including the combination between of LSTM with CRFs and transition-based approach inspired by shift-reduce parser algorithms based on a stack LSTM. The experimental results are very promising. It produces an F1 score of 90.34% when using CRFs with three types of features. The results of neural approaches are also very competitive, they outperform some previous methods without using any features. Besides, we have built an end-to-end system that can recognize named entities in Vietnamese documents.**

***Index Terms*—named entity recognition ; recurrent neural networks; conditional random fields; transition-based parser**

## I. Introduction

Named entity recognition is a fundamental task in Natural Language Processing. This task aims to recognize named entities such as Person, Organization, Location, Time, and Currency in general text or some entities in other kinds of text such as names of protein and diseases in bio-medical documents. This task is a sub-task of Information Extraction that helps to identify entities before finding the relationship between them.

There are two main approaches for the Named Entity Recognition including rule-based and machine-learning based approaches. The rule-based approach is a simple method that tries to recognize entities by discovering a set of rules. However, this method is language dependent and it is costly and time-consuming to build the rule set that can cover all types of entities. Machine learning approaches are more preferred in recent years by treating the task of Named Entity Recognition as the sequence labeling task or classification task.

Many popular machine-learning algorithms have been proposed to solve the NER task such as Conditional Random Fields [1], Hidden Markov Model [2], Maximum Entropy [3] [4].

In Vietnamese, [5] has proposed a classifier voting approach. The main idea of this approach is that it first uses several machine learning approaches to recognize entities, and then it will use the voting technique to make the final decision if the result of used approaches is conflict.

Recently, deep learning approaches have been adapted to many Natural Language Processing tasks including Named Entity Recognition. Despite of not using engineering features as well as additional resources, the results of deep learning approaches are very competitive with state-of-the-art results.

The remainder of this paper is organized as follows. Section II and Section III described some approaches that we have adapted for the Vietnamese Named Entity Recognition task including Conditional Random Fields and Neural architectures. Section IV presents the experimental results and some comments. A demonstration system for NER is described in Section V. Conclusion and Future works are showed in Section VI.

## II. Models

In this section we described machine-learning approaches that we have adapted to the Vietnamese Named Entity Recognition task including Conditional Random Fields and Neural architectures.

### A. Conditional Random Fields

It has been shown that Conditional Random Fields (CRFs) are powerful approaches for sequence labeling tasks [1]. CRFs got the lower error rate than other probabilistic models such as Hidden Markov Model (HMM) or Maximum Entropy Markov Model (MEMM)

In this work, we use CRF++ [6] to train the models for NER task. CRF++ is a tool implemented Conditional Random Fields to solve the sequence labeling tasks such as NER, chunking and Information Extraction.

In order to use CRF++ for training models, we need to describe the template file that helps CRF++ generate a set of feature functions. In Section III, we will describe the detail of the template file used in our experiments. These features are based on headwords, POS, chunk tags or combinations between them.

### B. Neural architectures

#### 1) LSTM and BI-LSTM:

Long Short Term Memory [7] is a variant of recurrent neural networks (RNNs), which is designed for solving the sequence-to-sequence task. LSTM solved the gradient vanishing / exploding that occurs in RNNs [8]. Because an RNN / LSTM network receives a sequence of words and produces a sequence of labels, they are very suitable to solve the task of Named entity Recognition.

LSTM are useful for capture memory in a long range. However, with a single LSTM, they can only predict the tag at the current time based on the previous. However, for the sequence labeling task, we can predict the current time not only depend on the previous information but also future information. Therefore, Bidirectional LSTM networks are designed to overcome this problem. A Bidirectional LSTM (BI-LSTM) contains two single LSTM networks including forward LSTM and backward LSTM.

Our approach use BI-LSTM implementation proposed by [9] and [10]. Figure 1 illustrates the architecture of BI-LSTM networks.

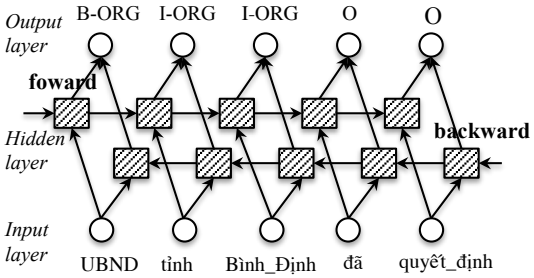



Figure 1. BI-LSTM architecture

#### 2) BI-LSTM-CRF:

It has been shown that CRFs are powerful approaches for sequence labeling tasks [1]. Therefore, [10] proposed the combination between LSTM and CRFs by adding a CRF layer that takes the output of LSTM networks as the input features to train CRFs. This CRF layer can be added into LSTM or BI-LSTM models. Figure 2 illustrates a CRF layer is attached to a BI-LSTM model. This combination has been shown that it can take advantages of both these two methods [10].

#### 3) Transition-based approaches based on Stack LSTM:

Another technique we used for this task is the transition-based NER system [9]. It uses the stack LSTM to train and predict the NER tags. In this method, the NER labeling task is modeled as a task to predict a sequence of actions to reach the goal state from the initial state, each state consists of a stack and a buffer. Figure 3 shows an example of the actions and corresponding states when parsing a sentence.

*Input word embedding for neural networks:* :

There are two ways for initializing the values of word embedding vectors for neural networks. If we have pre-trained words

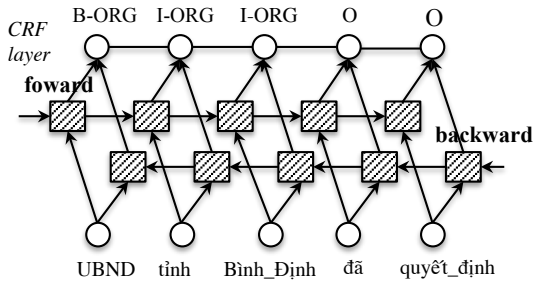



Figure 2. BI-LSTM-CRF architecture

| | |
|---|---|
| | [] [UBND, tỉnh, Bình_Định, đã, quyết_định] |
| SHIFT | [UBND] [tỉnh, Bình_Định, đã, quyết_định] |
| SHIFT | [UBND, tỉnh] [Bình_Định, đã, quyết_định] |
| SHIFT | [UBND, tỉnh, Bình_Định] [đã, quyết_định] |
| REDUCE(ORG) | [] [đã, quyết_định] |
| OUT | [] [quyết_định] |
| OUT | [] [] |
| *Actions* | *Stack and Buffer* |

Figure 3. Transition-based approach for predicting NER tags.

vectors, we just initialize them by use the vectors in the pre-trained source. However, we don't have any pre-trained vectors for Vietnamese words. Instead, we initialized the embedding matrix randomly and let the system learn this matrix during the training phase.

## III. Experiments

### A. Data sets

The data set given by the VLSP 2016 includes 267 text documents, which have already segmented. They contain 16858 sentences with more than 15000 named entities in which LOC and PER are two popular entity types (90%). The summarization of the given data set is described in Table I.

Table I
THE STATISTIC OF PROVIDED DATA SET

| | COUNT | PERCENT |
|---|---|---|
| LOC | 6245 | 41 % |
| MISC | 282 | 2 % |
| ORG | 1213 | 8 % |
| PER | 7480 | 49 % |
| #total of NE | 15220 | 100 % |
| #total of sentence | 16858 | |

The data format is followed the CONLL format. Each sentence in the data set is represented by a list of lines. Each

line represented a word (column 0) and word information including Part-Of-Speech tag (column 1), chunk tag (column 2).

The last column is the named entity tag (column 3), which is followed the IOB notation. In the task of Vietnamese Named Entity Recognition, we need to recognize 4 types of entity including $Person$, $Organization$, $Location$ and $Miscellaneous$. Therefore, the tag set in the training set including 9 types of tags B-LOC, I-LOC, B-ORG, I-ORG, B-PER, I-PER, B-MISC, I-MISC, O. B- tags and I-tags are used to mark words that are the beginning and the inside of entities. If a word doesn't belong to any entities, it will be marked by the O tag. Sentences are separated by blank lines. Table II showed an example sentence in the provided data set.

Table II
THE DATA FORMAT

| | | | |
|---|---|---|---|
| Không | R | O | O |
| chỉ | R | O | O |
| ở | E | B-PP | O |
| Dầu_Giây | NNP | B-NP | B-LOC |
| mà | C | O | O |
| ở | E | B-PP | O |
| rất | R | O | O |
| nhiều | A | B-AP | O |
| nơi | N | B-NP | O |
| nạn | N | B-NP | O |
| mãi_lộ | N | B-NP | O |
| vẫn | R | O | O |
| đang | R | O | O |
| hoành_hành | V | B-VP | O |
| . | CH | O | O |

### B. Evaluation method

In order to evaluate CRFs models, we divide the data set into 2 folds including the training set and the test set. However, for other methods, which based on LSTM networks, we need to use a validation set to monitor the training process. Therefore, we divide the data set into 3 folds including training, validation and test set. Recall that, the test sets for the evaluation of all models are the same to ensure that the results are comparable. Table III and Table IV show the details of the training set, validation set, and the test set which are used to train and evaluate our models.

We evaluate all trained models using Precision, Recall, and F1 score. We use third-party evaluation tool, called $conlleval$, which is provided by CoNLL 2000 Shared task [11], for the evaluation stage to ensure objective assessments.

Table III
THE STATISTIC OF DATA SETS USED TO TRAIN AND EVALUATE PERFORMANCE OF CONDITIONAL RANDOM FIELDS

| | TRAIN | TEST | TOTAL |
|---|---|---|---|
| LOC | 5596 | 649 | 6245 |
| MISC | 230 | 52 | 282 |
| ORG | 1074 | 139 | 1213 |
| PER | 6598 | 882 | 7480 |
| **#total of NE** | **13498** | **1722** | **15220** |
| **#sentence** | **15011** | **1847** | **16858** |

Table IV
THE STATISTIC OF DATA SETS USED TO TRAIN AND EVALUATE PERFORMANCE OF NEURAL MODELS

| | TRAIN | VALIDATE | TEST | TOTAL |
|---|---|---|---|---|
| LOC | 4991 | 605 | 649 | 6245 |
| MISC | 195 | 35 | 52 | 282 |
| ORG | 946 | 128 | 139 | 1213 |
| PER | 5973 | 625 | 882 | 7480 |
| **#total of NE** | **12105** | **1393** | **1722** | **15220** |
| **#sentence** | **13370** | **1641** | **1847** | **16858** |

### C. Experiment setting

*1) Experiments setting for CRF++:* Table V shows the template file for generating feature functions in our approaches. It will generate features functions that capture the information in a window with window size of 5 words. We use headword, Part-of-Speech tag and chunk tags as the features for training CRFs models. The rule for feature description in the must follow the format of the provided data set. The rule for writing the template and the explanation are described in [6].

We have experimented CRFs with different kinds of features. The feature sets of each model are described in Table VI.

*2) Experiment setting for BI-LSTM-CRF:* :
We used the same setting with [9] for the size of embedding vectors and the size of hidden layers. We train our BI-LSTM-CRF model using the back propagation method algorithms and update parameters on every training example using stochastic gradient descent (SGD) with learning rate of 0.002. We also set dropout rate of the network to 0.5. After 20 epochs, we choose the best model if they produce the best result on the validation set.

*3) Experiment setting for Stack LSTM:* When running Stack LSTM system, we set both the relation embedding dimension and action embedding dimension to 20. Word embedding dimension was 100 and dropout was 0.3. The provided data set was divided into three sets at 80-10-10 ratios: training, validation, and test. We ran the experiment on the training set and tuned the network parameters on the validation set. The best model on validation set was used to run on the test set.

In order to run the experiment, we had to convert the data set from CONLL format into a new format defined by the system.

Table V
THE TEMPLATE FILE FOR GENERATING FEATURES. FROM U00 TO U06 ARE FEATURES RELATED TO HEAD WORDS; FROM U11-U22 ARE FEATURES RELATED TO PART-OF-SPEECH OF WORDS. FROM U30-U38 ARE FEATURES RELATED TO CHUNK TAGS.

```
# Unigram
U00:x[-2,0]
U01:x[-1,0]
U02:x[0,0]
U03:x[1,0]
U04:x[2,0]
U05:x[-1,0]/x[0,0]
U06:x[0,0]/x[1,0]
U10:x[-2,1]
U11:x[-1,1]
U12:x[0,1]
U13:x[1,1]
U14:x[2,1]
U15:x[-2,1]/x[-1,1]
U16:x[-1,1]/x[0,1]
U17:x[0,1]/x[1,1]
U18:x[1,1]/x[2,1]
U20:x[-2,1]/x[-1,1]/x[0,1]
U21:x[-1,1]/x[0,1]/x[1,1]
U22:x[0,1]/x[1,1]/x[2,1]
U30:x[-2,2]
U31:x[-1,2]
U32:x[0,2]
U33:x[1,2]
U34:x[2,2]
U35:x[-2,2]/x[-1,2]
U36:x[-1,2]/x[0,2]
U37:x[0,2]/x[1,2]
U38:x[1,2]/x[2,2]
# Bigram
B
```

Table VI
FEATURE SETS CONFIGURATION OF DIFFERENT CRF MODELS

| | Word (U00-U06) | POS tag (U10-U22) | Chunk tag (U30-U38) |
|---|---|---|---|
| CRF1 | x | | |
| CRF2 | x | x | |
| CRF3 | x | | x |

The author provided all necessary scripts for this conversion[1].

### D. Experimental results and comments

Table VII and Table VIII show the experimental results of all our models. The results on the official test set is showed in Table IX.

Among three CRFs-based models, CRF3 produced the best result. The reason why CRF3 outperformed CRF1 and CRF2 is that CRF3 used richer feature sets including word, POS and chunk tags.

BI-LSTM-CRF and Stack-LSTM outperformed CRF1 and CRF2 although they don't use any kind of feature set. However, BI-LSTM-CRF and Stack-LSTM is worse than CRF3. This result is understandable because CRF3 uses chunk tags, a very important information for recognizing named entities, as the features. These results showed that neural architectures are promising approaches for Vietnamese Named Entity Recognition task. The performance of neural networks can be improved if we used pre-trained vectors or combine with other kinds of features.

Among four types of named entities, named entities in category *MISC* has the highest results. It is understandable,

[1]https://github.com/clab/stack-lstm-ner

Table VII
THE DETAILS OF EXPERIMENTAL RESULTS

| | Type | Precision | Recall | F1 |
|---|---|---|---|---|
| CRF 1 (WORD) | LOC | 91.02 | 67.18 | 77.3 |
| | MISC | 100 | 86.54 | 92.78 |
| | ORG | 75.51 | 26.62 | 39.36 |
| | PER | 94.09 | 63.15 | 75.58 |
| | Overall | 92.27 | 62.43 | 74.47 |
| | | | | |
| CRF 2 (WORD + POS) | LOC | 81.17 | 85.67 | 83.36 |
| | MISC | 98.00 | 94.23 | 96.08 |
| | ORG | 78.57 | 31.65 | 45.13 |
| | PER | 90.51 | 95.12 | 92.76 |
| | Overall | 86.61 | 86.41 | 86.51 |
| | | | | |
| CRF 3 (WORD + POS + CHUNK) | LOC | 85.96 | 91.53 | 88.66 |
| | MISC | 100 | 100 | 100 |
| | ORG | 85.88 | 52.52 | 65.18 |
| | PER | 92.94 | 95.46 | 94.18 |
| | Overall | 90.02 | 90.65 | 90.34 |
| | | | | |
| Bi-LSTM-CRF | LOC | 84.78 | 87.52 | 86.13 |
| | MISC | 90.91 | 96.15 | 93.46 |
| | ORG | 64.29 | 45.32 | 53.16 |
| | PER | 95.27 | 93.65 | 94.45 |
| | Overall | 89.17 | 87.51 | 88.34 |
| | | | | |
| Stack LSTM | LOC | 84.99 | 88.14 | 86.54 |
| | MISC | 94.44 | 98.08 | 96.23 |
| | ORG | 63.89 | 49.64 | 55.87 |
| | PER | 92.70 | 93.65 | 93.18 |
| | Overall | 87.95 | 88.15 | 88.05 |

Table VIII
THE OVERALL RESULTS

| | Precision | Recall | F1 |
|---|---|---|---|
| CRF 1 | 92.27 | 62.43 | 74.47 |
| CRF 2 | 86.61 | 86.41 | 86.51 |
| CRF 2 | 90.02 | 90.65 | 90.34 |
| Bi-LSMT-CRF | 89.17 | 87.51 | 88.34 |
| Stack LSTM | 87.95 | 88.15 | 88.05 |

because entities of *MISC* type are easy to predict such as name of languages: *Tiếng Việt* (Vietnamese), *Tiếng Anh* (*English*); name of ethic or nationality: *Người Pháp* (*French*) and so on. Named entities in *LOCATION* type are the most difficult type to predict. Organizations are usually accompanied by their locations, so the system failed in finding them. Therefore, the precision score of recognition entities in type *ORGANIZATION* is high but the recall score is low. These phenomenons happen in most approaches.

Table IX
RESULTS ON THE OFFICIAL TEST SET

| | Type | Precision | Recall | F1 |
|---|---|---|---|---|
| CRF 3 (WORD + POS + CHUNK) | LOC | 83.45 | 92.17 | 87.59 |
| | MISC | 100 | 87.76 | 93.48 |
| | ORG | 93.75 | 49.27 | 64.59 |
| | PER | 88.58 | 91.11 | 89.83 |
| | Overall | 86.42 | 87.72 | 87.06 |
| BI-LSTM-CRF | LOC | 86.52 | 84.39 | 85.44 |
| | MISC | 78.05 | 65.31 | 71.11 |
| | ORG | 54.85 | 47.45 | 50.88 |
| | PER | 94.06 | 81.99 | 87.61 |
| | Overall | 86.76 | 79.66 | 83.06 |
| Stack LSTM | LOC | 84.82 | 84.82 | 84.82 |
| | MISC | 85 | 69.39 | 76.4 |
| | ORG | 55.39 | 41.24 | 47.28 |
| | PER | 90.06 | 88.95 | 89.5 |
| | Overall | 85.06 | 82.36 | 83.69 |

## IV. DEMONSTRATION SYSTEMS


Figure 4. The demonstration system of Vietnamese Name Entity Recognition. The system can identify NEs based on the trained model and Users can modify the incorrect NEs to extend the corpus

After models are built, we also build a demonstration system that can recognize named entities in Vietnamese documents. The system receives a tokenized document or raw text then output the text in which named entities are marked. We used vnTagger [12] to tokenize input documents.

After named entities are recognized, the user can remove incorrect or add new named entities. The system can track the feedback from users to extend the corpus. A part of user interfaces is illustrated in Figure 3. Besides, system can automatically identify named entities from some Vietnamse news websites such as VnExpress by crawling the content of news use using RSS feeds provided by those websites.

## V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have adapted some approaches to Vietnamese Named Entity Recognition Task at the VLSP 2016 evaluation campaign. We used two popular approaches that are strong models for the Named Entity Recognition task including Conditional Random Fields and, neural architectures. In the neural architectures, we used two type models including BI-LIM-CRF, which is the combination of LSTM, and CRFs and the model based transition based models with Stack LSTM.

The result is very promising so we hope these results are good baselines for future works. In future works, we would like to integrate engineering features into neural architectures as well as using pre-trained vectors for the input layer of LSTM networks. The positive effects of using pre-trained vectors have been showed in some previous works [9], [10], [13], [14].

## ACKNOWLEDGMENT

We would like to thank VLSP 2016 organizer have proposed the evaluation campaign that help Vietnamese researcher can solve the Vietnamese NLP tasks together and their methods can be evaluated on the same data sets. We would like to thanks Lample et al. [9], have published the source code of neural architectures that we have used in this research.

## REFERENCES

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the eighteenth international conference on machine learning, ICML*, vol. 1, 2001, pp. 282–289.
[2] G. Zhou and J. Su, "Named entity recognition using an hmm-based chunk tagger," in *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 473–480.
[3] A. Borthwick, "A maximum entropy approach to named entity recognition," Ph.D. dissertation, Citeseer, 1999.
[4] A. Ratnaparkhi *et al.*, "A maximum entropy model for part-of-speech tagging," in *Proceedings of the conference on empirical methods in natural language processing*, vol. 1. Philadelphia, USA, 1996, pp. 133–142.
[5] P. T. X. Thao, T. Q. Tri, D. Dien, and N. Collier, "Named entity recognition in vietnamese using classifier voting," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 6, no. 4, p. 3, 2007.
[6] T. Kudo, "Crf++: Yet another crf toolkit," *Software available at http://crfpp. sourceforge. net*, 2005.
[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[8] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.
[9] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.
[10] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[11] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the conll-2000 shared task: Chunking," in *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*. Association for Computational Linguistics, 2000, pp. 127–132.

[12] P. Le-Hong, A. Roussanaly, T. M. H. Nguyen, and M. Rossignol, "An empirical study of maximum entropy approach for part-of-speech tagging of vietnamese texts," in *Traitement Automatique des Langues Naturelles-TALN 2010*, 2010, p. 12.

[13] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Part-of-speech tagging with bidirectional long short-term memory recurrent neural network," *arXiv preprint arXiv:1510.06168*, 2015.

[14] P. Wang, Y. Qian, F. Soong, L. He, and H. Zhao, "A unified tagging solution: Bidirectional lstm recurrent neural network with word embedding," *arXiv preprint arXiv:1511.00215*, 2015.

# DSKTLAB-NER: Nested Named Entity Recognition in Vietnamese Text

Cam-Van Thi Nguyen[1], Thai-Son Pham[1], Thi-Hong Vuong[1], Ngoc-Vu Nguyen[1,2], Mai-Vu Tran[1]
{vanntc_58, sonpt_59, hongvt_57, vutm}@vnu.edu.vn, nnvu@monre.gov.vn
[1]Data Science and Knowledge Technology Laboratory,
University of Engineering and Technology, Vietnam National University, Hanoi
[2]Department of Information Technology, Ministry of Natural Resources and Environment

***Abstract*—This paper presents a Vietnamese Nested Named-Entity Recognition system using combined methods, namely maximum entropy decoding by beam search. The task descriptions, datasets and evaluation methods are provided by The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)[1]. We considered the given task of detecting nested entities. Our system has achieved an F1 score of 83.84 on the final test sets. This paper also presents a detailed discussion about the characteristics of the Vietnamese Language and provide an analysis of the factors which influence performance in this task.**

***Keywords—Named entity recognition, Vietnamese nested NER,***

## I. INTRODUCTION

Considerable amount of research has been done on the named-entity recognition (NER) task, because of its important role in applications such as question answering or document and news searching. However, the performance of existing NER systems for Vietnamese is still below systems for English by a large margin. Vietnamese NER appears to present a significant challenge in a number of important respects. Firstly, words in Vietnamese are not always separated by spaces, so word segmentation is necessary and segmentation errors will affect the level of NER performance. Secondly, some proper names of foreign persons and locations are loanwords or represented by phonetic symbols, so we can expect wide variations in some Vietnamese terms. Thirdly, there is considerably fewer available extant resources such as lexicons, parsers, word nets, etc. for Vietnamese which have been used in previous studies [11]. Finally, a considerable fraction of named entities in Vietnamese contains other named entities inside. Nested named entities recognition is a difficult subtask due to some reasons: it is ignored in most of common corpora, and many traditional models cannot detect nested entities.

In this paper, we focus on recognizing three kinds of name entity (NE): person, location and organization from Vietnamese texts. In section 2, we discuss about some related works and their approaches. Section 3 describes the specific task and datasets provided. In section 4, we present our approach with the features selection and the hybrid machine learning model. The evaluation and analysis are provided in section 5. Finally, our conclusions and future plans are discussed in section 6.

## II. RELATED WORK

Named Entity Recognition was first looked into more concretely back in 1990 [12], when the main approaches were still based on heuristics and handcrafted rules. Shortly afterwards, it was already recognized as an essential subtasks of Information Extraction [7]. Different NER systems were evaluated as a part of the Sixth Message Understanding Conference in 1995. After 1995, NER has been attracted the attention of the community research. There were many systems which solved this problem such as Automatic Content Extraction, publishment of authors at Conference on Natural Language Learning in 2002 and 2003 [13], BioCreative (Critical Assessment of Information Extraction Systems in Biology). The NER systems learned data patterns which is labeled. Before 2000, researchers used the statistical machine learning methods as Nymble - a search tool based on Hidden Markov Models (HMMs). Then, there were machine learning models using to solve NER problem such as Maximum Entropy Models (MEMs) [1], Conditional Random Field (CRF) [3]. In addition to research methods, the authors also used hybrid model among ensemble model methods to result effectively as study of Florian et.al [8].

Besides NER on English texts, which is generally the language concentrating most efforts, a small number of approaches for other languages were also carried out, such as (IREX) [5] for Japanese or as well the systems on German, Dutch or Spanish presented during the CoNLL 2002 and 2003 Shared Tasks [4,6]. For the Vietnamese language, using supervised learning, Tu et al. (2005) [9] built an NER system with CRFs and reported 87.90% F1 score as their highest performance. Using SVMs, Tran et al. (2007) [11] achieved 87.75% F1 score for the task.

In our work, we implement a machine learning approach with Maximum Entropy Model decoding by Beam Search (MEM + BS) classifiers which are trained for the outer and nested spans of the NEs present in the VLSP 2016 dataset.

## III. NER TASK DESCRIPTION

### A. Task Description

The main aim of the campaign VLSP 2016 is to evaluate of ability of recognizing NEs in three entities, i.e. PER - name of persons, ORG - organizations and LOC - locations. This task is not only the detection of NEs, but as well the extension of the task specifically to one language - Vietnamese. Vietnamese has characteristics different from other languages,

[1]VLSP 2016: http://vlsp.org.vn

such as English. Unlike English, Vietnamese is isolating language which means almost every word consists of a single morpheme. The illustration of this is, the meaning of the word "science" in Vietnamese is "công nghệ". Meanwhile, the word "công" and "nghệ" mean "peacock" and "turmeric" in English, respectively. This causes difficulties when determining the boundaries of a word. The research so far given more named entity recognition methods successfully applied in the English corpus could not make the corresponding result for data use Vietnamese.

The task datasets provided consist mainly of articles collected from newspapers published on Vietnamese's website. Three types of NEs are compatible with the descriptions in the CoNLL Shared Task 2003[2]. Data have been preprocessing with word segmentation and POS tagging. The training dataset covers over 19,000 sentences corresponding to over 380,000 tokens. It contain five columns, in which two columns are separated by a single space. Each word has been put on a separate line and there is an empty line after each sentence. The first item on each line is a word, the second a part-of-speech (POS) tag, the third a syntactic chunk tag, the fourth the named entity tag and the fifth the nested named entity tag. The chunk tags and the named entity tags have the format I-TYPE which means that the word is inside a phrase of type TYPE. Only if two phrases of the same type immediately follow each other, the first word of the second phrase will have tag B-TYPE to show that it starts a new phrase. A word with tag O is not part of a phrase. There are 7 labels: B-PER and I-PER are used for persons, B-ORG and I-ORG are used for organizations, B-LOC and I-LOC are used for locations, and O is used for other elements.

### B. *VLSP 2016 Corpus Analysis*

According to the statistic from training data, we have faced with imbalanced data problem, namely the number of organization entity (ORG) is smaller than the remaining labels. Table 1 shows the amount of entities with IOB tags and corresponding the average length of entities which is tokenized in training dataset. Looking at the total number row, it is noticeable that the number ORG entities is smallest and only approximate one third compared with the remaining tags. It suggest that the data which is imbalanced. Further, the length of the entities also cause difficulties in the process of identification. The longer length entities increasingly difficult to identify which means in some cases, the model does not identify or recognize false. For example, sentence in figure 1 has 9 tokens. In this case. The model identify missing location entity which is "Đồng_Tháp".

<ENAMEX TYPE="ORGANIZATION"> Trung_tâm Nước sinh_hoạt & vệ_sinh môi_trường nông_thôn tỉnh Đồng_Tháp </ENAMEX>

**Fig 1.** An example sentence of the VLSP 2016 data XML format.

Besides, we found some noise in data. For example, some person entities still contain person-prefix such as: "Anh Hai", "Chị Hoa", etc. Meanwhile, the only person entities are labeled with the name does not contain the prefix. We also found ambiguity with several entities. Entities "bệnh viện" (hospital) is not informed in how labeling. In the contest guidelines, hospital is a public places which is labeled location. But in the testing dataset which is provided by organizer, entity "bệnh viện" is still being labeled as organization.

| | PER | ORG | LOC |
|---|---|---|---|
| **Total number** | 11020 | 3276 | 9735 |
| **Average entities length (words)** | 3.1 | 3.8 | 2.8 |

**Table 1.** The number entities and their length corresponding to the IOB tags.

"Anh Thanh là cán bộ Uỷ ban nhân dân Thành phố Hà Nội."

<ENAMEX TYPE="PERSON"> Anh Thanh </ENAMEX> là cán bộ <ENAMEX TYPE="ORGANIZATION"> Uỷ ban nhân dân <ENAMEX TYPE="LOCATION"> thành phố Hà Nội </ENAMEX> </ENAMEX>.

**Fig 2.** A nested entities example.

From the perspective of classifier training, imbalance in training data distribution often causes learning algorithms to perform poorly on the minority class. Moreover, the VLSP 2016 NER datasets also contains annotations of nested named entities. Besides the three main classes person - PER, organization - ORG, location - LOC, frequently, entities are nested within each other, such as "Ủy ban nhân dân thành phố Hà Nội" (People's Committee of Hanoi city" is organizations with nested locations and "Đại học Tôn Đức Thắng" (Ton Duc Thang University) is organization with nested person. This caused misleading in NER process, hence, we found certain challenge of identify organization - ORG entity. Figure 2 presents a nested entities example in VLSP 2016 corpus. Solving this problem, we will propose solutions in Section 4.

## IV. OUR APPROACH

### A. Nested NER Model

Our systems consist of four phases: pre-processing, featurisation, sequence labeling and nested entities recognition. In the pre-processing phase, we identify sentences and word segmentation from dataset. The abbr. expansion help recognize entities such as implicit name of persons, implicit name of organizations or implicit name of locations which is used in the previous sentences. For example, figure 3 shows an example sentence has token "HQ" that it is a short name of "Hàn Quốc". Another sentence in the same article in figure 4. With this context, our model is still recognize "HQ" is the location as "Hàn Quốc".

Đây là một trong những phim_trường lớn nhất <ENAMEX TYPE="LOCATION"> Hàn_Quốc </ENAMEX> ( HQ ) hiện nay"

**Fig 3.** An abbreviation example.

"Nếu một năm <ENAMEX TYPE="LOCATION"> HQ </ENAMEX> sản_xuất trên 100 bộ phim thì hơn 60 bộ phim thực_hiện tại phim_trường <ENAMEX TYPE = "LOCATION"> Yangsuri</ENAMEX>."

**Fig 4.** A sentence contains an abbreviation.

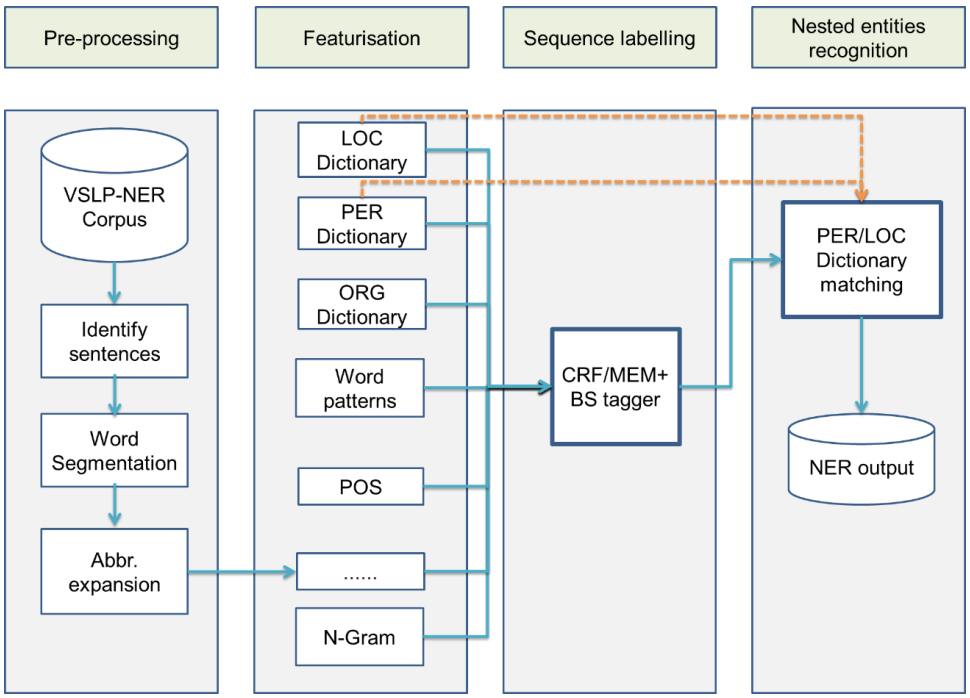



**Fig2.** The stages of our experimental NER systems

The second phase is features which help more accurate identification. More detail about feature selection, see at Table 2. Next, we used machine learning taggers to identify named entities. In this phase, model will label longest matching sequences to dictionary. Last, we show our resolution for nested entities recognition which is using dictionary matching.

B. Feature Selection

We design a rich feature set listed in table 2 by integrating some kinds of feature. We use Freebase person name dictionary (1,397,865 words) and our three supporting dictionaries to extract more useful features.

- Vietnamese person name dictionary has 20,669 words.
- Vietnamese location dictionary has 18,331 words.
- Prefix dictionary included person prefix (like "ngài" (Mr.), "PGS." (Assoc.), etc), location prefix (like "Quận" (district), "Thành phố" (city), etc) and organization prefix (like "trường đại học" (university), "công ty" (company), etc). This dictionary has 790 Vietnamese words.

| | **Features type** | **Notation** |
|---|---|---|
| 1 | Current word | $W_0$ |
| 2 | POS tag of the current word | $POS\ (W_0)$ |
| 3 | Is current word is lowercase, initial capitalization or all capitalization? | *Is_Lower(0,0)* *Is_Initial_Cap (0,0)* *Is_All_Cap (0,0)* |
| 4 | Context words | $W_i\ (i = -2,-1,1,2))$ |
| 5 | Syllable Conjunction | *Syllable_Conj (-2,2))* |
| 6 | Regular Expression tries to capture expressions describing date/time, numbers, marks, etc | *Regex(0,0)* |
| 7 | Vietnamese Syllable Detection | *Is_Valid_Vietnamese_Syllable(0,0)* |
| 8 | Is this word a valid entry in name dictionaries? | *dict:name, dict:first_name dict:vname dict:vfirst_name* |
| 9 | Is the previous word of considering word a valid entry in prefix dictionaries? | *prefix:per prefix:loc prefix:org* |

**Table 2.** Feature set for NER.

C. Classifier Selection

Usually, classifier NER systems use either Conditional Random Fields (CRF) [6], Maximum Entropy classifiers (ME) [1] or other machine learning methods. Apart from differing slightly in their generalization power, the classifiers also have a differences in the duration of training, classification and RAM usage. Besides using each method to solve separate problems, the method of combining or hybrid machine learning models to deliver better results as well as a direction to achieve high efficiency. These include study of Florian et al (2003) have achieved the best result (88.76%) in CONLL-2003 Shared Task [5].

We experimented with combined methods classifier that is Maximum Entropy decoding by Beam Search. In general, MaxEnt applied to the problem of sequence labeling, its original used dynamic programming algorithm Viterbi to decode. MaxEnt achieves good performances on many NLP tasks. To solve the problem of identifying the entity phenotype and related entities, we adopt an approximate search methods to decode which is Beam Search (BS) instead Viterbi. BS is a variation of the breadth-first search using parameter k. It can reduce dimensional search space. Our proposal model set k = 3. Advantage of using BS algorithm is control maximum entropy for each decision labeled but ignore search capability optimized label chain by dynamic programming techniques. This has enhanced the speed calculations.

Along with MEM + BS, we experimented with two other models that are CRF and Collin's perceptron to evaluate the results of my model.

The nested named entity recognition problems is mentioned above, the solution which we applied is using the dictionary about locations. That is, for each organization entities, our model will map to the dictionary and labeled if our model found the appearance of entities' location. This also improves this task's result.

## V. EVALUATION

### A. *Experimental Results and Analysis*

Within the sequence labeling phase we compare three widely used machine learning models: CRF, Collin's Perceptron, ME+BS. All are run as fully supervised models. Class labels for tokens follow the standard BIO system, i.e. each token receives the label O if it is not an NE, B plus the entity name when it starts an entity, and I plus the entity name when it is inside an entity.

We use OpenNLP for MEM+BS/Collin's perceptron learning and MALLET for CRF learning. OpenNLP and MALLET take the standard CoNLL NER shared task and XML format. We converted the data and features into the accepted format and trained the model using the tool's default parameter configuration (beam size = 3, iterator = 100, cutoff = 0). We did no feature selection and all folds use the same parameter setting (10 folds cross validation).

| | Label | Precision | Recall | F1 |
|---|---|---|---|---|
| **CRF** | PER | 86.54 | 87.31 | 86.92 |
| | LOC | 79.25 | 80.14 | 79.69 |
| | ORG | 63.82 | 60.41 | 62.07 |
| **Collin's perceptron** | PER | 81.34 | 80.13 | 80.73 |
| | LOC | 76.23 | 68.13 | 71.95 |
| | ORG | 53.14 | 49.23 | 51.11 |
| **MEM + BS** | PER | 92.41 | 87.24 | **89.75** |
| | LOC | 82.17 | 86.35 | **84.21** |
| | ORG | 66.38 | 65.21 | **65.79** |

**Table 3.** Comparison among models with training dataset

The results demonstrates the strength of ME+BS, achieved 89.75% in F1 for PER entity (2.83% better than CRF), 73.62 % in F1 for LOC (4.52% better than CRF) and 65.79% in F1 for ORG (3.72% better than CRF). We used ME+BS to recognize entities in test set.

| | Label | Precision | Recall | F1 |
|---|---|---|---|---|
| **MEM + BS** | PER | 91.74 | 89.19 | 90.45 |
| | LOC | 86.30 | 81.35 | 83.75 |
| | ORG | 58.28 | 41.24 | 48.29 |
| Overall | | | | 83.84 |

**Table 4.** Test set results.

Look at the result table, PER and LOC named entities has F1-score very well but ORG named entities has bad result. We had anticipated this result based on the initial data analysis. Like mentioned in section 2, this result is the consequence of an imbalance data and our models are still not good enough to be aware of all data in nested type.

**"Cuối_cùng chúng_tôi tìm đến gỡ_cửa Sở Nông_nghiệp & phát_triển nông_thôn."**

**Fig 5.** A missing example

**"Phóng_viên** *<ENAMEX TYPE="ORGANIZATION">* **Batam Pos** *</ENAMEX>* **bắt_tay chúng_tôi và nói: …"**

**Fig 6.** An error example

Our model is still existence with certain error. First, the model is missing identify. Sentence in figure 5 is not recognized with "Sở Nông_nghiệp & phát_triển nông_thôn" as organization. Second, the model is wrong identify. For example, sentence in figure 6 is mistakenly identified. "Batam Pos" is a person's name, not the name of organization. In some case, if organization contains person's name, the system will likely to identify the person's name rather than organization.

## VI. FUTURE WORK AND CONCLUSION

In this paper, we presented the participation of DSKTLAB - NER, which is a hybrid approach to NER, at the VLSP 2016 Named Entity Recognition Shared Task for Vietnamese. We evaluated the system on the test set provided by VLSP 2016, reaching an F-score 83.84% on the final test set, which is considerably good for the rich feature set that the system employs.

In the future, we would like to look deeper into the use of world knowledge for NER and explore solutions for Nested

NER problem. Another possible way of improving our system would be to reform the organization identification. We intend to develop a combination of methods to identify individual organization entity. They are pretty good and can give great results in the future.

## REFERENCES


[1] Bard, J.B and Rhee,S.Y. Ontologies in biology: design, applications and future challenges. Nature Review Genetics, 5(3), 213-222, 2014.

[2] Borthwick,A. A Maximum Entropy Approach to Named Entity Recognition. Phd, New York University, 1999.

[3] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1):3–26, 2007.

[4] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Walter Daelemans and Miles Osborne, editors, Proceedings of CoNLL-2003, pages 142–147. Edmonton, Canada, 2003.

[5] Florian, R., Ittycherial, A., Jing, H. and Zhang, T. Named Entity Recognition through Classifier Combination. Proceedings of CoNLL-2003. Edmonton, Canada, 2003.

[6] J. R. Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of ACL 2005, pages 363–370.

[7] Grishman,R. and Sundheim,B. Message Understanding Conference-6: A Brief History. In COLING (Vol.96), 1996.

[8] Lafferty,J., McCallum,A. and Pereira, F.C. Conditional random field: Probabilistic models for segmenting and labeling sequence data, 2001.

[9] Nguyen Cam Tu, Tran Thi Oanh, Phan Xuan Hieu, and Ha Quang Thuy. Named entity recognition in Vietnamese free-text and web documents using conditional random fields. In Conference on Some Selection Problems of Information Technology and Telecommunication, 2005.

[10] Satoshi Sekine and Hitoshi Isahara. Irex: Irand ie evaluation project in japanese. In Proceedings of International Conference on Language Resources & Evaluation (LREC 2000), 2000.

[11] Q. Tri Tran, T.X. Thao Pham, Q. Hung Ngo, Dien Dinh, and Nigel Collier. Named entity recognition in Vietnamese documents. Progress in Informatics, 5:14–17, 2002.

[12] Rau,L.F . Extracting company names from text. In Artificial Intelligence Applications. Proceedings. Seventh IEEE Conference on (Vol.1), 1991.

[13] Tjong Kim Sang,E.F, & De Meulder,F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. InProceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003.

# VLSP 2016 Shared Task: Sentiment Analysis

Le Anh Cuong
Faculty of Information Technology
Ton Duc Thang University, Ho Chi Minh city, Vietnam

Nguyen Thi Minh Huyen and Nguyen Viet Hung
Faculty of Mathematics, Mechanics and Informatics
VNU University of Science, Hanoi, Vietnam

***Abstract*—The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016) organizes the shared task of Sentiment Analysis at the first time for Vietnamese language processing. This campaign aims to bring together researchers interested in this topic as well as provides a benchmark dataset for this task. We received eight submissions with various methods and promising results.**

## I. Introduction

With the development of technology and the Internet, different types of social media such as social networks and forums have allowed people to not only share information but also to express their opinions and attitudes on products, services and other social issues. The Internet becomes a very valuable and important source of information. People nowadays use it as a reference to make their decisions on buying a product or using a service. Moreover, this kind of information also lets the manufacturers and service providers receive feedback about the limitations of their products and therefore should improve them to meet the customer needs better. Furthermore, it can also help authorities know the attitudes and opinions of their residents on social events so that they can make appropriate adjustments.

Since early 2000s, opinion mining and sentiment analysis have become a new and hot research topic in Natural language Processing and Data Mining. The paper [1] is a very good survey for the development of this topic. The major tasks in this topic include:

- *Subjective classification*: aims to classify subjectivity and objectivity documents.
- *Polarity sentiment classification*: aims to classify an subjectivity document into one of the three classes: "positive", "negative" and "neutral".
- *Spam detection*: aims to detect fake reviews and reviewers.
- *Rating*: rating the documents having personal opinions from 1 star to 5 star (very negative to very positive).

Besides these basic tasks, there are deeper studying tasks as follows:

- *Aspect-based sentiment analysis*: The goal is to identify the aspects of given target entities and the sentiment expressed for each aspect.
- *Opinion mining in comparative sentences*: This task focuses on mining opinions from comparative sentences, i.e., to identify entities to be compared and determine which entities are preferred by the author in a comparative sentence.

For popular language such as English, there are many campaign for this research topic. One of the most successful campaigns is described in [2]. Meanwhile, for Vietnamese language, so far there is no systematic comparison between the performance of Vietnamese sentiment analysis systems. The VLSP 2016 campaign, therefore, targets at providing an objective evaluation measurement about performance (quality) of sentiment analysis tools, and encouraging the development of Vietnamese sentiment analysis systems. As the first shared task on Sentiment Analysis, we just focus on the essential problem, that is polarity sentiment classification. We created a standard corpus for evaluating Vietnamese sentiment analysis systems which contains comments on technical articles from forums and social networks. This is actually the first benchmark dataset for this task. There are total eight submissions to the workshop and almost of them produce very promising results.

The remainder of this report is organized as follows: first, we describe the task, the preparation of the dataset and evaluation method; then, we summarize and discuss about the participating systems and their results and finally we make some conclusions on this campaign.

## II. Task Description

### A. Problem

The scope of the campaign this year is polarity classification, i.e., to evaluate the ability of classifying Vietnamese reviews/documents into one of three categories: positive, negative, or neutral. Other sentiment analysis tasks can be covered in the campaigns next years.

### B. Data preparation

*1) Data collection:* We collected data from three source sites which are Tinhte.vn, Vnexpress.net and Facebook. Our data consists of comments of technical articles in those sites. The quantities of comments are reported in Table I.

TABLE I
Analysis of data source

| No. | Source | Quantity |
|---|---|---|
| 1 | Tinhte.vn | 2710 |
| 2 | Vnexpress.net | 7998 |
| 3 | Facebook | 1488 |
| | **Total** | 12196 |

*2) Annotation procedure:* We have three annotators for our dataset. First, we split 12196 comments into three parts, one for each annotator. Each annotator had to give each comment one of four labels which are POS (positive), NEG (negative), NEU (neutral) and USELESS. Because a review can be very complex with different sentiments on various objects, we set some constraints on the dataset and used USELESS label to filter out the irrelevant comments. The constrains are:

- The dataset only contains reviews having personal opinions.
- The data are usually short comments, containing opinions on one object. There is no limitation on the number of the object's aspects mentioned in the comment.
- Label (POS/NEG/NEU) is the overall sentiment of the whole review.
- The dataset contains only real data collected from social media, not artificially created by human.

Normally, it is very difficult to rate a neutral comment because the opinions are always indeclinable to be negative or positive.

- We usually rate a review be neutral when we cannot decide whether it is positive or negative.
- The neutral label can be used for the situations in which a review contains both positive and negative opinions but when combining them, the comment becomes neutral.

After filtering the data, we had 2669 POS, 2359 NEG and 2122 NEU. Next, we changed the annotator for each part. After the annotators had labeled the their parts, we selected 2100 comments in each part for the next step. In the next step, we changed the annotator for each part again. The result of this step was compared to the ones in two previous steps. Then, discussions were made in order to reach agreement to the final result. The last step is selecting data for the evaluation campaign by removing all divergent comments (different labels by two annotators, including the data discussed and reached agreement). Finally, for each label, we had 1700 comments for training, 350 comments for testing.

### C. Evaluation

The performance of the sentiment classification systems will be evaluated using accuracy, precision, recall, and the F1 score.

$$\text{accuracy} = \frac{\#\text{ of correctly classified reviews}}{\#\text{ of reviews}} \quad (1)$$

Let $A$ and $B$ be the set of reviews that the system predicted as POS and the set of reviews with POS label, the precision, recall, and the F1 score of POS label can be computed as follows (similarly for NEG label):

$$\text{Precision} = \frac{|A \cap B|}{|A|} \quad (2)$$

$$\text{Recall} = \frac{|A \cap B|}{|B|} \quad (3)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

$$\text{Average\_F1} = \frac{\text{POS\_F1} + \text{NEG\_F1}}{2} \quad (5)$$

## III. Submissions and Results

There are eight teams participating in our campaign. We received full reports from five teams [3], [4], [5], [6], [7] and two short descriptions from two teams. The other one did not send us any papers. Generally, all of the participating systems treat our task as a classification problem and use statistical machine learning approaches with various feature extraction and selection techniques to solve it. From the experiments of the systems, we have some interesting points to discuss in the next sections.

### A. Methods and Features

The methods used by participating systems are presented in Table II. Support Vector Machine (SVM) is the most popular method chosen by the teams. Besides, neural network architectures such as multilayer neural network (MLNN) and long short-term memory (LSTM) network, are also used by two teams due to its success in the recent years. Other methods are maximum entropy (MaxEnt), perceptron, random forest, naive Bayes and gradient boosting which have been proved to be useful in NLP tasks. While almost teams tended to do experiments in individual models, there is one team (**sa3**) which tried to combine three models into one system using an ensemble methods [4].

TABLE II
Methods of participating systems

| Team | Methods | Features |
|---|---|---|
| sa1 | Perceptron<br>SVM<br>MaxEnt (best) | n-gram (1, 2, 3) on syllables,<br>dictionary of sentiment words and phrases |
| sa2 | SVM<br>MLNN (best)<br>LSTM | TF-IDF on 1,2-gram (best)<br>VietSentiWordNet<br>TFIDF-VietSentiWordNet |
| sa3 | Ensemble:<br>- Random forest<br>- SVM<br>- Naive Bayes | TF-IDF weighted n-gram (1, 2, 3) |
| sa4 | SVM | n-gram, booster word list,<br>reverser word list, emotion word list |
| sa5 | SVM<br>MLNN (best) | BOW<br>TF-IDF (best)<br>BOW-senti<br>TF-IDF-senti<br>Objectivity-score |
| sa6 | SVM | n-gram (1, 2 ,3) extracted on words,<br>n-gram (1, 2 ,3) extracted syllables,<br>n-gram (1, 2 ,3) extracted important words,<br>Word embedding (using GloVe),<br>Log-count ratio of n-gram,<br>Negation words |
| sa7 | Gradient boosting | TF-IDF on words<br>(remove words having low TF-IDF) |
| sa8 | No report | No report |

In term of features, almost all systems use the basic n-gram features. TF-IDF also plays an important role in many systems [4], [5], [6]. In addition, some systems use external dictionaries of sentiment words, booster words, reverser words

and emotion words to enrich their feature sets and help to gain better results [3], [7].

### B. Results

The best results of all teams are reported in Table III where systems are ranked by their average F1 scores. In case that a team had more than one system, the best one is marked with "best" in Table II. The highest score belongs to **sa1** team [7] who used MaxEnt model with n-gram features and phrase features extracted from hand-built dictionaries. In [7], the authors reported that with the same feature set, MaxEnt model significantly outperforms SVM by a gap of approximately 7% in term of F1 score. This strongly surprised us. The result of **sa1** is also much better than others'. We aware that their hand-built dictionaries of sentiment and intensity words may have an important effect on the result of the system in our test set.

TABLE III
RESULTS OF THE PARTICIPATING SYSTEMS (%)

| Team | Positive | | | Negative | | | Average |
|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | F1 |
| sa1 | 75.85 | 89.71 | 82.20 | 79.88 | 76.00 | 77.89 | 80.05 |
| sa2 | 72.42 | 74.29 | 73.34 | 69.94 | 69.14 | 69.54 | 71.44 |
| sa3 | 74.77 | 71.14 | 72.91 | 72.09 | 67.14 | 69.53 | 71.22 |
| sa4 | 68.11 | 72.00 | 70.00 | 60.59 | 70.29 | 65.08 | 67.54 |
| sa5 | 69.06 | 71.43 | 70.23 | 65.67 | 62.86 | 64.23 | 67.23 |
| sa6 | 71.80 | 70.57 | 71.18 | 67.10 | 59.43 | 63.03 | 67.11 |
| sa7 | 71.00 | 67.14 | 69.02 | 62.97 | 61.71 | 62.33 | 65.68 |
| sa8 | 21.25 | 4.86 | 7.91 | 44.72 | 67.71 | 53.86 | 30.89 |

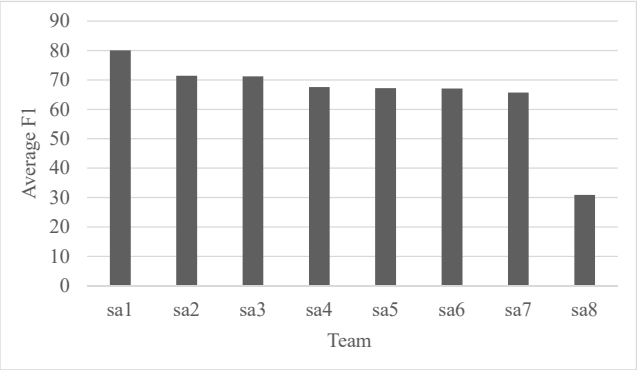



Fig. 1. Average F1 comparison.

The team **sa2** [6] only uses TF-IDF features in an MLNN to achieve a promising result: 71.44% for average F1. They also have experiments on SVM and LSTM with features extracted from VietSentiWordNet but the results are not as good as MLNN's. The ensemble system of **sa3** [4] combines three sub-systems which are random forest, SVM and naive Bayes. This system produces a good result at 71.22% for F1 score. The ensemble system also uses only TF-IDF weighted n-gram features.

Team **sa4** [3] used SVM as learning method combining with n-gram features and various other features extracted from external dictionaries that help to gain average F1 score at 67.54%. Next, the report of team **sa5** [5] also shows that MLNN outperforms SVM in our task. Various features is used by their system and they also found that TF-IDF helps to gain the best result. Meanwhile, the SVM-based system of team **sa6** uses various kind of features including n-gram on words, syllables, important words such as verb, noun, adjective, etc., word embedding, etc., however, its result is not as good as other SVM-based systems that make use of TF-IDF features.

## IV. CONCLUSION

We have described a new task in VLSP 2016 on Sentiment Analysis on Vietnamese texts for the first time. This first task attracted a good number of participants: 8 teams. All participating systems implement popular machine learning approach and also have many rich features to resolve the task.

In the campaign, the first dataset used to benchmark sentiment analysis systems for Vietnamese has been released. In fact, the dataset is quite simple because it only covers comments about just one object. However, we strongly believe that it will help to impulse the development of researching on this topic in the near future. In the next campaign, we hope that the new dataset will contain more complex cases, such as a review or comment can contain multiple objects and aspects with different sentiments. We also need other task such as aspect-based sentiment analysis.

## ACKNOWLEDGMENT

We would like to thank the sponsors for helping financially to the construction of the datasets for the sentiment analysis task. And special thanks to all the teams who have participated and contributed to the success of this evaluation campaign.

## REFERENCES

[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[2] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter," in *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*, 2016.

[3] N. V. Vi, H. V. Minh, and N. T. Tam, "Sentiment analysis for vietnamese using support vector machines with application to facebook," in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016.

[4] P. Minh Quang Nhat and T. T. Tran, "A lightweight ensemble method for sentiment classification task," in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016.

[5] T. Thy Thy, H. Xanh, and N. Nhung T.H., "A multi-layer neural network-based system for vietnamese sentiment analysis at the vlsp 2016 evaluation campaign," in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016.

[6] N. Hy, L. Tung, L. Viet-Thang, and D. Dien, "A simple supervised learning approach to sentiment classification at vlsp 2016," in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016.

[7] T. P. Quynh-Trang, N. Xuan-Truong, T. Van-Hien, N. Thi-Cham, and T. Mai-Vu, "Dsktlab: Vietnamese sentiment analysis for product reviews," in *The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP 2016)*, 2016.

# Sentiment Analysis for Vietnamese using Support Vector Machines with application to Facebook comments

**Vi Ngo Van**
Data Mining Team
Big Data Department, Admicro
VCCorp
Hanoi, Vietnam
Email: vingovan@admicro.vn

**Minh Hoang Van** and **Tam Nguyen Thanh**
School of Information and
Communication Technology
Hanoi University of Technology
Hanoi, Vietnam
Email: {hoangminh.it.hut,mrtamb9}@gmail.com

***Abstract*—Sentiment Analysis covers the area of research that studies people's opinions, sentiments, evaluations, attitudes and emotions from written text. This has become one of the most active research fields in Natural Language Processing, Text Mining and Data Mining in general. In recent years, with the rapid growth of World Wide Web, there has been an exponential increase in the number of texts available online. These days, people tend to express more and more opinions about various kinds of subjects in their lives. This communication of sentiments may have a great influence on others' decisions via social networking services, such as Facebook, Twitter and so forth.**
**In this paper, we present an effective Sentiment Analysis model for Vietnamese that is based on Support Vector Machine, an advanced Supervised Learning technique.**
**Keywords: Natural Language Processing, Text Mining, Sentiment Analysis, Vietnamese, Support Vector Machine, n-gram, Facebook, Supervised Learning.**

## I. Introduction

Facebook is one among the largest, most popular social networks in Vietnam. It has attracted a huge number of netizens who update their status, share their feelings, post comment and like others' postings on a daily basis. On this network, people also tend to give their either positive or negative reviews on products and services they have experienced. Therefore, the employment of information shared on Facebook surely helps companies to gather feedbacks from their customers. It is also useful when a company wants to do market survey or to learn more about the products of the rivals in order to adjust its own business strategy.

In a conventional way, customers' reviews are collected and analyzed completely by hands. It means that there should be a staff that are responsible for reading and grouping the feedbacks from different info channels. This job is obviously extremely tedious and requires a tremendous amount of time. When it comes to Facebook reviews, the task becomes impractical. Therefore, automating collecting and analyzing and classifying large amount of sentiment information become a must for companies professionally involved in data competitions.

However, fulfilling this goal is not easy. It is to say that doing sentiment analysis for formal text is difficult, doing sentiment analysis for informal text is even more challenging. We could name some of these challenges. First of all, Facebook postings and comments are often of variable lengths, some are short, some are long. This diversity makes it hard to apply techniques that require strict standardization. Second of all, informal language does not follow standard grammatical rule, i.e. no capitalized beginning letter, no punctuation, using abbreviations. Finally, sarcasms and ironies seem to be the hardest part to overcome.

Building a complete Customer Feedback System requires many processing steps from gathering information to exporting the final statistical report. However, in the scope of this paper, we focus only on the main part, opinion sentiment analysis. The core technique employed in our model is Support Vector Machines (SVM). This advanced classification method has demonstrated its strength in various Machine learning application.

## II. Related work

The existing methods applied to sentiment analysis task can be grouped into several classes as machine learning, lexicon-based, statistical and rule-based approaches. The rule-based approaches seek for words in a text that bring opinion or sentiment meaning and then classify based on the number of these positive and negative words. The lexicon-based approaches tend to compute sentiment polarity for a text according to the semantic orientation, a measure of subjectivity and opinion, of words. Statistical models tries to find the head terms, link them to the true aspects and sentiment them into ratings. Machine learning techniques employ sentiment knowledge in training dataset to give predictions. With the fast growth in the field of ML study, many strong tools have been applied to solve the problems of sentiment analysis. Since sentiment analysis is merely a text classification problem, any existing supervised learning method can be applied, e.g., Naive Bayes classifi-

cation or Support Vector Machines (SVM) (Joachims, 1999; Shawe-Taylor and Cristianini, 2000). The first paper to take these approaches to classify movie reviews into two classes, positive and negative was Pang et al. (2002). The sentiment analysis model presented in this paper is also motivated by this approach.

## III. Sentiment analysis model

### A. Model preparation

*1) Lexicon-sentiment list:* As noted above, sentiment classification is barely a text classification problem by its nature. Conventional text classification primarily groups documents of different topics, e.g., politics, society, sciences, education and sports. To achieve that goal, the key features are topic-related words. In contrast, in sentiment classification problem, sentiment and opinion words that indicate positive or negative are more important. e.g., *tốt, xấu, đẹp, tuyệt, tồi* etc. In these examples, *tốt, đẹp, tuyệt* are positive sentiment words and *xấu, tồi* are negative sentiment words. Most sentiment words are adjectives and adverbs, but nouns (e.g., *rác*) and verbs (e.g., *yêu, ghét*) can also be used to express sentiments. Apart from individual words, there are also sentiment phrases and idioms. Again, apart from sentiment words and phrases, there are also other expressions or language compositions that can be used to express or imply sentiment and opinions. This should be listed in the full lexicon-sentiment list.

To build this word list, we exploit data from over 3 millions comments crawled from various Vietnamese forums and webpages. This set of data is trained by using word2vec, a model of neural network to learn word embedding, and finally passed through a manual review process.

*2) Polarity reversers list:* Negations, or more generally, polarity reversers, create inconsistent words which are a major cause of errors for polarity classification. Let us consider the example "Sản phẩm này không tốt". This sentence contains *"tốt"*, which is a positive-meaning word in lexicon-sentiment list. However, because of the preceding word *"không"* the sentiment turns into negative. *"không"* and similar words are commonly called as negation. Thus, the capture of these negation words is important while analyzing sentence sentiment.

*3) Booster words list:* Booster words, or degree modifier, are the words that impact sentiment intensity by either increasing or decreasing the intensity. Consider the examples "quá", "rất", "hơi", these words express different levels of sentiment in sentence and usually make the sentence become less neutral.

*4) Emotion words list:* Facebook postings are usually short, informal and contain many Internet slangs and emoticons. While this piece of information is helpful and usually used to improve sentiment analysis accuracy, it is sometimes hard to analyze the texts from complex topics that present sarcasms and ironies like social or political discussions.

*5) Stop words list:* Stop words are common words in a given language that do not bring any important meaning. They appear in almost every sentence of different contexts; thus, the removal of these words from processed data usually helps improving performance of sentiment models. One important thing that we should take care of is that having the list of given stop words, it requires to filter text again to keep meaningful stop words. For example, the stop words like *"không", "rất", "quá"* could be used for extracting important sentiment meaning.

### B. Data preprocessing

The preprocessing stage of proposed SA model works as follows

- Word segmentation
- Removal of meaningless words
  - Proper nouns and proper noun phrases
  - Abbreviations
  - Stop words
  - Words that make non-sense in Vietnamese
- Extraction of uni-grams and bi-grams from text
- Determination of number and score of lexicon-sentiment words found in sentence. The rule is as follows:
  - If a sentiment word (positive or negative) is preceded by a booster word, we add score to sentiment polarity.
  - If a sentiment word (positive or negative) is preceded by a reversers, we add score to reversed sentiment polarity.
- Count emotion words.

Details about several steps of this preprocessing workflow are described below.

*1) Segmentation:* Before feeding raw data into the Sentiment Analysis model, we need to divide them into a sequence of component words. In English and several other languages using some forms of the Latin alphabet, the space is an ideal word delimiter. In Vietnamese, it is not words but syllables that are delimited, so a word could be composed by one or several tokens. There exists the case where a stand-alone token does not bring any meaning. According to this reason, a statistical learning tool for segmentation has been applied.

*2) POS tagging:* In the next step, part-of-speech (POS) tagging is the process of assigning a word to its right grammatical label, in order to truly perceive its role within the sentence. In the scope of Sentiment Analysis, we do not pay attention to the proper nouns. The words belonging to this class does not play any role in sentiment classification. Moreover, if we do not remove these words, they will naturally add more noise to the training set fed into SVM model.

Let us take an example, given the text about the infamous fly-in-the-bottle scandal of Tan Hiep Phat Beverage Group. If we do not remove the proper noun “Tan Hiep Phat” or proper noun phrase “ Tan Hiep Phat Beverage Group”, the model tends to learn and implicitly assign this key words to “negative” label which may bias the new coming results.

*3) Stop words removal:* Besides the mentioned above stop words and proper nouns, we also need to remove other meaningless tokens appeared in Vietnamese texts. These meaningless tokens are often composed of number or special characters. A simple way to accomplish this task is done by

using a full set of accepted signed and unsigned Vietnamese characters.

In the next part, we describe the core method using in our SA model.

### C. Method

*1) Support Vector machines:* Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given training samples with classes, SVM tends to output the optimal hyperplane, a high-dimensional space, which can be used for categorizing new coming samples. Intuitively, optimal separation could be achieved by generating the hyperplane that has the largest distance to the nearest data point of any class. In almost case where the dataset is not linearly separable, SVM use kernel function to map a space of data points onto a new space in which data is linearly classifiable. A tutorial on SVM and its formulation could be found in Burges (1998) and Cristiani and Shawe-Tailor (2000). For details of the application of this model we refer to Joachims (2001).

*2) n-gram and additional features:* Machine Learning techniques do not work directly with raw text data until we could find a way to convert its textual content into numerical representation. In a simple manner, this representation could be done by encoding value of each word feature as its presence by 0 or 1, or TF-IDF score of that word. Depending on the number n of taken words, we may come up with coressponding so-called n-gram features.

An issue when using n-gram language models are out-of-vocabulary (OOV) words. This issue is encountered when the input includes words which were not present in our built model's dictionary. In such a scenario, the n-grams in the corpus that contain an OOV word are ignored and n-gram probabilities are still smoothed over all the words in the vocabulary even if they were not observed.

In addition to n-gram features, we could optionally utilize prior knowledge of Vietnamese lexicon-sentiment words. This is done by concatenating number of positive/negative words, sentiment score on positive/negative polarity and number of positive/negative emotion-words with original n-gram representation of input sentence. In experimentation section, we review the result using this additional information as input to SVM model.

## IV. Experimentation

### A. Experimental setup

When using Support Vector Machines or many other Machine Learning techniques for Sentiment analysis task, we need to consider defining an appropriate set of hyperparameters. The two important factors that may affect the classification performance of SVM are C and Gamma, the parameters for nonlinear SVM with Gaussian Radial Basis function kernel.

As solution to linear separable problem, standard SVM seeks to find a margin that separates all data points. However, for a nonlinear problem, this can lead to poorly fit models. This issue leads to the emergence of the concept "soft margin" SVM that allows some data samples to be "ignored" or placed on the wrong side of the margin. Soft margin SVM helps to come up with a better overall fit while the impact of each individual support vector could be regularized by C, the parameter for the soft margin cost function.

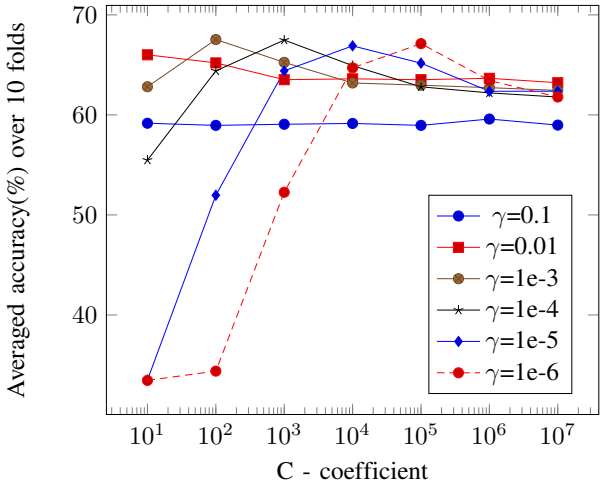



**Fig. 1** Accuracy for validation

Gamma plays a role as parameter handling non-linear classification. Let us say, when data is not linearly separable, we need to transform them to a higher dimension space or to raise them using nonlinear kernel function. For such a kernel function as Gaussian Radial basis function, Gamma is the free parameter to be adjusted.

There is a trade-off between setting large or small values of C and Gamma. In order to find the optimal hyperparameters setting of SVM model, we employ cross-validation framework, i.e. we first split the training data into folds and then do developing-and-validating model. The best values of C and Gamma are sought by looking at experimental result.

Training and test data contains 3 labels: positive, negative and neutral class, which brings neither positive nor negative meaning. When splitting data we make sure that percentage of each label class remains the same. This is to avoid causing bias in learning.

### B. Results

*1) Training and validation:* This section shows the performance of our proposed sentiment analysis model on varying values of SVM hyperparameters. Having 10 folds splitted from given training data, we calculate classification accuracy for each fold and average these values. We set C values in range of 100, 1000, 1e4, 1e5, 1e6, 1e7 and Gamma values in range of 0.1, 0.01, 1e-3, 1e-4, 1e-5, 1e-6. The results are then depicted in Figure 1. As shown in this plot, the highest accuracy is achieved at C=100 and Gamma=0.001. This setting is fixed and used for the next part of experiment.

*2) Experiment on test data:* Having well-tuned hyperparameters from previous experiment, we use this optimal setting for learning Sentiment analysis model applied to the whole given training data. To show the effectiveness of the proposed method, we compare its result to other models. The first

TABLE I
F1 SCORES(%) COMPARISON

| Labels - | SVM without lexicon-sentiment dictionary | SVM with lexicon-sentiment dictionary | Naive Bayes - |
|---|---|---|---|
| Positive | 62.8 | 63.07 | 61 |
| Negative | 58.39 | 60.13 | 55.36 |
| Neutral | 43.01 | 41.86 | 44.95 |
| Average | 54.73 | 55.02 | 53.77 |

model used for that comparison is SVM without using lexicon-sentiment features. The second model is Naive Bayes, a simple but effective tool with small number of training samples. With these methods, we compute precision and recall values for each class label. These measures are then used to calculate average F1 scores for comparison.The final results are depicted in table I.

As shown in table, the proposed model of combining SVM with features derived from n-gram and additional lexicon-sentiment representation gains the best score, 55.03% in average. Model with SVM and pure n-gram features stays second with 54.73% and Naive Bayes scores 53.77%.

## V. CONCLUSION

In this paper, an effective and complete model for Facebook Sentiment Analysis is presented. The model employs the strength of SVM in text classification along with building the features of words served for sentiment purpose.

In experimental part, the training data was firstly divided into 10 folds, which are used in cross-validation manner for developing and validating the model. We vary different settings of SVM to find the optimal hyperparameters. After that, we also conduct experiments to compare the results of SVM model with the baseline method using Naive Bayes. The experiments show our proposed model with n-gram and additional sentiment features performs the best in comparison to pure SVM model and baseline Naive Bayes. There is also to note that the average scores of all three models are not high. This could be caused of confusing data in neutral class, for which as a result we only gain the scores 43.01%, 41.86% and 44.95% respectively.

In general, sentiment analysis and opinion mining is still a challenging task. Sentiment analysis for Facebook and similar social interactive data is even more difficult. Unstructureness, large variance of length, are some among factors that may weaken many sentiment tools.

With recent achievements in Artificial Neural Network research, especially in the area of Deep Learning, there creates a big room for improving sentiment performance. Deep learning allows learning model to embed sentence structure and semantics well. These algorithm atempts to build representation of the entire sentence based on how the words are arranged and interact with each other, for example, *word2vec* and *paragraph vectors* have been shown to work very well. These methods are simple to train and implement. *Recurrent Neural Networks* like *LSTMs*, have proven to be able to gain very good performance. One important thing that should be in consideration while atempting to employ Deep Learning is to choose appropriate representation of input data for each applied method. Finally, in the top of this models, applying machine learning paradigms like ensemble learning may also help to improve the quality of our SA model.

In this work, to get sufficient data for building the dictionary of sentiment words, we use a set of manually-annotated text. For future work, we plan to improve the quality and coverage of this data source by automating the annotation process.

## ACKNOWLEDGMENT

This research was supported by Admicro, VCCorp (Vietnam Communications Corporation). The authors would like to thank our colleagues from Data Mining team, who provided insight and expertise that greatly assisted the research. We also thank Mr. Tuan Hoang Anh, CTO at Admicro, for his comments and Mr. Bao Nguyen Chi, Mr. Thanh Luong for their tremendous assistance that greatly improved the manuscript.

## REFERENCES

[1] J. Yi, T. Nasukawa, W. Niblack, R. Bunescu., *Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques*. In Proceedings of the 3rd IEEE international conferenc e on data mining (ICDM 2003), USA, pp. 427– 434,2003.

[2] T. Joachims: *Text categorization with support vector machines:learningwith many relevant features*. Proc. of ECML-98, 10th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE,pp. 137-142,1998.

[3] B. Liu, *Sentiment analysis and opinion mining. Synthesis lectures on human language technologies* , 5(1), pp.1-167.

[4] Y. Singh, P. K. Bhatia, and O. Sangwan, *A review of studies on machine learning techniques* International Journal of Computer Science and Security, vol. 1, no. 1, pp. 70–84, 2007.

[5] Mouthami, K., Devi, K.N. and Bhaskaran, V.M., *Sentiment analysis and classification based on textual reviews* In Information Communication and Embedded Systems (ICICES), 2013 International Conference on (pp. 271-276). IEEE.

[6] Zhou, X., Tao, X., Yong, J. and Yang, Z., *Sentiment analysis on tweets for social events* In Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on (pp. 557-562). IEEE, 2013, June.

[7] P.H. Shahana and B. Omman, *Evaluation of Features on Sentimental Analysis* Procedia Computer Science, 46, pp.1585-1592, 2015.

[8] Tripathy, A., Agrawal, A. and Rath, S.K., *Classification of Sentimental Reviews Using Machine Learning Techniques.* Procedia Computer Science,57,pp.821-829, 2015.

# A Simple Supervised Learning Approach to Sentiment Classification at VLSP 2016

Hy Nguyen*, Tung Le*, Viet-Thang Luong† and Dien Dinh‡
Faculty of Information Technology
University of Science VNU-HCM
* Email: {1212166, 1212494}@student.hcmus.edu.vn
† Email: vietthang.hcmus@gmail.com
‡ Email: ddien@fit.hcmus.edu.vn

***Abstract*—This paper describes our system which participates in opinion mining task of VLSP evaluation campaign. The system uses multilayer neural network classifier with unigram and bigram TF-IDF feature. Our system achieves a comparative result with 69.4% F1-score on test data.**

## I. Introduction

Opinion mining and sentiment analysis are the tasks to determine users opinion about products, movie, etc. In this topic, sentiment classification is a major task to classify opinion of a sentence or document into three categories: positive, negative and neutral. The prediction is extremely important because the users opinion becomes more and more value. The public interest is the main factor to affect the profit of products like movies, books, etc. Therefore, this problem is the interest of both researchers and companies.

In the context of VLSP competition, we build the system to determine the users opinion of a document into three labels: positive, negative and neutral. To make more practical, our system uses the simple features and classifier. However, our systems accuracy is really promising and acceptable with the low complexity.

The remainder of this paper is organized as follows. Section 2 provides a detail of our system. Section 3 describes the experimental setup and results. Section 4 concludes the paper and points to avenues for future work.

## II. System description

Figure 1 illustrates some processes of sentiment classification systems which determine the sentiment label for a sentence or a document from the input. After preprocessing data by removing low-frequency words, there are two approaches to build a sentiment classification system: feature extraction and deep learning. In feature extraction approach, we extract sentence or document feature vector like TF-IDF, VietSentiWordnet. Then these feature vector is input to a classifier such as Support Vector Machine (SVM) or Multilayer Neural Network (MLNN) to determine sentiment label of sentence or document. Besides, in deep learning approach, we just use Long Short Term Memory (LSTM) to determine the sentiment label of sentence or document. Section II-A and II-B demonstrate more details about feature used in feature extraction approach and about classifier (SVM, MLNN, LSTM) in both approaches.

### A. Features

This section briefly presents features which we try to use in feature extraction approach. The features extracted from each sentence or each document include:

*a) TF-IDF (Term Frequency * Inverse Document Frequency):* It usually used in information retrieval to determine which words are importance. This feature has solved the local and global information problem in feature extraction approach through TF and IDF score. In our experiment, we use TF-IDF score of unigram and bigram to extract the feature from a sentence or a document.

*b) VietSentiWordnet:* SentiWordNet is an important lexical resource supporting sentiment analysis in opinion mining applications [1]. A review is represented by a vector which length is the same as SentiWordNet vocabulary size. For each dimension, we compute the objectivity score corresponding to the term which it represents. The equation to compute the objectivity score is described by equation 1

$$score = 1 - (positivescore + negativescore) \quad (1)$$

*c) TFIDF-VietSentiWordNet:* this feature vector is created by concatenate both feature vectors above.

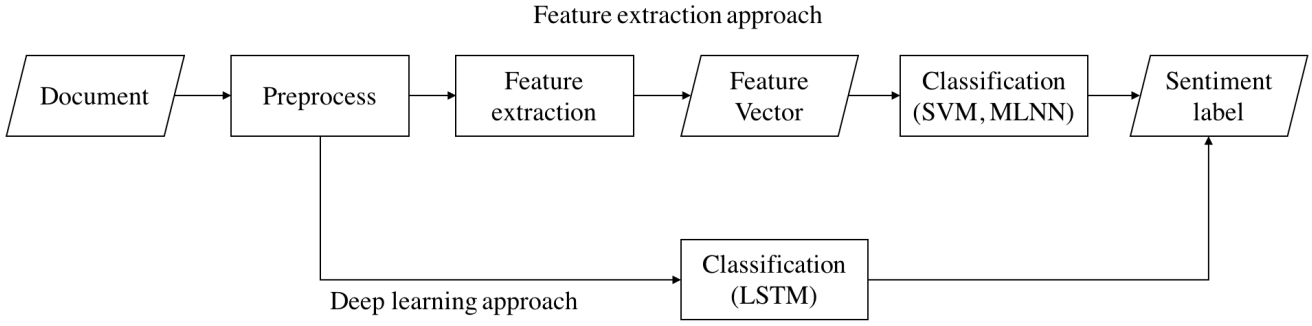
### B. Classification

Classifiers are divided into 2 groups based on the input: Firstly, Support Vector Machine and Multilayer Neural Network take a feature vector as input which is gotten after feature extraction stage. Secondly, the input of LSTM is a list of word vector which is generated randomly and updated by training.

- Support Vector Machine (SVM) [2] SVM is the classic supervised machine learning algorithm. The goal of SVM is to determine the hyperplane that has the largest distance to the support vectors. With its effectiveness in classification, SVM is popularly used in many areas as the hand-written recognizer, opinion mining, etc. In this work, we implement SVM from Scikit-Learn[1] with default parameters to classify feature vector.

[1]http://scikit-learn.org/

Fig. 1. Sentiment classification system




- Multilayer Neural Network (MLNN) In neural network classifier, sentence's features are integrated into a multilayer full connected network. The last layer use softmax function classify feature vector. We built a Multilayer Neural Network classifier with simple architecture to avoid the over-fitting problem, it has one hidden layer with 100 neural units inside. Particularly, MLNN was trained by stochastic gradient descent optimizer (learning rate 0.1) and it uses sigmoid function as activation function at hidden layer.
- Long Short Term Memory Network (LSTM) [3] LSTM is a special kind of Recurrent Neural Network proposed in 1997 by Hochreiter. In every step, LSTM determines what information need be saved, so it is capable of learning long-term dependencies through the time. We built LSTM classifier based on the public code from DeepLearning.net[2]. In details, LSTM has the size of word vector and hidden vector are 50 and 100 respectively. For optimizing LSTM, we used Adam[4], a new method for stochastic optimization.

## III. Experiments

### A. Dataset

To train and test our systems, we use the data provided by VLSP evaluation campaign in opinion mining task. It contains user's reviews about technological device following three categories: "negative", "positive" and "neutral". In details, table I shows the data distribution for opinion mining task.

TABLE I
Data distribution

| | Positive | Neutral | Negative | Total |
|---|---|---|---|---|
| Train | 1700 | 1700 | 1700 | 5100 |
| Test | 350 | 350 | 350 | 1050 |

### B. Systems setup

At figure 1, feature extraction approach consists of 2 stage extracting feature and classification. There are 3 kind of features: TF-IDF, VietSentiWordNet and TFIDF-VietSentiWordNet and 2 classifiers: SVM and MLNN. Therefore, we tried several systems which use a pair of feature and classifier. Example: SVM + TF-IDF is a system that uses Support Vector Machine classifier and TF-IDF feature. In deep learning approach, we have LSTM system which is described above.

[2]http://deeplearning.net/tutorial/lstm.html

### C. Metrics

Since we have to choose only one system which achieves the best performance to submit the result on test data, we compared systems by using F1-score. F1-score was chosen because it represents the balanced score between precision and recall score which are the evaluation metrics of VLSP opinion mining task. Table II shows F1-Score of our systems. Each one is average of F-Score of 10-Fold cross-validation.

### D. Results

In table II, the system uses MLNN classifier and TF-IDF feature perform the best result with 69.73% F1-score. Beside that, using VietSentiWordNet feature not only does not obtain good result but also reduce the result when combining with TF-IDF feature. Finally, although LSTM performs the good result, its results is less than system use MNLL and TF-IDF feature 1.83% F1-score. So we chose the system which uses MLNN classifier and TF-IDF feature to compete in VLSP opinion task. Table III illustrates the performance of competitive system on test data.

TABLE II
Our systems result

| Classifier | Feature | F1-score |
|---|---|---|
| SVM | TF-IDF | 68.16 |
| | VietSentiWordNet | 50.94 |
| | TF-IDF & VietSentiWordNet | 67.00 |
| MLNN | TF-IDF | **69.73** |
| | VietSentiWordNet | 50.97 |
| | TF-IDF & VietSentiWordNet | 68.09 |
| LSTM | | 67.90 |

TABLE III
PERFORMANCE OF COMPETITIVE SYSTEM

| | Precision | Recall | F1-Score |
|---|---|---|---|
| Negative | 69.94 | 69.14 | 69.54 |
| Neutral | 65.80 | 64.86 | 65.32 |
| Positive | 72.42 | 74.29 | 73.34 |
| Average | 69.39 | 69.43 | 69.40 |

## IV. CONCLUSION AND FUTURE WORDS

In this paper, we present several classification systems for opinion mining task of VLSP evaluation campaign. The systems follow two kinds of approaches: feature extraction and deep learning. In training data, the system use multilayer neural network classifier with unigram and bigram obtains the best result up to 69.63% F1-score through 10 fold cross-validation. In testing data, this system also achieves 69.4% F1-score.

In future, we will try to use different of features such as POS tag, NER, and semantic label and try to combine feature extraction and deep learning approach. It is regarded as the promising development of this work.

## REFERENCES

[1] X. Vu and S. Park, "Construction of vietnamese sentiwordnet by using vietnamese dictionary," *CoRR*, vol. abs/1412.8010, 2014. [Online]. Available: http://arxiv.org/abs/1412.8010

[2] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

# A Lightweight Ensemble Method for Sentiment Classification Task

Minh Quang Nhat Pham, Tran The Trung
FPT Technology Research Institute (FTRI)
8 Ton That Thuyet, My Dinh 2, Nam Tu Liem, Ha Noi, Viet Nam
{minhpqn, trungtt}@fpt.edu.vn

***Abstract*—In this report, we describe our system for sentiment classification task at VLSP 2016 evaluation campaign. Sentiment classification is to classify documents, articles, or product reviews into classes the reflect their sentiments about some subject matters. We propose a lightweight ensemble method for the task. Our ensemble model combines three classification models, namely Random Forests, Support Vector Machines, and Multinomial Naive Bayes by the majority voting strategy. In feature extraction, we use *n*-gram features ($n = 1, 2, 3$) with TF-IDF weighting scheme to train three classifiers. Our system obtained 69.6% accuracy over all classes, and 70% F1-score on average.**

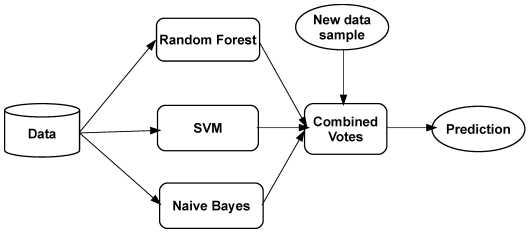



Fig. 1. System architecture

## I. Introduction

Sentiment analysis is the task of mining opinions in sentences, users' reviews, or articles according their sentiment towards some subject matters such as products [1]. Sentiment analysis is useful in many business intelligence applications. For instance, sentiments of product reviews give us a quick summary of users' opinions about products. The task has received extensive attentions of natural-language-processing and data mining communities since early 2000s.

For Vietnamese language, VLSP 2016 (Vietnamese Language and Speech Processing) evaluation campaign is the first effort to provide the benchmark data and to perform a systematic comparison between Vietnamese sentiment analysis systems. This year, the scope of the campaign is polarity classification in which participant systems need to classify Vietnamese reviews/documents into one of three categories: "positive", "negative", or "neutral."

In this report, we describe the sentiment classification system which we used to produce our submission for the VLSP 2016 evaluation campaign. We proposed a lightweight ensemble method which uses the majority voting strategy to combine three classification models trained with Random Forests algorithm [2], Support Vector Machine (SVM) [3] and Multinominal Naive Bayes [4]. We also give an analysis on errors that our system made and discuss some directions for further improvements.

The rest of the paper is organized as follows. In section II, we describe our participant system. In section III, we present our evaluation results on the test set. In section IV, we give the error analysis. Finally, section V gives conclusions about the work.

## II. System Description

Figure 1 shows the architecture of our participant system. After preprocessing data, we train three classifiers using the training data set and combine three classifiers with majority voting strategy. We use three algorithms: Random Forests [2], Support Vector Machines (SVM) [3] with linear kernel, and Multinomial Naive Bayes (MNB) [4]. Random Forests is an ensemble method that combine multiple tree predictors. Random Forests algorithm have been shown to be an effective method for classification problems [2]. We choose SVM and Multinomial Naive Bayes with the same reason. That is because previous work has shown that they are very effective for sentiment classification task [5]. In our system, since the number of features is much larger than the number of training examples and by some preliminary experiments, we decided to use linear kernel in SVM.

To train three classification models, we use TF-IDF weighted *n*-grams features, in which $n = 1, 2, 3$. In our implementation, we use python scikit-learn library [6] for both extracting features and training models.

An important step to improve the prediction accuracy of a machine-learning system is model selection. We tune parameters for classification models by using grid search and evaluating F1-score of 5 folds in 5-fold cross validation. Parameter tuning is done on the training set provided by the VLSP 2016 evaluation campaign. More specifically, for Random Forest algorithm, we tune the number of trees in the forest. For the linear SVM model, we select the parameter *C*.

In preprocessing step, we just perform word segmentation by using tool vnTokenizer [7]. To produce our submitted results, we re-used the word-segmented data provided by the organizer of VLSP 2016 evaluation campaign.

TABLE I
DISTRIBUTION OF CLASSES IN DATA SETS

| | positive | negative | neutra | Totall |
|---|---|---|---|---|
| Training set | 1700 | 1700 | 1700 | 5100 |
| Test set | 350 | 350 | 350 | 1050 |

TABLE II
EVALUATION RESULTS OF OUR SYSTEM

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| positive | 0.75 | 0.71 | 0.73 | 350 |
| negative | 0.72 | 0.67 | 0.70 | 350 |
| neutral | 0.63 | 0.71 | 0.67 | 350 |
| avg/total | 0.70 | 0.70 | 0.70 | 1050 |

TABLE III
CONFUSION MATRIX 1. ROWS ARE NUMBERS OF INSTANCES WITH ACTUAL LABELS. COLUMNS ARE NUMBER OF INSTANCES WITH PREDICTED LABELS

| Predicted<br>Actual | negative | neutral | positive | all |
|---|---|---|---|---|
| negative | 235 | 75 | 40 | 350 |
| neutral | 59 | 247 | 44 | 350 |
| positive | 32 | 69 | 249 | 350 |
| All | 326 | 391 | 333 | 1050 |

## III. EVALUATION RESULTS

### A. Data sets

We use the training data provided by the organizer to train our ensemble model. The trained model is used to predict the labels for examples in the test data. Table I shows the distribution of classes in the training and test data. The table indicated that the data sets are perfectly balanced.

### B. Evaluation measures

VLSP 2016 evaluation campaign uses accuracy, precision, recall, and F1 score as evaluation measures. In this report, we report precision, recall, F1 score for all three classes in the data set.

### C. Results

Table II shows our evaluation results (precision, recall, f1 score for each class) on the test set. We obtained 69.62% accuracy.

The results indicated that in the evaluation data, the class "positive" is the easiest class and the class "neutral" is the most difficult to predict.

## IV. ERROR ANALYSIS

### A. Confusion Matrix

Table III shows the confusion matrix between gold standard labels and predicted labels. Table IV is the confusion matrix with normalized values (probabilities).

We can see that our system predicted "neutral" with the highest number of instances. However the precision for the class "neutral" is lowest. We hypothesize that it is because "neutral" reviews may contain both positive and negative opinions and "neutral" reviews are cases where even annotators find they are difficult to decide whether it is positive or negative.

TABLE IV
CONFUSION MATRIX 2. CONFUSION MATRIX WITH PROBABILITY VALUES

| Predicted<br>Actual | negative | neutral | positive | all |
|---|---|---|---|---|
| negative | 0.22 | 0.07 | 0.04 | 0.33 |
| neutral | 0.06 | 0.23 | 0.04 | 0.33 |
| positive | 0.03 | 0.06 | 0.24 | 0.33 |
| All | 0.31 | 0.37 | 0.32 | 1.0 |

| id | Gold | Predicted | Review |
|---|---|---|---|
| 1 | POS | NEG | Nước_miếng rơi rơi ... . nước_mắt rưng_rưng ! |
| 2 | NEG | NEU | Nếu vẫn thế_này thì R.I.P Apple . Đây là cơ_hội cho Samsung có những bước_ngoặt lớn |
| 3 | NEG | POS | Mình dùng suface_pro_3 thấy chạy tốt mà đẹp hơn con này |
| 4 | POS | NEG | ZenFone_3 không làm tui thất_vọng . Đang trên tay , ngất_ngây con gà_tây ! |
| 5 | POS | NEU | con này dùng hay phết gprs chát_chít vèo_vèo . ai có game nào hay post lên đây đi . mình tìm được mấy cái game nhung chơi chán quá |
| 6 | NEG | NEU | Mắc kinh , cơ_mà kính đeo vào cái phá được kỷ_lục Olympic thì 120.000 $ ngta cũng mua |
| 7 | POS | NEU | Mình đang dùng 1 con này rất ngon , mạng 1-2 vạch , dùng nó kéo lên 5 vạch , mạng miếc nhanh hơn hẳn ( dĩ_nhiên cũng chỉ hòm_hòm khi lại gần router thôi ) . Mình lên công_ty , chỗ mình ngồi wifi công_ty chỉ 2-3 vạch là kịch , mà phải đúng chỗ ngồi , còn đi khuất tí là rất khó vào mạng . Từ hồi có nó có_thể lỉnh vào 1 góc , chui xuống gầm bàn xem phim , chơi game ... mà thằng này lại khá mạnh , mình đi cách nó hơn 10m , xuyên qua 1 bức tường ( toalet ) mà vẫn 4 vạch . Nhưng nó có 1 nhược_điểm đang cố tìm cách khắc_phục là nó chỉ nhớ 1 mạng . Nghĩa_là nếu cài mạng ở công_ty thì khi về nhà hay đi công_tác , gặp mạng khác phải reset nó và cài lại từ đầu . |

Fig. 2. Some miss-classified instances made by our system

### B. Some phenomena

For each class (positive, negative, neutral), we get samples of 30 examples of that class, for which the system failed to predict their gold labels. In total, we analyze 90 examples to investigate phenomena that our system could not capture. We found some phenomena that are difficult as follows. Figure 2 shows some instances extracted from our sample.

In some cases, we need to use common sense knowledge to infer the sentiment of a review. Examples 1 and 2 in the Figure 2 belong to this kind. Solving this problem is difficult because inference using common sense knowledge is still a challenging problem in natural language processing [8].

We find that an user may use comparison to express her/his opinion. In example 3, the user compare a product with its rival product to convey her/his negative opinion.

One of problems in analyzing product reviews on social medias is dealing with slang, rare words, *teen codes* or *trolling* comments of users. Example 5 and 6 are two examples of that phenomena.

Another phenomenon we found by error analysis is that a review may contain the main part that discusses about the product in the question and other parts discuss related things. Example 7 is an example of that phenomenon. In that case, the sentiment of the main part decides the sentiment of the whole

review. Other parts of review may give noises to the machine learning algorithm. In such cases, we need to identify the main phrase or sentence that shows the sentiment about the product.

## V. Conclusion

In this report, we present our participant system for the sentiment classification task at VLSP 2016 evaluation campaign. We adopted a very lightweight ensemble method that combined three classification models trained on the training data using Random Forests, Support Vector Machines, and Multinomial Naive Bayes. The method is lightweight and easy to implement with the library scikit-learn. Despite its simplicity, the system obtained 0.70 F1-score on average over all classes. We also analyse errors made by our system and discuss some difficulties of the task.

## References


[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86. [Online]. Available: http://dx.doi.org/10.3115/1118693.1118704

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[4] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[5] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ser. ACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 90–94. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390665.2390688

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] L. Hong Phuong, N. Thi Minh Huyen, A. Roussanaly, and H. T. Vinh, *A Hybrid Approach to Word Segmentation of Vietnamese Texts*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 240–249. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88282-4_23

[8] P. LoBue and A. Yates, "Types of common-sense knowledge needed for recognizing textual entailment," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 329–334. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002736.2002805

# DSKTLAB: Vietnamese Sentiment Analysis for Product Reviews

Quynh-Trang Thi Pham[1], Xuan-Truong Nguyen[1], Van-Hien Tran[1], Thi-Cham Nguyen[1,2], Mai-Vu Tran[1]

[1]Data Science and Knowledge Technology Laboratory

University of Engineering and Technology, Vietnam National University, Hanoi

[2]Haiphong University of Medicine and Pharmacy

{trangptq_58, hientv, truongnx_58, vutm}@.vnu.edu.vn, nthicham@hpmu.edu.vn

***Abstract— Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). Sentiment analysis has gained much attention in recent years. In this paper, we aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. We proposed a Vietnamese Sentiment Analysis system with detailed descriptions, which classifies the opinion of a review into one of three types: "positive", "negative" and "neutral". Data used in this study are provided by The Fourth International Workshop on Vietnamese Language and Speech Processing (VLSP2016). Our system has achieved an accuracy score of 81% on the sample set and 76.38% on the final test set.***

***Keywords—Vietnamese sentiment analysis, maximum entropy, SVM, product reviews***

## I. INTRODUCTION

Sentiment is an attitude, thought prompted by feeling of people. In recent years, the explosion of social networking sites, blogs and review sites provide a lot of valuable information [1]. Millions of people express their uninhibited opinions about product features, etc. via posting their own content through various social media like online social networking sites. Therefore, sentiment analysis has a strong fundament with the available support of massive online data.

However, these types of online data have some problems that prevent the process of sentiment analysis. Firstly, the quality of their opinions cannot be guaranteed since people can freely post their own content. For example, some spammers post spam or fake opinions on forums. Secondly, because of a free writing style, people use abundant emoticons, abbreviations, slang, etc. It is very difficult for analysis and understanding exactly product reviews in review-level categorization.

From a technical point of view, there are some approaches on resolving sentiment analysis, namely *machine learning*, *lexicon-based*, *statistical* and *rule-based* approaches. The machine learning method uses several learning algorithms to determine the sentiment by training on a labeled dataset. The lexicon-based approach involves calculating sentiment polarity for a review using the semantic orientation of words or sentences in the review. The rule-based approach considers different rules for classification like dictionary polarity, negation words, emoticons, mixed opinions. Statistical models represent each review as a mixture of latent aspects and ratings.

In this paper, we focus on classifying the opinion of a review into one of three types, which are "positive", "negative" and "neutral". Based on sample datasets provided by VLSP campaign, we built a Vietnamese Sentiment Analysis system using various machine learning methods, along with effective selected features. In section 2, we describe the specific task and datasets provided. Section 3 presents our approach with a hybrid machine learning model to resolve the task. The evaluation and analysis is given in Section 4. Finally, section 5 discussed our conclusions and directions for future work.

## II. SA TASK DESCRIPTION

The main aim of the campaign VLSP 2016 on sentiment analysis for Vietnamese language is to evaluate the ability of classifying Vietnamese reviews/documents into one of three categories: "positive", "negative", or "neutral". This task only focuses on one language - Vietnamese. Vietnamese has specific characteristics different from other languages as English. Unlike English, Vietnamese is more very difficult and complicated about grammar, vocabulary, etc. Therefore, sentiment analysis methods successfully applied in the English corpus could not make the corresponding result for data use Vietnamese. It requires a separately optimal model which is more suitable for Vietnamese analysis sentiment.

Sample dataset provided by VLSP campaign includes three files: POS file (positive reviews), NEU file (neutral reviews), NEG file (negative reviews) with the same 1700 reviews on each file. These files are collected from comments on mobile review forums and labeled by people. The data is balanced for training our model. The test dataset contains 1050 reviews for evaluating the model. To understand and resolve the problem better, we carry out statistical analysis on training dataset in the below table.

| Measure | NEG file | NEU file | POS file |
|---|---|---|---|
| The number of sentences | 2841 | 3510 | 2792 |
| Sum of characters | 195082 | 297831 | 179714 |
| Average length of each sentence | 69 | 85 | 64 |

| | | | |
|---|---|---|---|
| Sum of all words | 38734 | 58343 | 35346 |
| Sum of unique words | 7256 | 9078 | 6592 |

**TABLE 1**. SOME STATISTICS OF THE TRAINING DATASET.

In general, measures between NEG file and POS file are more balanced, while these figures for NEU file are larger than two remaining files. It shows that when giving a review on positive or negative opinion, a user tends to write more concisely. Reviews on neutral opinion are often longer because they can contain both positive and negative opinions although they are neutral when combining two opinions. Besides, a neutral review often includes comparable sentences, analysis sentences on aspects of products. The average length of each review is quite short so it can be difficult to hold syntactic information as well as other important information.

Another problem is negation opinion. For example, a positive opinion is "Sản phẩm này *đẹp*"/"This product is *nice*" but this opinion becomes negative when adding "không"/"not" as: "Sản phẩm này *không đẹp*"/ "This product is *not nice*". Besides, the word "không"/"no" is written in a free style writing in Vietnamese such as: 'khong', 'ko', 'khg', 'k', etc. These problems also hinder the process of sentiment analysis. After the preprocessing (POS tagger), we statistic the number of unique nouns, verbs, adjectives in each file and the appearance frequency of each word in the below table.

| Type of Word | NEG file | NEU file | POS file |
|---|---|---|---|
| Noun | 766/15 | 825/22 | 679/15 |
| Verb | 2718/3 | 3357/4 | 2456/3 |
| Adjective | 1654/2 | 1994/2 | 1463/2 |

**TABLE 2**: SOME STATISTICS OF THE TRAINING DATASET.

Basically, the neutral file uses a number of very abundant adjectives. It can be easy to understand since neutral reviews contain both negative opinions and positive opinions. We also built two dictionaries: sentiment dictionary and intensity dictionary for supporting our model. From our model built, it is evaluated on the test dataset, depending on classifying result of each review. In the next section, we will present our approach to resolve the task.

## III. OUR APPROACH

### *A. Features Selection*

Based on survey training dataset and recent researches on sentiment analysis [2,3,4], we decided to utilize two main features for machine learning models, including n-gram feature and phrase feature.

Firstly, n-gram is a very successful approach to represent text because of holding word sequences. The benefit of using n-gram instead of single words as features comes in being able to capture some dependencies between the words and the importance of individual phrases. Thus, N-gram is used for developing features for supervised machine learning model such as SVM, Maxent. We intend to use Bigram for machine learning models. However, since the number of n-gram features is quite large because of abundant sample dataset, we filter n-gram features based on chi-square measure to remain important n-gram features.

Secondly, to use phrase feature, we built manually two dictionaries: Intensity Dictionary (191 words) and Sentiment Dictionary (5760 words). Some keywords in Intensity Dictionary such as 'rất'/'very', 'khá'/'quite' and Sentiment Dictionary contains some terms like 'đẹp'/'nice', 'tốt'/'good'. Since a sentiment word and an intensity word often appear together in a sentence, we find out the occurrence of them in a maximum length of 3 words. We use it as an important feature for machine learning models.

### *B. Classifier Selection*

In this task, we select three typical machine learning methods using for analysis sentiment task: Maximum Entropy, Support Vector Machine and Perceptron. We access three models separately and compare the performance among them.

## IV. EVALUATION

### *A. Experimental Data*

As said above, we received sample datasets including three files: POS file (positive reviews), NEU file (neutral reviews), NEG file (negative reviews) with the same 1700 reviews on each file. Firstly, we use 10-folds cross-validation to evaluate three machine learning methods. These results are presented in the below table:

| | POS | NEU | NEG |
|---|---|---|---|
| **Perceptron** | 65.88 | 51.07 | 61.78 |
| **SVM** | 78.59 | 72.12 | 73.14 |
| **MaxEnt** | 85.59 | 79.66 | 82.41 |

**TABLE 3**: 10 FOLDS CROSS-VALIDATION ON THREE VARIOUS METHODS

As can see from the above table, MaxEnt method gives the best result. Thus, we intend to use Maxent model for classifying review opinions for test dataset (1050 reviews provided by VLSP). The result is given in the table below:

| | Precision | Recall | F1 |
|---|---|---|---|
| Accuracy | 76.38 | | |
| Positive | 75.85 | 89.71 | 82.2 |
| Negative | 79.88 | 76.0 | 77.89 |

**TABLE 4**: THE EVALUATION RESULT ON THE TEST SET.

### *B. Experimental Results and Analysis*

In general, the result is quite satisfactory. It proves the effective of two main features selected as well as Maxent machine learning method.

## V. Future Work and Conclusion

We intend to select and build proper features to improve the performance of the current system. Besides, we will use some hybrid models to achieve better results in the future.

## References

[1] Thi-Ngan Pham, Thi-Hong Vuong, Thi-Hoai Thai, Mai-Vu Tran, Quang-Thuy Ha, Sentiment Analysis and User Similarity for Social Recommender System: An Experimental Study

[2] Huyen-Trang Pham, Tien-Thanh Vu, Mai-Vu Tran, Quang-Thuy Ha, A Solution for Grouping Vietnamese Synonym Feature Words in Product Reviews

[3] Yelena Mejova, Padmini Srinivasan, Exploring Feature Definition and Selection for Sentiment Classifiers, 2011, Association for the Advancement of Artificial Intelligence

[4] Subhabrata Mukherjee, Pushpak Bhattacharyya, Feature Specific Sentiment Analysis for Product Reviews.

[5] Xing Fang and Justin Zhan, Sentiment analysis using product review data, 2015 Fang and Zhan, Journal of Big Data.

# A Multi-layer Neural Network-based System for Vietnamese Sentiment Analysis at the VLSP 2016 Evaluation Campaign

Thy Thy Tran*, Xanh Ho† and Nhung T.H. Nguyen‡
*Faculty of Information and Technology*
*VNUHCM University of Science*
*Email: *thy2512@gmail.com, †xanhhocntt@gmail.com, ‡nthnhung@fit.hcmus.edu.vn*

***Abstract*—We present a description of our system submitted to the VLSP 2016 Evaluation Campaign of Sentiment Analysis for the Vietnamese Language. This year the campaign focussed on polarity classification, i.e., to classify Vietnamese reviews or documents into positive, negative or neutral. In order to address the task, we implemented a multi-layer neural network-based method that uses three types of features as input. Our internal evaluations indicate that by using TF-IDF feature to represent sentences, we can obtain the best performance with 66% of precision and 65% of recall. The official accuracy of the proposed method on the testing set (evaluated by the organiser) is 65.9%.**

## 1. Introduction

Recently, with the explosive of social media, there is a high demand from brands and industries to automatically analyse the customers' comments on their products so that they can know how consumers perceive their products as well as those of their competitors. This sentiment information is not only useful for marketing and product benchmarking but also useful for product design and product development [1]. Extracting opinion or sentiment from text can be defined as sentiment analysis or opinion mining. The task receives raw texts talking about brands or products as input, and outputs sentiment information, which indicates the author's opinions about a specific brand, product, or about products' features. In general, the task of sentiment analysis can be divided into three levels: document-based, sentence-based, and aspect-based. The VLSP 2016 evaluation campaign of sentiment analysis for the Vietnamese language focussed on the first level[1], i.e., classify input documents or reviews into one over three sentiment classes: positive, negative and neutral.

Most approaches for sentence-based sentiment analysis are supervised learning-based methods such as Naive Bayes, support vector machine (SVM) and neural networks (NN). In this paper, we have implemented multilayer NN (MLNN) for Vietnamese sentiment analysis. Specifically, our model is a fully connected neural network with only one hidden layer and gets feature vector as input. We have conducted experiments to compare the proposed method with SVM implementation from scikit-learn [2] and fastText from facebook [3]. For SVM, we also extract the same features that we use for MLNN. Meanwhile, regarding fastText, the model first randomizes a weight matrix that is a lookup table over words, then the word representations are averaged into a text representation and experience classification on it. Experimental results show that MLNN with TF-IDF feature as input produced the best scores among the models. We have applied the best performing method to the testing set and obtained an official accuracy of 65.9%.

1. http://vlsp.org.vn/evaluation_campaign_OM

## 2. Related Work

While research on sentiment analysis for English has been being grown and obtained the state of the art [4], [5], [6], [7], [8], there is a few work for Vietnamese. Kieu and Pham [9] introduced a rule-based system using the GATE framework for sentence-level sentiment analysis. They conducted their experiments on a corpus of computer product reviews and obtained 61.16% of precision, 64.62% of recall. Ha et al. [10] described an extension of a feature-based opinion mining and summarizing model to extract sentiments from reviews on mobile phone products. Feature words and opinion words were extracted based on some Vietnamese syntactic rules. Opinion orientation and summarization on features were determined by using VietSenti WordNet. Their model produced 69.16% of precision and 68.86% of recall. Nguyen et al. [11] proposed an approach extract opinions from Vietnamese documents using a domain specific sentiment dictionary. The sentiment dictionary is built incrementally by applying statistical methods to 20,083 comments about mobile products crawled from the Internet. Nguyen et al. [12] introduced an annotated corpus for document-level sentiment analysis that consists of hotel reviews. They implemented several machine learning-based methods that use different types of features. Their experimental results indicated that using word-based features produced better performance than using syllable-based ones. Among different types of $n$-grams, only unigrams were effective for the task.

## 3. System Overview

For each input document, we extract a feature vector that can represent as exactly as possible the characteristic of the input. Specifically, we extract three types of features: Bag-Of-Word (BOW), TF-IDF and SentiWordNet-based features. Then, each document vector based on the above features along with its label will be fed to a classifier to learn a model that can determine the sentiment class.

### 3.1. Feature Extraction

BOW and TF-IDF are common features that have been used in text mining as well as other NLP problems. While SentiWordNet-based features are extracted based on Viet-Senti Wordnet [13]–a lexicon resource that contains sentiment expressions and its three types of scores. Details about these features will be presented below.

**3.1.1. BOW.** Each review is represented by a sparse vector as the bag of its words over a fixed vocabulary, ignoring grammar and word order.

**3.1.2. TF-IDF.** TF-IDF reflects how importance is a word to a document in the corpus. It is composed by two terms: (1)Term Frequency (TF) computes the number of times that word occurs in the current document, and (2) Inverse Document Frequency (IDF) shows how much information a term provides. In this work, we use TF-IDF scores as review representation vectors.

**3.1.3. SentiWordnet-based.** VietSentiWordNet [13] is a synset of words that express sentiment. However, it contains not only sentiment words but also phrases with some included sentiment shifter such as *not*. Therefore, we have to extract $n$-gram, range from unigram to pentagram, to map vocabulary of VietSentiWordnet. In details, we use the lexicon in three ways:

- **BOW-senti** We extract the BOW feature based on a fixed vocabulary extracted from VietSenti Wordnet. The whole collocation contains 1198 terms with their corresponding frequency of $n$-grams showed in Table 1.
- **TF-IDF-senti** We also used the VietSentiWordnet to compute TF-IDF scores.
- **Objectivity-score** The Wordnet also provides information of the sentiment scores involving positive, negative, and objectivity scores extracted from positive and negative as

$$ObjScore = 1 - (PosScore + NegScore)$$

Based on the objectivity score, we build a feature vector represents VietSenti Wordnet, for each sentiment word/phrase appears in a sentence, we add its corresponding objectivity score.

TABLE 1. VIETSENTIWORDNET $N$-GRAM OCCURENCE

| $N$-gram | Count | Example |
|---|---|---|
| **1-gram** | 885 | 'khủng_khiếp' |
| **2-gram** | 236 | 'tốt nhất' |
| **3-gram** | 61 | 'giá quá cao' |
| **4-gram** | 13 | 'không có ý định' |
| **5-gram** | 3 | 'không tạo được cảm hứng' |

TABLE 2. CHARACTERISTICS OF THE PROVIDED DATA

| | POS | NEG | NEU | Total |
|---|---|---|---|---|
| **TRAIN** | 1700 | 1700 | 1700 | 5100 |
| **TEST** | 350 | 350 | 350 | 1050 |

### 3.2. Classification

In this work, we use three different algorithms to classify reviews' polarity. The first one is a linear **support vector machine** (**SVM**) classifier provided by the scikit-learn toolkit [2]. The second is an implementation of **multilayer neural network** (**MLNN**) using NumPy which provides multidimensional arrays and functions [14]. The last is an extra experiment using a recent released library named **fastText** [3].

Regarding the SVM classifier, we use the linear kernel which handles multiclass classification by a one-vs-rest scheme. In details, the strategy involves training a single classifier per class in order to produce a confidence score for its decision, then the label that has the highest confidence would be the predicted class.

For the MLNN model, we conducted experiments with several architectures and hyperparameters include learning rate, and $l2$ regularization scale. Specifically, we use Stochastic Gradient Descent (SGD) to optimize the objective function and use a flag for early stopping if the validation accuracy is not increased after 20 epochs. The extracted features will be used as input to the network.

For comparison, we also train a supervised classifier using fastText, which is a library for learning word representations and sentence classification introduced by Facebook.

We evaluate all the above-mentioned models using cross-validation to assess how the models generalize for an independent data.

## 4. Experiments

### 4.1. Dataset

The provided data this year (2016) is in the domain of technical devices, of which the training data consists of 5100 reviews with an equal proportion for each sentiment class. The data distribution is presented in Table 2. A noticeable point is that a neutral label will be assigned to a review if the review contains both positive and negative opinions. Table 3 illustrates this phenomenon, which makes the classification task more difficult.

TABLE 3. A REVIEW THAT IS LABELED AS NEUTRAL SINCE IT CONTAINS BOTH POSITIVE AND NEGATIVE SENTIMENT

| |
|---|
| Rất tốt nha bạn , hơi cấn ngón_cái chút vì có hai nút bấm gồ lên , xài lâu cũng quen , hơi nặng chút , nếu bạn xài Mac nên đầu_tư một con mighty để bàn , con này đem đi_lại , về kết_nối bluetooth thì cảm_giác lag hơn Magic_mouse , còn nếu dùng usb receiver thì rất ngon , battery thì mình dùng một tháng nay chưa hết , không_bao_giờ Off nguồn . |

TABLE 4. PERFORMANCES OF SVM WITH DIFFERENT TYPES OF FEATURES

| SVM | Precision | Recall | F1-score |
|---|---|---|---|
| **BOW** | 0.61 | 0.62 | 0.61 |
| **TF-IDF** | **0.65** | **0.65** | **0.65** |
| **BOW-senti** | 0.44 | 0.38 | 0.34 |
| **TF-IDF-senti** | 0.44 | 0.39 | 0.34 |
| **Objectivity-score** | 0.49 | 0.48 | 0.47 |

## 4.2. Evaluation Metrics

Evaluation metrics in this work include accuracy, precision, recall, and F1 score. However, we only compute the last three ones for each model while running cross-validation, including the average scores for three categories (positive, negative, and neutral) and the total scores overall. The purpose is to select the best performing model that will be applied to the official testing set of the sentiment analysis campaign.

## 4.3. Experimental Results

Table 4 presents the performances of SVM when we use five separate features as input through cross-validation. Overall, TF-IDF obtains the highest scores for all precision, recall, and F1 with 0.65, 0.65 and 0.65, respectively. Meanwhile, the performance when using features extracted from the VietSenti Wordnet, is even worse than that of BOW.

Regarding MLNN, we designed its architecture as follows. The input layer has a dimension that depends on the dimension of the feature vectors. The hidden layer has 100 computational units. The last layer with 3 neurons stands for the number of labels which is positive, negative, and neutral. The neural structure was chosen as the model that produces the best results after several running and evaluating using cross-validation Table 5 shows the performance of MLNN using the same features with SVM. As can be seen, using TF-IDF as input for MLNN also performs the best among our features, similarly to SVM. Specifically, the F1-score over the three sentiment classes is also 0.65, which is equal to the best score produced by SVM. However, MLNN produces a little higher precision score (0.66 vs. 0.65 by SVM).

Training a classifier using fastText is a bit different with that for SVM and MLNN. We do not need to feed into the input layer the extracted features, the model already contains a random initialization weight matrix as a word representation look up table. The matrix would be adjusted through the training step. Then, the text representation is composed by averaging its word representations before transform to the classification layer. By using fastText, we obtained the same scores of precision and recall with 0.498.

TABLE 5. PERFORMANCES OF MLNN WITH DIFFERENT TYPES OF FEATURES

| MLNN | Precision | Recall | F1-score |
|---|---|---|---|
| **BOW** | 0.64 | 0.63 | 0.63 |
| **TF-IDF** | **0.66** | **0.65** | **0.65** |
| **BOW-senti** | 0.43 | 0.37 | 0.29 |
| **TF-IDF-senti** | 0.39 | 0.37 | 0.30 |
| **Objectivity-score** | 0.47 | 0.45 | 0.43 |

Among the three classifiers, when using SVM and MLNN with TF-IDF features, we obtained the best score of 0.65 F1-score. However, precision produced by MLNN is a little bit higher than that by SVM. Therefore, we select MLNN with TF-IDF features to apply to the official testing set of the campaign. The results on the testing set are presented in Table 6.

TABLE 6. OFFICIAL TESTING RESULTS ON INDIVIDUAL CLASS. THE OVERAL ACCURACY IS 65.9%

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| **POS** | 69.06 | 71.43 | 70.23 |
| **NEG** | 65.67 | 62.86 | 64.23 |

## 4.4. Discussion

The above-mentioned experimental results indicate that with this specific data of Vietnamese sentiment analysis, SVM and MLNN produce comparable scores for all metrics. This situation is a bit conflict with the trend in English natural processing, in which neural networks beat state-of-the-art in many areas such as speech recognition [15], language modeling [16], and especially sentiment classification [8]. However, such situation can be explained by the fact that the provided training data is smaller than that for the English. Our dataset involves only 146,440 tokens and the vocabulary size is only 2,248 words. Meanwhile, to obtain the state-of-the-art, the English sentiment tree bank consists of 184,837 tokens with informative treebanks. This means that in order to effectively train a neural network-based model, we need a bigger training data with more linguistic information.

Another interesting phenomenon is that, although the training data has a balanced ratio of sentences in each class, the average length among them is quite significant different as showed in Table 7. The average length of a neutral review is nearly 6 to 7 words longer than a positive or negative review. Likewise, the number of tokens in neutral reviews is also close to twice times of that in positive or negative ones. These points make the problem hard to solve the imbalance word occurrence over three classes. Even in our final model, the average performance through cross-validation also bias to learn well on positive class than the other two, and often classify the neutral label incorrectly (see Figure 1).

TABLE 7. AVERAGE LENGTH (IN WORDS) OF DOCUMENTS IN THE TRAINING DATA

| | POS | NEG | NEU | Total |
|---|---|---|---|---|
| **Avg. length** | 11.67 | 12.60 | **18.79** | 14.36 |
| **Num of tokens** | 39689 | 42845 | **63906** | 146440 |

Figure 1. Precision, Recall and F1-score of MLNN using TF-IDF over three categories: positive (POS), negative (NEG) and neutral (NEU)


## 5. Conclusion

This paper has described our system submitted to the sentiment analysis evaluation campaign of the VLSP 2016. To conclude, for small data, the traditional feature is still better than neural networks with word representation. Due to the limited collocation of the SentiWordnet, it is not a good choice to use separately as the only feature. Moreover, our experiments only use distinct features as input without combining them, it should be considered as an avenue for the further work.

## References

[1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, May 2012. [Online]. Available: http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[3] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[4] R. Tong, "An operational system for detecting and tracking opinions in on-line discussions," in *Working Notes of the SIGIR Workshop on Operational Text Classification*, New Orleans, Louisianna, 2001, pp. 1–6.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *IN PROCEEDINGS OF EMNLP*, 2002, pp. 79–86.

[6] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Volume 50*, pp. 723–762, 2014.

[7] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," 2009.

[8] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[9] B. T. Kieu and S. B. Pham, "Sentiment Analysis for Vietnamese," IEEE CS. IEEE CS, 2010.

[10] Q. Ha, T. Vu, H. Pham, and C. Luu, "An Upgrading Feature-Based Opinion Mining Model on Vietnamese Product Reviews," in *Active Media Technology - 7th International Conference, AMT 2011, Lanzhou, China, September 7-9, 2011. Proceedings*, 2011, pp. 173–185. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23620-4$_2$1

[11] H. N. Nguyen, T. V. Le, H. S. Le, and T. V. Pham, "Domain Specific Sentiment Dictionary for Opinion Mining of Vietnamese Text," in *Multi-disciplinary Trends in Artificial Intelligence - 8th International Workshop, MIWAI 2014, Bangalore, India, December 8-10, 2014. Proceedings*, 2014, pp. 136–148. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-13365-2$_1$3

[12] N. T. Duyen, N. X. Bach, and T. M. Phuong, "An empirical study on sentiment analysis for Vietnamese," in *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, Oct 2014, pp. 309–314.

[13] X.-S. Vu and S.-B. Park, "Construction of vietnamese sentiwordnet by using vietnamese dictionary," *The 40th Conference of the Korea Information Processing Society*, vol. 21, pp. 745–748, 2014.

[14] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science and Engg.*, vol. 13, no. 2, pp. 22–30, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1109/MCSE.2011.37

[15] G. E. Dahl, M. Ranzato, A. Mohamed, and G. E. Hinton, "Phone recognition with the mean-covariance restricted Boltzmann machine," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 469–477.

[16] T. Mikolov, "Statistical language models based on neural networks," Ph.D. dissertation, Ph. D. thesis, Brno University of Technology, 2012.

# Báo cáo kỹ thuật hệ thống thông tin Bất Động Sản

Nguyễn Thành Dương, Nguyễn Thị Ngọc Tú, Nguyễn Thị Thu Hà, Bùi Khánh Linh
Khoa CNTT, trường Đại học Điện lực
Hà nội, Việt Nam
{duongnt_d7cnpm, tunn, hantt, linhbk}@epu.edu.vn

Nguyễn Việt Anh
Viện CNTT, Viện HLKHVN
Hà nội, Việt Nam
nva.nguyen@gmail.com

***Abstract*—Trong báo cáo này, chúng tôi trình bày giải pháp xây dựng hệ thống thông tin bất động sản có sử dụng một số kỹ thuật xử lý ngôn ngữ tự nhiên: độ tương tự thông tin, trích rút thực thể. Tiếp theo đó, chúng tôi sử dụng các kỹ thuật phân tích thông tin bất động sản dựa trên dữ liệu đã được tổng hợp về hệ thống một cách tự động theo thời gian thực. Các kết quả trả về trên trang tra cứu của người dùng cuối trên một số nền tảng hệ điều hành khác nhau như windows, ios và android.**

***Keywords—tra cứu thông tin, trích rút thực thể, bất động sản, độ tương tự thông tin.***

## I. GIỚI THIỆU

Thị trường bất động sản ở bất cứ quốc gia nào luôn là một thị trường đặc biệt bởi đất đai là một loại hàng hóa đặc thù so với hàng hóa khác bởi tính "bất động" của nó. Hầu hết tại mọi thời điểm cũng có giao dịch dù ít hay nhiều, dù thị trường có bị "đóng băng" hay "sốt".

Ở Việt Nam hiện nay, các dự án được phát triển nhiều, số giao dịch khá lớn. Mỗi dự án thông thường được rao bán bởi nhiều công ty khác nhau, do đó người mua hoặc những nhà đầu tư thường gặp các khó khăn bởi lượng thông tin bị "nhiễu" và khó khăn khi tìm đúng địa chỉ bán hoặc tìm mức giá phù hợp với dự án mình định đầu tư.

Các công ty bất động sản hiện tại ở Việt Nam hoặc những sàn giao dịch trực tuyến trên Internet thông qua những website để đưa tin cần mua, cần bán. Tuy nhiên, những thông tin này đôi khi khó kiểm chứng được sự chính xác của thông tin hoặc sự trùng lặp của các tin tức khi đưa lên. Ngoài ra, các mức giá rao bán hầu hết được định giá dựa trên kinh nghiệm hoặc mong muốn của người bán hoặc công ty môi giới nhà đất mà chưa có một định lượng cụ thể với mỗi tiêu chí giả sử như: vị trí, hướng, hình dạng đất, phong thủy, giao thông,…

Với những thông tin đưa lên rời rạc như vậy, việc quản lý tin tức bất động sản trở nên khó khăn hơn. Khó định giá, khó phân tích thị trường, phân tích giá cả, phân tích xu hướng hay dự báo thông tin của thị trường bất động sản.

Nhận thấy, dữ liệu bất động sản cần được tổng hợp và lưu trữ phục vụ cho mục đích tra cứu, phân tích, thống kê và dùng cho tương lai. Nhóm đã thực hiện những nhiệm vụ sau:

Tổng hợp, lưu trữ thông tin bất động sản từ nhiều nguồn khác nhau và cập nhật theo thời gian thực.

Trích rút các thực thể bao gồm: tên người, tên địa danh, thời gian, tiền tệ và lưu trữ cơ sở dữ liệu hệ thống.

Thực hiện tra cứu thông tin với nhiều tiêu chí khác nhau, hiển thị trực quan trên bản đồ số.

Phân tích dữ liệu: phân tích giá, phân tích quan điểm khách hàng, thống kê giao dịch theo thời gian, theo địa điểm,…hiển thị dữ liệu phân tích và thống kê trên biểu đồ.

Định giá bất động sản.

Dự báo xu hướng giá bất động sản.

Trong khuôn khổ của báo cáo này, chúng tôi dự kiến trình bày các mục: 1, 2, 3, 4 và tập trung trình bày mục trích rút thực thể. Phương pháp tiếp cận của đề tài dựa trên xây dựng các luật, mẫu dựa trên thông tin cấu trúc và trình bày của tài liệu, kết hợp với những từ điển, ontologies và thư viện sẵn có của GATE để rút trích các thông tin bất động sản trên các mẩu tin quảng cáo.

## II. LỰA CHỌN PHƯƠNG PHÁP NHẬN DIỆN VÀ TRÍCH RÚT THỰC THỂ

Các thông tin bất động sản được rao trên Internet là những thông tin phi cấu trúc, không đồng nhất về hình thức.

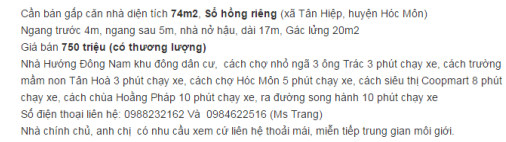
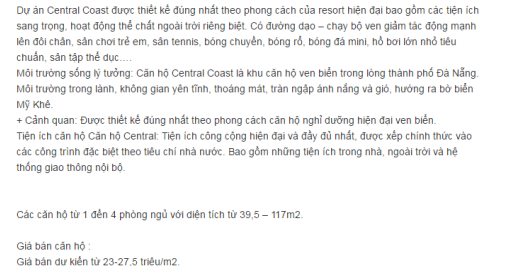

Cần bán gấp căn nhà diện tích **74m2**, **Sổ hồng riêng** (xã Tân Hiệp, huyện Hóc Môn)
Ngang trước 4m, ngang sau 5m, nhà nở hậu, dài 17m, Gác lửng 20m2
Giá bán **750 triệu (có thương lượng)**
Nhà Hướng Đông Nam khu đông dân cư, cách chợ nhỏ ngã 3 ông Trác 3 phút chạy xe, cách trường mầm non Tân Hoà 3 phút chạy xe, cách chợ Hóc Môn 5 phút chạy xe, cách siêu thị Coopmart 8 phút chạy xe, cách chùa Hoằng Pháp 10 phút chạy xe, ra đường song hành 10 phút chạy xe
Số điện thoại liên hệ: 0988232162 Và 0984622516 (Ms Trang)
Nhà chính chủ, anh chị có nhu cầu xem cứ liên hệ thoải mái, miễn tiếp trung gian môi giới.

Fig. 1. Tin rao bán nhà


Dự án Central Coast được thiết kế đúng nhất theo phong cách của resort hiện đại bao gồm các tiện ích sang trọng, hoạt động thể chất ngoài trời riêng biệt. Có đường dạo – chạy bộ ven giảm tác động mạnh lên đôi chân, sân chơi trẻ em, sân tennis, bóng chuyền, bóng rổ, bóng đá mini, hồ bơi lớn nhỏ tiêu chuẩn, sân tập thể dục….
Môi trường sống lý tưởng: Căn hộ Central Coast là khu căn hộ ven biển trong lòng thành phố Đà Nẵng. Môi trường trong lành, không gian yên tĩnh, thoáng mát, tràn ngập ánh nắng và gió, hướng ra bờ biển Mỹ Khê.
+ Cảnh quan: Được thiết kế đúng nhất theo phong cách căn hộ nghỉ dưỡng hiện đại ven biển.
Tiện ích căn hộ Căn hộ Central: Tiện ích công cộng hiện đại và đầy đủ nhất, được xếp chính thức vào các công trình đặc biệt theo tiêu chí nhà nước. Bao gồm những tiện ích trong nhà, ngoài trời và hệ thống giao thông nội bộ.

Các căn hộ từ 1 đến 4 phòng ngủ với diện tích từ 39,5 – 117m2.

Giá bán căn hộ :
Giá bán dự kiến từ 23-27.5 triệu/m2.

Fig. 2. Tin rao bán chung cư

Trong hình 1 là tin rao bán nhà đất, hình 2 là tin rao bán chung cư, hai thông tin trên không cùng một định dạng và giá tiền cũng không cùng một đơn vị tính. Để xử lý thông tin trong máy tính cần tách các dữ liệu trên, nhận dạng các thực thể để lưu trữ, sau đó mới có thể thực hiện các phân tích, so sánh và đánh giá.

Chúng tôi sử dụng cách tiếp cận theo luật để trích rút thực thể và sử dụng GATE framework để định nghĩa các luật trích rút.

GATE là một kiến trúc, một nền tảng và một môi trường phát triển giao diện cho các ngôn ngữ kỹ thuật. Nó được tạo ra và phát triển bởi một nhóm các nhà phát triển dẫn đầu bởi giáo sư Cunningham tại đại học Sheffield từ năm 1995. Hiện nay, nó được sử dụngrộng rãi trên thế giới bởi cộng đồng các nhà nghiên cứu thuộc nhiều lĩnh vực của xử lý ngôn ngữ, đặc biệt là rút trích thông tin. Nó được sử dụng cho nhiều dự án rút trích thông tin của nhiều ngôn ngữ và miền vấn đề. Một ví dụ điển hình của hệ thống rút trích thông tin là ANNIE (A Nearly-New Information Extraction System). Nó được đóng gói như một plugin trong GATE.

## III. XÂY DỰNG HỆ THỐNG

### A. *Thông tin trích rút*

Qua quá trình quan sát các dữ liệu thu thập được, nhóm lựa chọn và quyết định chọn các thực thể cần trích rút trong một tin quảng cáo nhà đất.

- Loại tin (VietnameseType)
- Loại nhà đất (VietnameseCategory)
- Diện tích (VietnameseSquare)
- Giá cả (VietnamesePrice)
- Dự án (VietnameseProject)
- Nhà đầu tư (VietnameseInvestor)
- Đường phố (VietnameseStreet)
- Phường xã (VietnameseWard)
- Quận huyện (VietnameseDistrict)
- Tỉnh thành (VietnameseCity)
- Tên người đăng tin (VietnameseName)
- Số điện thoại (VietnamesePhone)
- Email (VietnameseEmail)

### B. *Một số luật trích rút thông tin*

- Luật trích rút giá

Luật trích rút giá được mô tả như sau

```
Phase: VietnamesePriceExtraction
Input: Token Lookup
Options: control = appelt

Macro: AMOUNT_NUMBER
(
    ( {Token.kind == number}
    ( ( {Token.string == ","} | {Token.string == "." } ) {Token.kind ==
number} )* )
)

Macro: VI_PRICE_PREFIX
(
      { Lookup.majorType == "price", Lookup.minorType == "price_prefix",
Lookup.language == "vi" }
)

Macro: VI_PRICE_SUFFIX
(
      { Lookup.majorType == "price", Lookup.minorType == "price_suffix",
Lookup.language == "vi" }
)

Rule: VietnamesePriceExtraction
(VI_PRICE_PREFIX)?
(
    (AMOUNT_NUMBER)
    (VI_PRICE_SUFFIX)
): price
-->
: price.VietnamesePrice = { kind = "VietnamesePrice", rule =
"VietnamesePriceExtraction" }
```

- Luật trích rút diện tích

```
Phase: VietnameseSquareExtraction
Input: Token Lookup
Options: control = appelt

Macro: AMOUNT_NUMBER
(
    ( {Token.kind == number}
    ( ( {Token.string == ","} | {Token.string == "." } ) {Token.kind ==
number} )* )
)

Macro: VI_SQUARE_PREFIX
(
      { Lookup.majorType == "totalSquare", Lookup.minorType ==
"square_prefix", Lookup.language == "vi" }
)

Macro: VI_SQUARE_SUFFIX
(
      { Lookup.majorType == "totalSquare", Lookup.minorType ==
"square_suffix", Lookup.language == "vi" }
)

Rule: VietnameseSquareExtraction
(VI_SQUARE_PREFIX)?
(
    (AMOUNT_NUMBER)
    (VI_SQUARE_SUFFIX)
): totalSquare
-->
: totalSquare.VietnameseSquare = { kind = "VietnameseSquare", rule =
"VietnameseSquareExtraction" }
```

- Luật trích rút địa chỉ

```
Phase: VietnameseLocationExtraction
Input: Token Lookup
Options: control = appelt

Macro: VI_CITY_PREFIX
({ Lookup.majorType == "location", Lookup.minorType == "city_prefix",
Lookup.language == "vi" })
Macro: VI_CITY
({ Lookup.majorType == "location", Lookup.minorType == "city",
Lookup.language == "vi" })
Rule: VietnameseCityExtraction
( (VI_CITY_PREFIX)? (VI_CITY): city )
-->
: city.VietnameseCity = { kind = "VietnameseCity", rule =
"VietnameseCityExtraction" }
Macro: VI_DISTRICT_PREFIX
({ Lookup.majorType == "location", Lookup.minorType == "district_prefix",
Lookup.language == "vi" }
Macro: VI_DISTRICT
({ Lookup.majorType == "location", Lookup.minorType == "district",
Lookup.language == "vi" })
Rule: VietnameseDistrictExtraction
((VI_DISTRICT_PREFIX)?(VI_DISTRICT): district)
-->
: district.VietnameseDistrict = { kind = "VietnameseDistrict", rule =
"VietnameseDistrictExtraction" }
Macro: VI_WARD_PREFIX
({ Lookup.majorType == "location", Lookup.minorType == "ward_prefix",
Lookup.language == "vi" })
Macro: VI_WARD
({ Lookup.majorType == "location", Lookup.minorType == "ward",
Lookup.language == "vi" }
Rule: VietnameseWardExtraction
((VI_WARD_PREFIX)(VI_WARD): ward)
-->
: ward.VietnameseWard = { kind = "VietnameseWard", rule =
"VietnameseWardExtraction" }
Macro: VI_STREET
({ Lookup.majorType == "location", Lookup.minorType == "street",
Lookup.language == "vi" })
Macro: VI_STREET_PREFIX
({ Lookup.majorType == "location", Lookup.minorType == "street_prefix",
Lookup.language == "vi" })
Rule: VietnameseStreetExtraction
((VI_STREET_PREFIX)(VI_STREET) : street)
-->
: street.VietnameseStreet = { kind = "VietnameseStreet", rule =
"VietnameseStreetExtraction" }
```

- Luật trích rút tên người đăng tin

```
Phase: VietnameseNameExtraction
Input: Token Lookup
Options: control = appelt

Macro: VI_NAME_PREFIX
(
      { Lookup.majorType == "name", Lookup.minorType == "prefix",
Lookup.language == "vi" }
)

Macro: VI_NAME_FIRST
(
      { Lookup.majorType == "name", Lookup.minorType == "first",
Lookup.language == "vi" }
)

Macro: VI_NAME_LAST
(
      { Lookup.majorType == "name", Lookup.minorType == "last",
Lookup.language == "vi" }
)

Rule: VietnameseNameExtraction
(
    (
        (VI_NAME_PREFIX)
        ((VI_NAME_LAST) (VI_NAME_FIRST)+) : full_name
    ) |
    (
        (VI_NAME_PREFIX)
        ((VI_NAME_FIRST)+) : first_name
    )
)
-->
: full_name.VietnameseName = { kind = "VietnameseName", rule =
"VietnameseNameExtraction" },
: first_name.VietnameseName = { kind = "VietnameseName", rule =
"VietnameseNameExtraction" }
```

### C. *Kiến trúc hệ thống*

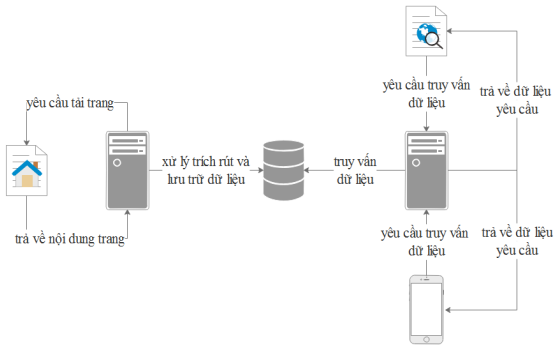



Fig. 3. Sơ đồ hệ thống thu thập, trích rút và truy vấn trên thiết bị smartphone và tablet

### D. *Một số giao diện*

Hệ thống được phát triển trên 03 nền tảng hệ điều hành:

- Hệ điều hành windows, chạy trên nền desktop.
- Hệ điều hành Android chạy trên smartphone và tablet
- Hệ điều hành IOS chạy trên smartphone và tablet

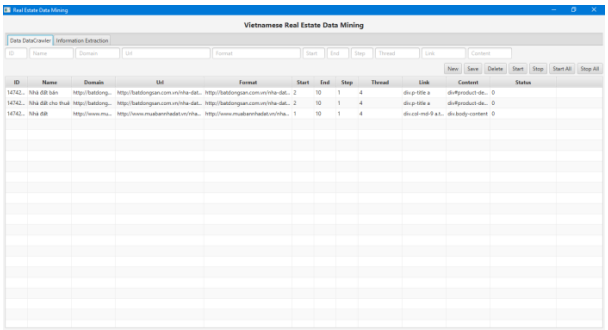



Fig. 4. Giao diện thu thập dữ liệu

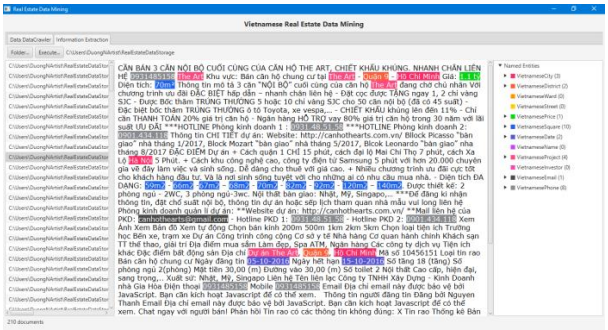



Fig. 5. Giao diện gán nhãn thực thể và trích rút

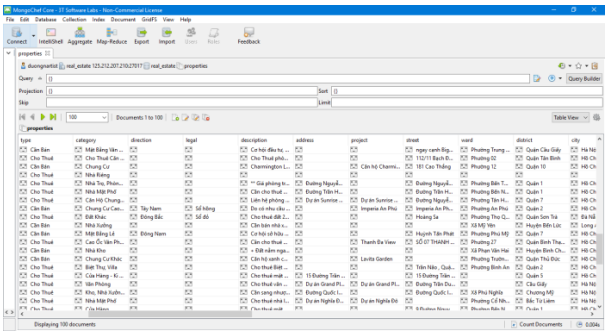


Fig. 6. Dữ liệu chuẩn hóa sau khi trích rút

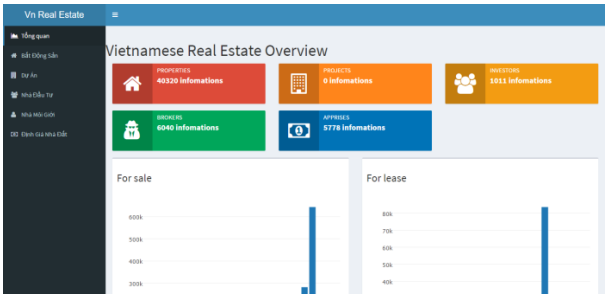



Fig. 7. Giao diện web phân tích

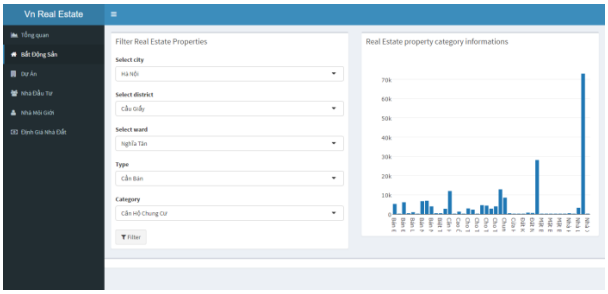



Fig. 8. Tra cứu bất động sản theo khu vực


Fig. 9. Phân tích bất động sản theo khu vực


Fig. 10. Tra cứu thông tin trên di động

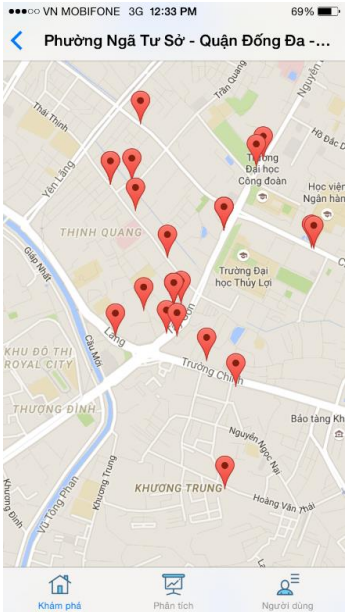



Fig. 11. Hiển thị dữ liệu trên bản đồ số


Fig. 12. Lọc dữ liệu

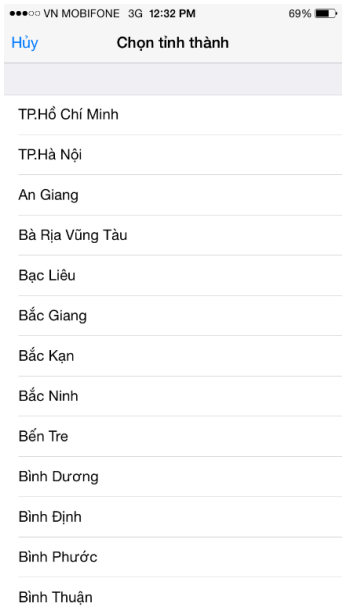



Fig. 13. Lựa chọn theo tỉnh thành

## IV. ĐÁNH GIÁ THỰC NGHIỆM

### A. *Thực nghiệm và đánh giá*

Nhóm đã tiến hành thực nghiệm với 2800 tin quảng cáo từ trang web batdongsan.com.vn. Để đánh giá kết quả qua cách tiếp cận nhóm sử dụng các độ đo truyền thống được dùng trong truy vấn thông tin đó là độ chính xác Recall (R), độ tin cậy (P) và độ đo F-measure.

$$R = \frac{tp}{(tp+tn)}; \quad P = \frac{tp}{(tp+fp)}; \quad F = \frac{2\times P\times R}{(P+R)}$$

Trong đó tp: số kết quả đúng được tìm thấy

tn: số kết quả đúng mà không tìm thấy

fp: số kết quả tìm thấy mà không đúng

### B. *Kết quả*

Kết quả thực nghiệm được đo trên một số thuộc tính của tin bất động sản như trên trang batdongsan.com.vn và kết quả được thể hiện trong bảng dưới đây:

TABLE I. KẾT QUẢ THỰC NGHIỆM

| Metadata | Precision (%) | Recall (%) | F-Measure (%) |
|---|---|---|---|
| Type | 96.55% | 93.33% | 94.92% |
| Category | 97.44% | 88.05% | 92.51% |
| Price | 100% | 100% | 100% |
| Square | 100% | 100% | 100% |
| Date | 100% | 100% | 100% |
| Name | 92.72% | 89.47% | 91.07% |
| Email | 100% | 100% | 100% |
| Phone | 100% | 100% | 100% |

## V. KẾT LUẬN

Với mục tiêu xây dựng cơ sở dữ liệu và khai phá dữ liệu bất động sản từ các tin quảng cáo mua bán nhà đất để hướng đến xây dựng một hệ thống thị trường bất động sản thông minh và chính xác hơn. Nhóm tập trung nghiên cứu tổng quan về lĩnh vực trích rút thông tin từ văn bản, tra cứu thông tin, trích rút thực thể và các phân tích thống kê ứng dụng trên đa nền tảng.

## TÀI LIỆU THAM KHẢO

[1] Nancy A. Chinchor, MUC-7 Named Entity Task Definition (Version 3.5), Message Understanding Conference, 1998.

[2] Line Eikvil. Information Extraction from World Wide Web – A Survey. Norwegian Computing Center, PB, Citeseer. July 1999.

[3] https://gate.ac.uk/ie/

[4] http://www.batdongsan.vn/

[5] https://nhadatso.com/?gclid=CIW14r7J388CFQsjvQod2aYFAQ

[6] https://muaban.net/mua-ban-nha-dat-cho-thue-ho-chi-minh-l59-c3

[7] http://tinbatdongsan.com/

[8] http://alonhadat.com/

# VAIS-TTS: A High Quality Speech Synthesis Service for Vietnamese

Quoc Truong Do
Vietnam Artificial Intelligent System Team (VAIS)
Email: truongdo@vais.vn

Chi Mai Luong
Information Technology Institute of Vietnam
Email: lcmai@ioit.ac.vn

***Abstract*—In this paper, we describe our high quality speech synthesis service for Vietnamese based on speech concatenation and hidden semi-Markov models (HSMMs) techniques. The main goal is to provide a high quality, high availability, and easy to access speech synthesis service for everyone including end-users and developers. The system provides two female voices for Northern and Southern regions of Vietnam. The service can be accessed via API end-points including HTTP and Websocket connection. Moreover, we also provide SDK toolkits allowing developers to easily integrate our services into their products.**

## I. Introduction

Speech synthesis is an active research area that has been well studied and deployed in many softwares and companies. However, there is not yet a public service provides high quality speech synthesis technologies for Vietnamese language.

In this paper, we present the first public speech synthesis service for Vietnamese[1] including 2 female voices for both Northern and Southern accents using speech concatenation and HMM techniques. Furthermore, the speech synthesis engine is also available for offline access.

The basic system architecture is described in Fig. 1. The main components of the system is the Web server and the speech server. The web server manages user registration and API keys and the speech server processes requests from clients. All components are designed to be easily scale up and scale down with zero downtime.

## II. Available voices

At the current stage, we provide 2 speech concatenation female voices with Northern and Southern accent and 1 HSMM voice. All voices are made available for online access and the HSMM voice is also available for offline access including PC and mobile platforms.

The dataset used for training models is described as follows:

- **Southern accent:** 5.8k utterances of female voice collected from VOV audio speech. The length of each utterance varies from 5 to 15 words.
- **Northern accent:** 6k utterances of female voice. The length of each utterance varies from 14 to 17 words.

[1]The service can be accessed at http://vais.vn

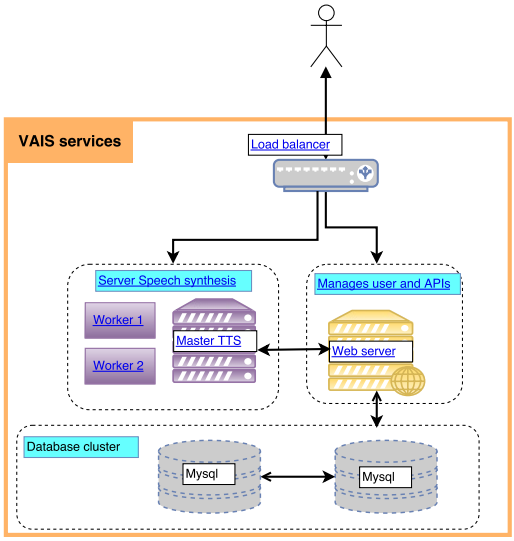



Figure 1. VAIS-TTS system overview

## III. Technologies

## IV. Speech concatenation approach

Speech concatenation [1] is an approach that synthesize audios by concatenate speech segments from a database. The idea is as follows, first, audio and text are aligned at the phoneme level to provide information about where the phoneme is located in the speech sound. Second, the text is also analyze to provide linguistic information, such as phonetic context, part-of-speech tags, word position. Finally, a database is built which contains all speech segments and phonetic information.

At the synthesis time, the speech segment is chosen to minimize the combination of the target cost which measured by a heuristic distance between contexts and the concatenation cost which measured by speech parameter distortion (Fig. 2).

Because the audio is synthesized using the speech segment extracted directly from original audios, it provides very high quality speech waveform signal. If the model can choose correctly speech segments, it is very difficult for human to distinguish the synthetic voice and the natural voice. However, allthough the concatenation approach can produce high quality speech waveform, it comes with some disadvantages. First, it

has large footprints because all speech segments are stored in the model. Our actual model size is approximately 1GB when trained on 6k utterances. Second, the sound sometime has unstable quality due to wrong alignment and mistake during segment selection.

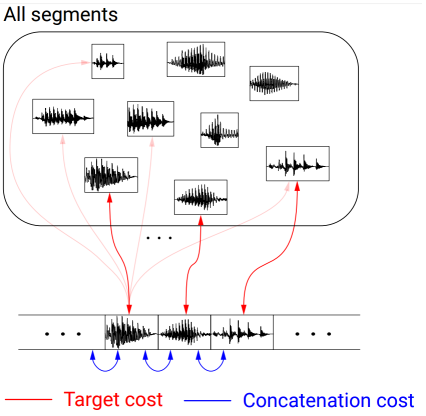



Figure 2. Speech concatenation procedure.

## V. HMMs approach

The HMM approach models speech by using Gaussian mixture models. Each phoneme is modeled by an HMM instead of many speech segments in concatenation approach. At the synthesis time, a sentence HMM is constructed from the given text. Then the speech parameter is predicted to maximize the likelihood probability,

$$\hat{\boldsymbol{O}} = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T), \tag{1}$$

where $\boldsymbol{O}$ is speech parameters, $\lambda$ is the model parameters, and $T$ is the length of speech that we want to generated.

Unlike unit concatenation approach where we need to collect large amount of speech data to have good quality, the HMM approach can trained a model with just few hundreds phonetic-balanced utterances. This allows us to quickly train a fairly good model given small amount of data.

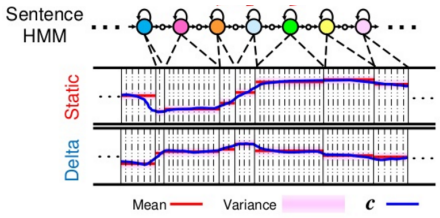



Figure 3. HMM-based speech synthesis

The HMM voice has very low footprints, in fact, our model trained on 6k utterances takes only 5MB of storage and 10MB of memory when fully loaded. The approach can also provide very flexible voices by changing model parameters [2], and also be able to adapted to someone else voices with minimal data collection required[3].

## VI. Front-end text processing

Vietnamese is a complex language where one word can be pronounced in different ways depending on the context, for example, the number "1" in the word "21" is pronounced as "hai mươi mốt" , however, if it is in the word "11", the correct pronunciation should be "mười một" . Another problem is abbreviations that is very often being used in newspaper, such as TPHCM, VKSND. The list of abbreviation is endless and have no rules to pronounced.

To make the speech synthesis more useful for general tasks, we define a set of regression rules for date, numbers, date-of-birth, time, units (such as currencies, temperature, weights), and develop a toolkit for define abbreviation words. The processing procedure is illustrated in Fig. 4.

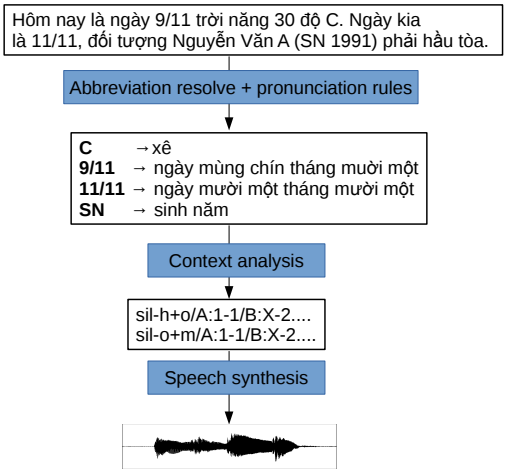



Figure 4. The text analysis front-end procedure.

## VII. Service integration

To create an easy way for all people can access to our service, we create various end-points connection, SDKs, and also applications described as follows,

### A. API end-points

- **HTTP**: The HTTP connection provides a simple, easy to use interface.
- **Websocket**: The HTTP api end-point is simple, however, it slows down the processing request due to hand-shaking procedure. To address the problem, we provide websocket connection where it only requires 1 time opening connection. All requests after the first one will go directly to the synthesis engine. Therefore, it is suitable for real-time required applications.

### B. SDK tookits

*1) Android SDK:* To make things even easier for developers, we provide an Android SDK toolkit that can perform not only online speech synthesis, but also have offline speech synthesis engine built-in. If a valid API key and the Internet connection are provided, the high quality speech engine will be

used. When any of the above conditions is not met, it fallbacks to a local speech synthesis engine used HMMs models. It is guaranteed that the speech synthesis service operates all the time, even when there is no Internet connection.

## VIII. Conclusion

In this paper, we briefly describe our first public high quality speech synthesis system for Vietnamese language. The system is easy to deploy, easy to scale, and high availability. Two accents for Northern and Southern are available, we are expecting to have more voices in near futures.

## References

[1] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *In Proceeding of ICASSP*, 1996, pp. 373–376.

[2] N. Takashi, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based expressive speech synthesis," *IEICE*, vol. 90, no. 9, pp. 1406–1413, 2007.

[3] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE*, vol. 90, no. 2, pp. 533–543, 2007.

# Instrumenations controlled by Vietnamese voice

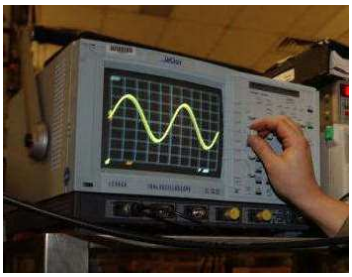
*Dr. Viet Son Nguyen, Dr Viet Tung Nguyen*
*Dr. Thi Ngoc Diep Do, Dr. Dang Khoa Mac, Dr Trung Kien Dao*

**International Research Institute MICA**
**Multimedia, Information, Communication & Applications**
**UMI 2954**

Hanoi University of Science and Technology
1 Dai Co Viet - Hanoi - Vietnam

## OBJECTIVES

- **Human vs. Osc./FG interaction**
  - Adjust parameters by hand
  - Observe results on small screen
- **But**
  - Hands of technician are used to keep probes
  - Technician has to move her/his hand to
    - Keep probes
    - Adjust parameters of machine



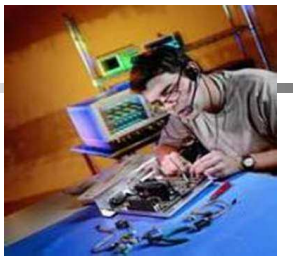
→ **Difficult:**
  - Keep in touch the measuring point
  - Focus on measurement

→ **Not efficiency in manipulation**

MICA 2016

2

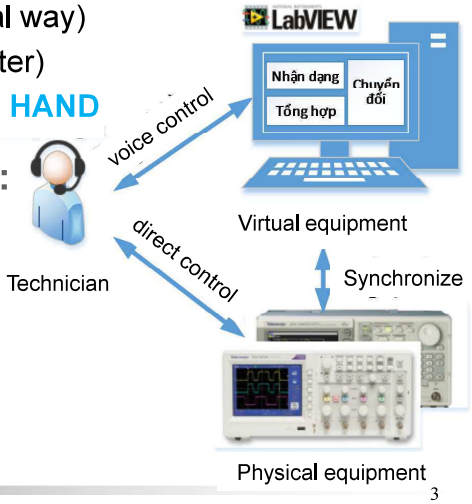
# OBJECTIVES



- **Integrate HMI by VN voice system**
  - Keep all orginal functions of Osc./FG
  - **03 ways to control**:
    - Physical buttons (traditional way)
    - **Mouse, keyboard** (computer)
    - **Voice command** → **FREE HAND**
- **Synchronize observations:**
  - Screen
  - Indicators
- **Save data to computer**

→ **Easier observation**

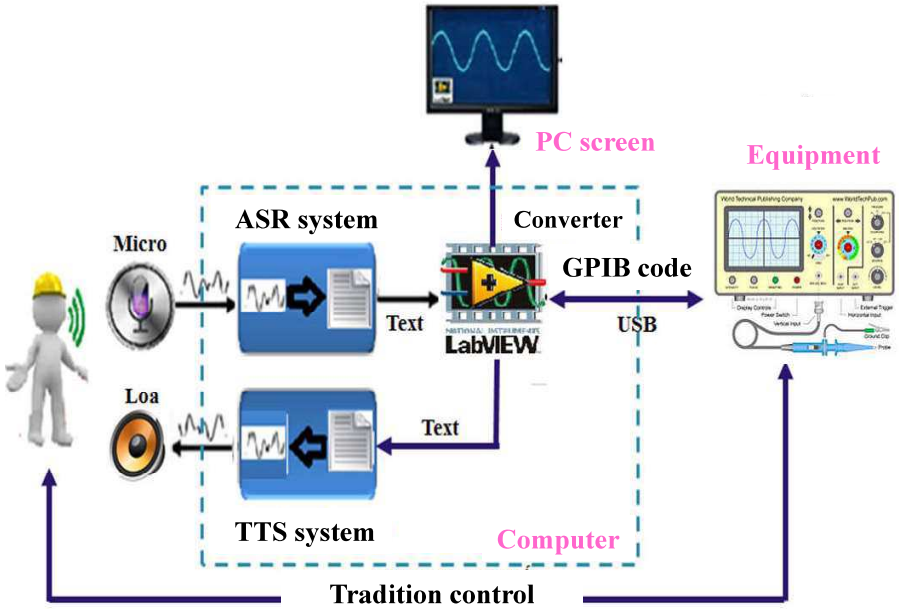
→ **Easier manipulation**




MICA 2016

3

# Infrastructure of HMI by voice system

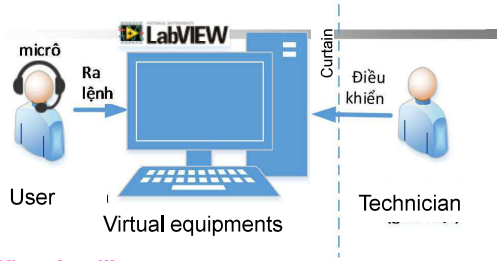



MICA 2016

4

# HMI by voice system

- **Voice command sets:**
  - Natural
  - Regular
- **Approach:**
  - Wizard of Oz
- **Results:**
  - Voice command
  - Voice query
  - Function Generator: 65
  - Oscillator: 87




Virtual oscillator

Virtual F.G.

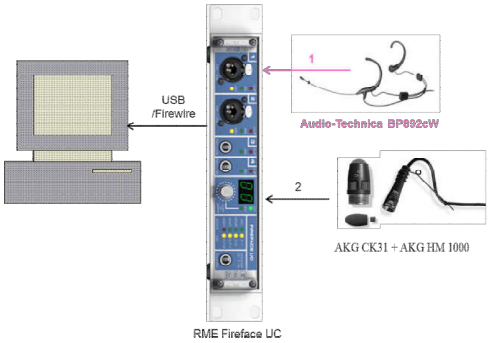

**Regular voice command sets**

| Function | Regular 1 | | Regular 2 | |
|---|---|---|---|---|
| Bật kênh tín hiệu | Bật kênh X | (65%) | Phát kênh X | (30%) |
| Tắt kênh tín hiệu | Tắt kênh X | (90%) | Kênh X tắt | (8%) |
| Chọn dạng sóng | Sóng Y | (70%) | Phát sóng hình Y | (20%) |

MICA 2016

# Training corpus for ASR system

- **Speech corpus:**
  - 02 synchronized channels:
    - 16bits
    - 16kHz
  - Recording in real environment:
    - Electronic/Electrical lab.
    - Noise
  - Dialects:
    - North: 20 subjects (10M + 10F)
    - Center: 20 subjects (10M + 10F)
  - 70.000 utterances (~40 hours)




MICA 2016

## Evaluate HMI system: 03 criteria

- **Accuracy of ASR system**
  - Independence speaker
  - 20 subjects:
    - 10 from the North, 10 from the Center
    - 10 Males + 10 Females
  - 03 times / subject
  - Noisy environment: Electronic/Electrical lab.
- **Quality of TTS system**
  - Use MOS
  - 25 subjects
  - Environment: Sound-proof room
- **Performance of HMI system**
  - Response time

MICA 2016

## Evaluate HMI system: Results

- **Accuracy of ASR system**

| Accuracy | Function Generator | Oscillator | Average |
|---|---|---|---|
| **First time 1** | 97.38% | 97.14% | **97.24%** |
| **Repetition 2** | 99.44% | 99.40% | **99.42%** |
| **Repetition 3** | 99.80% | 99.73% | **99.76%** |

- **Performance of HMI system**

| Response time | Function Generator | Oscillator | Average |
|---|---|---|---|
| **Average** | 1.30s | 1.36s | **1.33s** |
| **Average for all commands shorter than 5 syllables** | 1.24s | 1.27s | **1.26s** |

- **Quality of TTS system: 4,75 / 5**

MICA 2016

8

## Instrumentation control by Vietnamese voice

- **Features:**
  - Control by 03 ways
    - Using buttons on machine interface
    - PC (mouse, keyboard)
    - Voice commands
  - Synchronized observation
    - Screen
    - Indicator
  - Voice command sets:
    - Function Generator: 65
    - Oscillator: 87
  - HMI by voice:
    - North & Center voice
    - Accuracy: ~97%
    - Time response: < 2s
    - Answer by VN-TTS
  - Data save on PC

MICA 2016

## Thanks for your attention !

MICA 2016