P-ISSN 2586-9000
E-ISSN 2586-9027

Homepage : *https://tci-thaijo.org/index.php/SciTechAsia* **Science & Technology Asia**

Vol.26 No.1 January - March 2021 Page: [64-83]

Original research article

Information Extraction for Thai Celebrities from Free Text

Jian Qu*, Chinorot Wangtragulsang

*Faculty of Engineering and Technology, Panyapiwat Institute of Management,
Nonthaburi 11120, Thailand*

Received 31 May 2019; Received in revised form 10 July 2020
Accepted 11 August 2020; Available online 16 March 2021

ABSTRACT

Automatic extraction of Thai-language information still has challenges because of language structure, lack of word segmentation, presence of vowel and intonation, and specific words that are not in a dictionary. Challenges encountered in Thai-language personal information extraction are low candidate recall and candidate ambiguity. This work proposes an automatic personal information extraction approach capable of extracting date of birth, height, heritage, Instagram, Twitter and film names of Thai celebrities from 22,484 Thai-language webpage snippets using novel pattern matching, feature selection and machine learning methods to select the most likely piece of information out of a number of possible candidates. We compare performances of our method with a large, actively maintained website like MThai.com that contains some personal information. In this case, performance of Mthai.com is up to 70% in recall and precision. Further comparison is done with state-of-the-art works on automatic Thai information extraction that used tokenizer and rules-based extraction, which could perform at only 40-50% in terms of recall and precision. According to experiments, our approach can extract date of birth, height, Instagram, and Twitter with recall and precision being between 70-90%. Furthermore, we can extract some heritage and film names where existing methods cannot.

Keywords: Information extraction; Named entity; Personal information extraction; Social media; Unstructured data

1. Introduction

Personal information extraction pertaining to Thai celebrities, such as actors, actresses and singers is generally done by hand either by the web administrator or blog owner. For example, in Thailand there is MThai.com which is currently managed by the MONO Group, a large public company limited with annual revenue of over two billion baht. The website ranked the 13th

***Corresponding author:** Jianqu@pim.ac.th

doi: 10.14456/scitechasia.2021.7

most visited website in Thailand according to Alexa [1]. Some other websites about celebrities in Thailand include Parwat.com and Thaiza.com that are managed by a staff. Some websites such as the Internet Movie Database (IMDb) use crowdsourcing that allow other people, including the celebrities themselves, to add information. This signifies how expensive information extraction could be, especially when a large number of source information is needed.

As the amount of information and celebrities grow, keeping a celebrity's information updated becomes increasingly difficult due to poor scalability of manual methods and/or inadequate crowdsourcing. This causes incompleteness, i.e. some information is present in one website but not the other. One example is Thanakorn Posayanon on MThai.com. Although Mr. Possayanon had debuted in 1995, MThai.com does not have information on him before 2016. Even on Channel 3's website, which he is affiliated with, there are only eight series attributed to him. On the other hand, Thairath.com, a major newspaper website in Thailand (Ranks #16 according to Alexa [2]) lists Mr. Possayanon's work from 1995 up until 2012. It is notable that all three websites do not have complete film information in comparison with the Thai version of Wikipedia. Although Wikipedia has extensive information on Mr. Possayon's filmography, there is no information on his height, heritage and social network channel. Moreover, the number of Thai celebrities with Wikipedia entries remains low compared to MThai.

In summary, information on Mr. Possayanon which can or cannot be obtained from MThai, Channel 3, Thairath and Wikipedia is shown in Table 1. This shows that information incompleteness across websites is common.

Another major problem is ambiguity of the information. For example, Praiya Suandorkmai (born Nataya Lundberg) was reported in some websites as having Arabic heritage while she is actually half-Swedish. Another is the famous Nadech Kugimiya who is a half-Austrian, but due to his adopted surname, was reported by some websites as half-Japanese until his clarification in interviews later on. Some information such as height can be ambiguous, such as in case of Thanya-Thanyaret Ramnarong, as some websites state she is 160cm in height, while other state she is actually 162 cm. This ambiguity problem presents a major challenge for a simple pattern-matching approach.

Table 1. Thanakorn Possayanon's available.

Information	MThai	Channel 3	Thairath	Wikipedia
Birthday	Yes	Yes	Yes	Yes
Height	No	Yes	Yes	No
Heritage	No	No	No	Yes
Social media contact	Yes	Yes	No	No
Film	Yes (2016 onwards)	Yes (Incomplete)	Yes (up to 2012)	Yes (Complete)

Due to high human expense, incompleteness and ambiguity related with personal information extraction as described, this study aims to introduce a novel approach to tackle these problems. Regarding human expense in extraction, we introduce an automated information extraction, automated candidate generation and automated candidate selection method. Incompleteness is handled by using automatic web crawling to get as much raw information from the Internet as possible. Finally, to handle ambiguity problems, a novel pattern matching method, and candidate selection method is proposed. As we attempt to obtain as much information as possible, there are noises in the raw information, therefore a keyword method is introduced to reduce noise.

The paper is organized as follows: Section 2 is an extensive review of existing methods, Section 3 presents the research design and describes the research method in detail, Section 4 includes results of our experiment, comparison with existing methods, and discussion, and Section 5 is the conclusion.

2. Literature Review

In this section similar information extraction works will be summarized in the overview sub-section. Then, each reviewed work's process will be compared with this work in three subsections: web crawling, candidate boundary detection, and candidate selection.

2.1 Overview of similar(related) works

Information extraction works can be divided into two groups: extraction from structured texts, and extraction from unstructured texts.

Works involved with structured texts include work by Noordin and Othman [3] which proposed an information retrieval system for Quranic texts from a set of websites (125 in total). Kopparapu [4] developed a system to extract personal information from "free-structured" resumes using various Natural Language Processing techniques such as rules and machine learning (the latter used to create a knowledge base). Chen et al. [5] developed an information extraction system for resume documents in PDF using heuristic rules and Conditional Random Format. Chen et al. [6] made a two-step resume information extraction system with a novel feature. In this work the structured resume was extracted, filtered and classified according to writing style.

The next group of reviewed works is about information extraction from unstructured text. This group uses various methods, one being lexicon-based named entity extraction. In this method, words or sentences are compared with a lexicon of named entities to find matching pairs. This approach is popular in the healthcare domain due to wide availability of medical lexicons. One example is AZDrugminer developed by Liu et al. [7] that extracted adverse drug-events from online social media sites such as DailyStrength and PatientsLikeMe. Outside of the healthcare domain, Elsebai et al. [8] developed a name entity extraction system from Arabic text.

The next group of works extracts information from unstructured text. Sharma and De Choudhury [9] developed a computational method to extract nutritional information from Instagram posts. The method uses a manually-compiled list of food-related words to extract food names and calorific information from Instagram posts. Imran et al. [10] used conditional random fields and ARKNLP (a powerful tokenizer developed specially for Twitter) to obtain information about natural disasters from Twitter posts. Chen et al. [11] created a robust web personal name information extraction system that used existing preprocessing programs Beautiful Soup (an HTML parser) and MXTERMINATOR to find sentence boundaries. Then, the system tested rules- based methods, machine learning methods and hybrid methods on unstructured information. Cheng et al. [12] developed Legal TRUTHS (TuRning Unstructured Texts to Helpful Structure), using Balie, a tokenizer and named entity recognizer, to extract information like case number, penalty, pay, moral damage, and order. Freitag [13] used a machine learning approach to extract information from seminar announcements and corporate acquisitions. Androutsopoulos et al. [14] used machine learning for spam email filtering.

The aforementioned works involve English-language texts, but there are similar works on non-English texts as well.

Examples of information extraction studies in other languages are those by Emami et al. [15] which used pattern-matching to obtain personal information from websites in Farsi language. Aside from Farsi, Arabic-language information extraction was done by Aboaoga and Ab Aziz [16], which focused on extraction of Arabic personal names. Qu and Lu [17] developed an approach to automatically translate Chinese out-of-vocabulary words into English by extracting candidates and using pattern-matching and association rules.

There are examples of Thai language information extraction approaches such as Chainapaporn and Netisopakul [18] that developed an approach to extract Thai herb information from multiple websites. This approach went directly to random websites, extracted content in the HTML files and then used regular expressions to extract herb name and effects. Wanichayapong et al. [19] extracted traffic information from Twitter. This approach had a list of roads in Bangkok and a dictionary of words describing traffic conditions.

In-depth examination of the aforementioned information extraction works along with some other examples will be done in the next section grouped by each step of the entire process: 1) web crawling, 2) candidate boundary detection, and 3) candidate selection.

2.2 Web crawling

Web crawling is automatic extraction of web pages, or text thereof, from the Internet. The program used in such crawling can be called a "Spider" or "Spiderbot". The goal is to index or collect information from the Internet or specific websites using keywords to limit the scope of desired information.

There are many methods that deal with web crawling. Papers reviewed in this part are grouped according to the type of texts they worked on. The first group extracts information from structured text such as resumes or medical records. The second group extracts information from unstructured text such as articles and social media posts on the Internet.

2.2.1 Web crawling in works related to structured text

Chen et al. [5] used a web crawler to obtain resume information using Google Search Engine. Non-English resumes or resumes that could not be properly parsed were simply discarded. Then, the resumes were manually annotated.

Chen et al. [6] drew 15,000 resumes from Kanzhun, the biggest company review website in China much like Glassdoor in the rest of the world. Apache Tika was used to parse words from the Word resumes, and JSoup was used to extract words from HTML resumes.

2.2.2 Web crawling in works related to unstructured text

Works in the second group include work by Chainapaporn and Netisopakul [18] randomly selected 400 webpages from 80 Thai herb-related websites by hand. Then, the JSoup API was used to extract HTML source from each web page.

Gossen et al. [20] proposed iCrawl, a system that could extract information from Twitter and other social media. In this work, Apache Nutch, an open-source web crawler was used. This crawler required a "Seed" URL for it to crawl the page and parse information into a database.

Sharma and De Choudhury [9] crawled Instagram posts by manually obtaining a list of 564 words from an online food vocabulary word list. In addition, 24 more words which were likely to be in food posts were added to the list. This word list was used to seed tags in order to get more English-language public posts from Instagram.

Most of the reviewed works that involve extracting information from Twitter use hashtags or keywords to obtain desired tweets for further processing. Imran et al. [10] crawled tweets about natural disasters from Twitter using the Twitter API. Tweets with certain hashtags were retrieved and stored in a database. Wanichayapong et al. [19] likewise used Twitter search API. In this work searching was set to be done automatically every 5 minutes, and 20 new tweets could be extracted each time.

Qu and Lu [17] used Microsoft Bing search API to get English out-of-vocabulary words from English-language web pages, Bing's search results are not ranked as highly as Google but Bing was offering a better free search environment than Google back in 2015.

2.3 Candidate boundary detection

Candidate boundary detection is done in order to get possible candidates from the text. Methods used are pattern matching, brute force and machine learning. These three methods are explained below.

2.3.1 Pattern matching

In the pattern matching approach, keywords within the text are located and then a pattern of words that match one of the predefined rules is extracted as a possible candidate.

Kopparapu [4] used Natural Language Processing techniques such as rules and machine learning (the latter is used to create a knowledge base). This work assumed that all information about one person was stated in a single document and was somewhat structured. So, the system in this paper would look for keywords related to qualifications, skills, and names and then extract words to be candidates for further evaluation. This approach has the advantage of high accuracy, while the disadvantage is possibility of low recall if this method is used on less-structured text.

Another similar work is Liu's AZDrugminer [7], which uses lexicons of medical terms to extract medical entities from the text. This dependence on existing lexicons means if there is no existing infrastructure available, the lexicon must be generated anew.

Aboaoga and Ab Aziz [16] used two lists of keywords: Introductory Word List and Introductory Verb List as keywords to extract personal names from Arabic texts. This concept could be applied into other languages such as Thai, because Thai texts mentioning personal names would at least have some introductory words. However, it could not be used for other information like birthday, social media, or films with the same effectiveness.

Sharma and De Choudhury [9] developed a computational method to extract nutritional information from Instagram posts. The method uses a manually compiled list of food-related words to extract food names and calorific information from Instagram posts. Then, human raters are selected to validate the nutritional information. This specific work is limited to Roman alphabet- based texts due to the list of words used, and if used as- is for the more ambiguous Thai- language text, recall and precision would inevitably suffer.

Chainapaporn and Netisopakul [18] used the JSoup API to look for tags with herb information in them. For more complex webpages, some indicator words are used to get information. After that, Lexito, a word separator program for Thai language, was used to split the text into words. Then, indicator words like treats, remedies, and assuages were located and words next to them were extracted as candidates or "learn" words as this work mentioned.

Emami et al. [8] used pattern-matching to extract attributes of named entities. This work did not use machine learning, and its ability to extract information like affiliation, relatives, and occupation was limited due to huge variations of how such information was written in Farsi.

Wanichayawong et al. [19] built a dictionary consisting of 46,241 names of roads, places, crossroads, and alleys (Soi in Thai), and another dictionary consisting of 1,093 words that describe traffic conditions. After crawling the Tweets, the tweets would be checked by a tokenizer called Lexto. Then, the tokenized tweets would be checked with the dictionary to find traffic conditions, starting point and ending point.

2.3.2 Brute force

In contrast with pattern matching, brute force has much greater recall because some languages might result in low recall for pattern matching. Qu and Lu [17] resorted to a brute force approach to extract Chinese translation of English out-of-vocabulary words. They generate the candidates by cutting the Chinese sentence into characters, and join the characters one by one to form many possible candidates; although this method has a good recall, it generates many noise (wrong candidates) and usually one sentence will produce 20-50 possible candidates. This work used a combined approach. First, pattern matching was used to extract translations based on an assumption that correct translations should be enclosed in brackets and near the original Chinese word. If pattern matching did not yield any result, brute force candidate generation was used instead.

2.3.3 Machine learning-assisted candidate boundary detection

Research works in this group use machine learning to help in candidate boundary detection. Imran et al. [10] used conditional random fields and ARKNLP (a powerful tokenizer developed specially for Twitter) to extract words about natural disasters from Twitter posts. As Twitter posts used in the study were in English and a dedicated tokenizer was used, this approach might be less useful for posts in other languages such as Thai.

2.4 Candidate selection

Candidate selection is to pick the correct candidate from a list of possible candidates generated from the previous step. Literature review revealed that there are generally two groups of methods for candidate selection for NE tasks. One is a rules-based approach where the statistical rules are applied to select the possible correct candidates. The second method is machine-learning where features of each candidate are extracted, and then applied to traditional machine learning method to select the possible correct candidates. We will explain these methods in detail as follows.

2.4.1 Rules-based candidate selection

In rules-based candidate selection, candidates are compared with the correct answer that has been manually verified.

Chainapaporn and Netisopakul [18] was another that used pattern matching to extract information like common names of Thai herbs from candidates. Furthermore, it used “part-of-used” to extract the name of the Thai herbs’ parts (such as root, leaf, or trunk) that are used for treatment along with their benefits.

Chen et al. [6] used heuristic rules to extract information candidates from raw resume text (after filtering out non-text parts) such as a rule to get attribute-value pair, part-of-date, block keywords, and comma.

Aboaoga and Ab Aziz [16] used a dictionary of personal names to see whether the extracted candidates were in the dictionary, and if the candidates did not match the dictionary, further rules would be applied.

Cheng [12] used keywords such as “penalty”, “order”, “sentence” to select information related with such keywords in legal documents. Due to stringent writing standards of legal documents (more efforts into disambiguation of words), a rules-based method works favorably.

2.4.2 Machine learning-based candidate selection

This group of works use machine learning to select the likely information out of a list of possible candidates.

Freitag [13] used a Hidden Markov Model (HMM) and Shrinkage to extract purchasing price information. In this work, each HMM would extract one type of information from the documents, so if one document has many types of information it would have many different HMMs working on it. It is notable that there is no pre-processing for the document: the entire document is modeled. After that the HMMs were shrunk to improve performance.

Androutsopoulos et al. [14] compared Naive Bayesian classification and memory-based (a variant of k- Nearest Neighbor) classification for a system that selects legitimate emails from spam mails. It was found that these methods gave much higher precision than widely-used anti-spam methods.

2.5 Notable machine learning algorithms

Given the potential use of machine learning in information extraction, we can use some traditional ML methods for candidate selection if given some statistical candidate features, such as candidate frequency and candidate-keyword co-occurrence frequency, etc. Some common machine learning algorithms such as Decision Tree, Random Forest, Support Vector Machine, k- Nearest Neighbor, and Artificial Neural Network will be explained below.

Decision Tree learning is quite simple to implement, needing little data preparation and is easy to understand. Furthermore, decision trees can be used for both classification and regression. However, the risk of overfitting (overly-complex trees with excessive splits) is large and thus some measures such as pruning are needed to prevent this. Pruning is a process of reducing tree complexity by building a tree just big enough that the residual sum of squares (RSS) stops decreasing.

Random Forests is a more generalized, recent multiple tree method. The first algorithm was created by Ho in 1995 using the random subspace method. According to Ho, Decision Tree was, despite its strength, restrictive and was at risk of overfitting. The idea of oblique decision trees was a more general approach, as hyperplanes are not always parallel to any of the axes and could give a smaller tree (thus less overfitting). Two methods for tree growing were examined, the first was finding splitting hyperplanes that are perpendicular to a line connecting two data clusters together with the goal to divide at least two classes at each non- terminal node in one pass, creating a central axis projection. The second method used the fixed- increment perceptron training to choose the hyperplane at each non-terminal mode. Both methods could be used to grow complex trees that could perfectly classify the training data but bias could result in poor generalization, so Ho proposed to create multiple trees in randomly selected subspaces. It was later extended by Leo Breiman and Adele Cutler who registered "Random Forests" as a trademark.

A Support Vector Machine (SVM) is another algorithm that tries to find a hyperplane in a space that clearly separates two groups of data. It was developed in the 1960s by Vapnik and Chervonenkis. It is a generalization of a classifier named the maximal margin classifier. Although simple and intuitive, the maximal margin classifier cannot be applied to datasets which cannot be separated by a linear boundary. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. It is claimed by Gareth et al. as one of the best "out-of-the-box" classifiers.

The next algorithm is k- Nearest Neighbor (k- NN). k- NN is based on an assumption that similar things will stay close to each other. This algorithm directly uses the text as a model instead of a training example. Like the other algorithms described, k- NN can be used for either classification or regression.

The k-NN process starts from loading the data and initializing a "K" value to a data group of choice. In this step, the K value is arbitrary: usually derived from manual trial-and- error runs to find the best value. However, the lower the K-value gets, the less stable prediction will be, while the higher the K-value (preferably odd number breaks a tie) means majority rules can be used.

After getting the K- value, the next steps are calculating distance between the query example and current example, adding the example' s distance and index to a collection, sorting the collection by distance, selecting the first K entries from the collection, and obtaining labels of the selected entries. Results from this process depend on whether regression or classification is being done; the former will return the mean of the K-labels, and the latter will return the mode of the K-labels. The k-NN has simplicity and flexibility as the strength, at the cost of speed.

The last algorithm covered in this work is the Artificial Neural Network or ANN. The ANN is basically an imitation of a biological neural network in machine learning. In this case, there are artificial "neurons" that are interconnected much like in the human brain. These neurons work together to produce a result. If there is only one layer of interconnected neurons or neural network, it is called a perceptron that produces a single result. However, there can be multiple neural network layers to handle complex problems. This method does not need prior programming, but learns from an example set. Errors between the predicted value and actual value (called Cost Function) are fed back to the network.

In the next part, additional machine learning methods to boost performance will be explained.

2.6 Feature selection

Feature selection is an attempt to find the best combination of features. According to Guyon and Elisseeff [21] the number of data features was not a problem until recently, when hundreds of thousands of features are frequently explored and many of them might be redundant or irrelevant. In this case, selection of only necessary features will benefit the machine learning process in many ways. James et al. [22] stated that feature selection can be used to simplify the machine learning process, shorten the training time, and reduce overfitting. Common methods for feature selection, including backward elimination will be explained below.

Brute force feature selection is the most direct approach that will get the best result, but as seen in its application elsewhere, this method needs a lot of memory [23-24].

There are many strategies to select features. Some other methods include variable elimination, which itself can be divided into filter and wrapper methods. In this case, the filter method produces a ranking of features, with the high-ranking features selected for use. In the wrapper method, a search algorithm is used to find the best-performing feature subset. There is another method called the embedded method which combines the strength of both the filter and wrapper methods.

One commonly used method, backward elimination, can be used to select features. It works by taking all the features and generating multiple sets of the same features, each with one feature omitted based on statistical calculation. Then, the one with the possibly best performance will be used as the initial set of features for the next round of elimination until performance stops improving.

This study aims to extract information from unstructured, Thai-language text such as webpage snippets and develop an automated information extraction that extracts personal information such as birthday, height, heritage, filmography, and social media contacts of the Thai celebrities from Thai-language web articles.

3. Research Method

This study aims to retrieve the celebrity's personal information such as date of birth, height, social media contacts and film names from unstructured, Thai-language webpage snippets from Google, using pattern matching and machine learning. Our experiment uses Thai celebrities listed on MThai.com/starthai. Our approach is compared against the MThai website and the pattern matching and tokenization used by Chainapaporn and Netisopakul [18] in terms of accuracy.

The process of celebrity information extraction consists of four steps. The first step is web crawling, where snippets are gathered and stored in a database. In this step, the input is the name of celebrity which is used to crawl snippets from the Internet. The second step is rules-based pattern matching, where two novel rules-based methods are proposed: the first method is to get date of birth, height, and heritage candidates and the second method is to get Instagram and Twitter candidates. The third step is feature extraction. Here, we extract features from the candidate. After that, in the fourth step, the backward elimination method is used to optimize the base machine learning algorithm. Then, there are five base machine learning algorithms (Decision Tree, Random Forest, Support Vector Machine, Artificial Neural Network, and k-Nearest Neighbor) that we apply to select the correct candidate, which would be the final output such as birth, height, heritage, Instagram and Twitter. On the other hand, preliminary tests show that film names have different characteristics than other personal information, and thus a novel F-filter-based method is used to reduce disparity in numbers between correct and incorrect candidates. A flowchart of the entire process is shown in Fig. 1.

3.1 Web crawling

The Google search API is implemented to obtain information on Thai celebrities in the form of webpage snippets. We use celebrity names as inputs to obtain the snippets, and snippet count for each celebrity is limited to 100 to ensure maximum coverage with acceptable noise as mentioned in Qu et al. [25]. Due to limitation of free Google API, we are able to only retrieve 100 snippets per 24 hours. Fig. 2 shows what a typical snippet looks like.

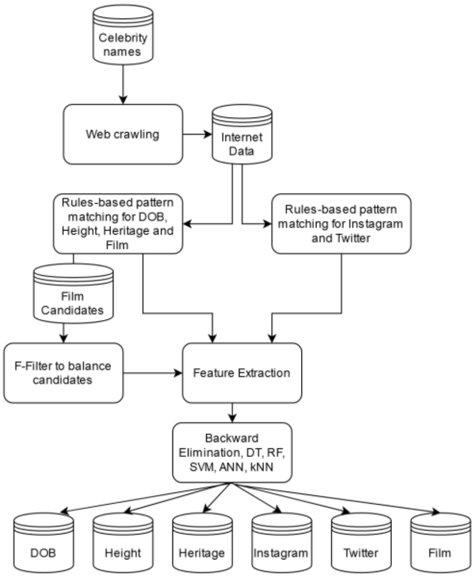
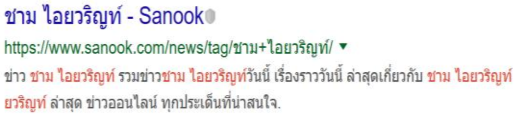

Celebrity names
Web crawling
Internet Data
Rules-based pattern matching for DOB, Height, Heritage and Film
Rules-based pattern matching for Instagram and Twitter
Film Candidates
F-Filter to balance candidates
Feature Extraction
Backward Elimination, DT, RF, SVM, ANN, kNN
DOB
Height
Heritage
Instagram
Twitter
Film

Fig. 1. Flowchart of the information extraction process.



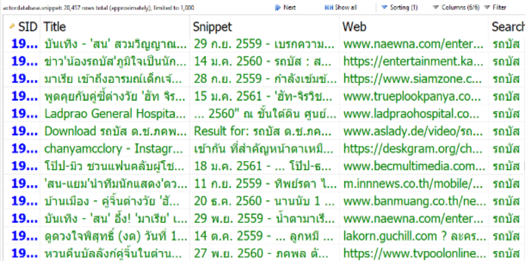
ชาม ไอยวริญท์ - Sanook

https://www.sanook.com/news/tag/ชาม+ไอยวริญท์/ ▾

ข่าว ชาม ไอยวริญท์ รวมข่าวชาม ไอยวริญท์วันนี้ เรื่องราววันนี้ ล่าสุดเกี่ยวกับ ชาม ไอยวริญท์ ยวริญท์ ล่าสุด ข่าวออนไลน์ ทุกประเด็นที่น่าสนใจ.

Fig. 2. Example of a snippet.

As shown in Fig. 2, a snippet contains a title, a web URL, and part of the webpage. The snippet is then inserted in a table shown in Fig. 3.



actordatabase.snippet: 20,457 rows total (approximately), limited to 1,000 ▶ Next ⏭ Show all ▾ Sorting (1) ▾ Columns (6/6) ▾ Filter

SID	Title	Snippet	Web	Searcl
19...	บันเทิง - 'สน' สวมวิญญาณ...	29 ก.ย. 2559 - เบรกความ...	www.naewna.com/enter...	รถบัส
19...	ข่าว'น้องรถบัส'ภูมิใจเป็นนัก...	14 ม.ค. 2560 - รถบัส : ส...	https://entertainment.ka...	รถบัส
19...	มาเรีย เข้าถึงอารมณ์เด็กเจ้...	28 ก.ย. 2559 - กำลังเข้มข้...	https://www.siamzone.c...	รถบัส
19...	พูดคุยกับคู่ซี้ต่างวัย 'ฮัท จิร...	15 ม.ค. 2561 - 'ฮัท-จิรวิช...	www.trueplookpanya.co...	รถบัส
19...	Ladprao General Hospita...	... 2560" ณ ชั้นใต้ดิน ศูนย์...	www.ladpraohospital.co...	รถบัส
19...	Download รถบัส ด.ช.ภคพ...	Result for: รถบัส ด.ช.ภค...	www.aslady.de/video/รถ...	รถบัส
19...	chanyamcclory - Instagr...	เข้ากัน ที่สำคัญหน้าตาเหมื...	https://deskgram.org/ch...	รถบัส
19...	โป๊ป-มิว ชวนแฟนคลับผู้โช...	18 ม.ค. 2561 - ... โป๊ป-ธ...	www.becmultimedia.com...	รถบัส
19...	'สน-แยม'นำทีมนักแสดง'ดว...	11 ก.ย. 2559 - ทิพย์รดา ไ...	m.innnews.co.th/mobile/...	รถบัส
19...	บ้านเมือง - คู่จิ้นต่างวัย 'ฮั...	20 ธ.ค. 2560 - นานนับ 1 ...	www.banmuang.co.th/ne...	รถบัส
19...	บันเทิง - 'สน' อึ้ง! 'มาเรีย' เ...	29 พ.ย. 2559 - น้ำตามาเรี...	www.naewna.com/enter...	รถบัส
19...	ดูดวงใจพิสุทธิ์ (งด) วันที่ 1...	14 ต.ค. 2559 - ... ลูกหมี ...	lakorn.guchill.com ? ละคร...	รถบัส
19...	หวนคืนบัลลังก์คู่จิ้นในตำน...	27 พ.ย. 2560 - ภคพล ตั...	https://www.tvpoolonline...	รถบัส

Fig. 3. Part of the snippet table.

Aside from the title, URL and snippet, the table includes search terms, in this case celebrity name, which is given an ID.

3.2 Rules-based pattern matching

A set of regular expressions is used to extract possible pieces of information or "candidates" from the snippets.

Manually-written rules are used to extract date of birth, height, heritage and film candidates. The pseudo-code of our approach is shown in Fig. 4.

```
Input: Snippet S, Name of the celebrity NE, keywords Key,
Keyword list [BKey, HKey, HeriKey, Filmkey] ,Regular
Expression RegEx, The number of regular expressions a
Output: Date of Birth DOB, Height Height, Heritage Heritage,
Film Filmname
For each S do
    Search NE, Key in S
If (Key and NE found in S) Then
Find BKey, HKey, Herikey, Filmkey in S with RegEx[a]
Return DOB, Height, Heritage, Filmname
Else
      Return "NA"
End
  End
End
```

Fig. 4. Pseudo code of the birthday, height, heritage, and film candidate extraction.

As shown in Fig. 4, the pattern-matching method picks a snippet S, then searches the name entity and keyword Key inside S. Once both are found, the method locates the birthday keyword and then extracts information that has a pattern perfectly matching with one of the rules.

The rules are strict at first and then gradually loosen in later iterations. If there is still no result after the last (and the greediest) rule, the result will be "NA", the pattern-matching program then moves to the next snippet.

The name of the celebrity is used as a keyword to ensure correct attribution, and keyword (Key) to locate relevant information. Possible keywords are "วันเกิด" or "เกิดวันที่" and "พ.ศ." (Buddhist Era).

```
Input: Title T, Snippet S, Web URL URL, Name of the celebrity
NE, keywords Key, Keyword list[IKey, TKey,],Regular
Expression RegEx, The number of regular expressions a
Output: Name of Instagram account Instagram, Name of
Twitter account Twitter
For each T,S, URL do
    Search NE, Key in T,S
              If (Key and NE found in T,S) Then
Find Instagram, IKey, TKey, in T,S,URL with RegEx[a]
    If Instagram or Twitter is found
               Return Instagram or Twitter
              Else
                     Return "NA"
              End
  End
End
```

Fig. 5. Pseudo code for Instagram and Twitter candidate extraction.

A similar algorithm is used to obtain other information such as social media channels. Fig. 5 shows pseudo-codes for Instagram and Twitter channel extraction.

From Fig. 5, title T, snippet S and website URL are used as input. Then, the pattern-matching method searches both T and S for the celebrity name and keyword Ikey. If any are found in S or T, the pattern-matching method locates the candidate Instagram/Twitter account which matches the pattern. Snippet title is included because unlike date of birth, Instagram and Twitter accounts can be shown on the title, sometimes along with the name of its owner. Likewise, the pattern matching applies to the URL if the celebrity name is found in the snippet or title but not the candidate.

3.3 Feature extraction

From the previous section, numerous personal information candidates for each celebrity would be generated, and many candidates would not be correct. Thus, we propose that machine learning is used to identify and extract the correct candidates. Features used for candidate extraction in this work are support, confidence, lift, and conviction.

a) Support

Support is frequency of a celebrity's name and a candidate in a snippet is determined in relation to the entire database. Respective formulae are for support of celebrity name and candidate co-occurrence, support of only celebrity, and support of candidate respectively:

$$\mathrm{supp}\left(E_{\mathrm{actor}} \rightarrow (t_1 \ldots t_n)\right) = \frac{f\left(E_{\mathrm{actor}} \cap (t_1 \ldots t_n)\right)}{N} \quad (3.3.1)$$

$$\mathrm{supp}\left(E_{\mathrm{actor}}\right) = \frac{E_{\mathrm{actor}}}{N} \quad (3.3.2)$$

$$\mathrm{supp}\left(t_n\right) = \frac{t_n}{N} \quad (3.3.3)$$

Where $t_1 \ldots t_n$ means a set of Thai characters. N is the total number of snippets.

b) Confidence

Confidence is calculated by comparing the support of celebrity's name and information candidate to the support of celebrity's name.

$$\mathrm{conf}\left(E_{\mathrm{actor}} \rightarrow (t_1 \ldots t_n)\right) = \frac{\mathrm{support}\left(E_{\mathrm{actor}} \cap (t_1 \ldots t_n)\right)}{\mathrm{support}\left(E_{\mathrm{actor}}\right)} \quad (3.3.4)$$

c) Lift

Lift is calculated by getting support of celebrity to information candidate, and then support of celebrity and support of information candidate separately.

$$b = t_1 \ldots t_n$$

$$\mathrm{lift}\left(E_{\mathrm{actor}} \rightarrow (b)\right) = \frac{\mathrm{support}\left(E_{\mathrm{actor}} \cap (b)\right)}{\mathrm{support}\left(E_{\mathrm{actor}}\right)\mathrm{support}(b)} \quad (3.3.5)$$

d) Conviction

Conviction is comparison of support of celebrity name to the support of celebrity name without information candidate, and then support of information candidate is compared against support of information candidate without celebrity name.

$$\mathrm{conv}\left(E_{\mathrm{actor}} \rightarrow (t_1 \ldots t_n)\right) = \frac{\mathrm{support}\left(E_{\mathrm{actor}}\right)}{\mathrm{support}\left(E_{\mathrm{actor}} \cap \neg(t_1 \ldots t_n)\right)} \quad (3.3.6)$$

3.4 Selection of candidates by machine learning

Rapidminer is used to implement machine learning. In this section, five machine learning algorithms (Decision Tree, Random Forest, Support Vector Machine, k-NN, and Artificial Neural Network) are used. In order to boost accuracy, feature selection such as backward elimination, forward selection, bidirectional elimination, all-in, or combinations can be used. The reason backward elimination is selected is that the all-in approach is computationally expensive. The candidates are manually tagged to serve as a training set for this step.

4. Results and Discussion

4.1 Experiment result

In this part, results of experiments with the five algorithms, from web crawling to machine learning, will be shown below.

4.1.1 Web crawling

In this study, 317 celebrity names presented on MThai.com/starthai were used. Snippets were obtained by using Google API, using the celebrity names as keywords for searching. In total, 22,484 snippets were obtained, an average of around 70 snippets per celebrity. Additionally, a master student was hired to manually collect the correct information from the Internet as a human baseline.

4.1.2 Pattern matching and extraction

In this step, we use our candidate generation method to generate a list of candidates for processing. Numbers of extracted candidates are as follows: 458 for birthdays, 98 for height, 78 for heritage, 479 for Instagram, 71 for Twitter and 44,501 for film names.

4.1.3 Selection of candidates by machine learning

Parameter setup of the first algorithm-Decision Tree is shown in Table 2. This is the default setting given by Rapidminer without further adjustment.

Table 2. Decision tree setup.

Properties	Value
Criterion	Gain Ratio
Maximal depth	10
Confidence	0.1
Minimal gain	0.01
Minimal leaf size	2
Minimal size for split	4
Number of prepruning alternatives	3

For the second algorithm or Random Forest, settings are shown in Table 3.

Table 3. Random Forest setup.

Properties	Value
Number of trees	100
Criterion	accuracy
Maximal depth	10
Voting strategy	Confidence vote

The next algorithm is k-NN which setting is shown in Table 4:

Table 4. k-NN setup.

Properties	Value
K	5
Weighted vote	Yes
Measure types	MixedMeasures
Mixed measure	MixedEuclideanDistance

Setting for the SVM algorithm is specified in Table 5.

Table 5. SVM setup.

Properties	Value
Kernel type	Dot
Kernel Cache	200
C	0.0
Convergence epsilon	0.001
Max iterations	100000
L Pos	1.0
L Neg	1.0
Epsilon	0.0
Epsilon Plus	0.0
Epsilon Minus	0.0

Lastly, setting for the ANN algorithm is shown in Table 6.

The following tables will show results of backward elimination, along with recall and precision of our approach, categorized by type of information (birthday, height, heritage, Instagram and Twitter).

Table 6. ANN setup.

Properties	Value
Training cycles	200
Learning rate	0.01
Momentum	0.9
Decay	No
Shuffle	Yes
Normalize	Yes
Error Epsilon	1.0E-4
Use local random seed	No

In the tables, ActorSup means occurrence of celebrity name, SupportOcc means occurrence of the candidate, SupportCooc means occurrence of the candidate and celebrity name together, SupportActNOTCan means occurrence of celebrity name without the candidate.

Table 7. Features selected by backward elimination process for date of birth.

Name of features (DOB)	DT	RF	SVM	k-NN	ANN
ActorSup	Yes	Yes	Yes	Yes	Yes
SupportOcc	Yes	Yes	Yes	Yes	Yes
SupportCooc	No	No	Yes	Yes	Yes
SupportAct NOTCan	Yes	Yes	Yes	No	Yes
Confidence	Yes	Yes	No	Yes	No
Lift	No	Yes	Yes	No	No
Conviction	No	Yes	Yes	Yes	Yes

It can be seen in Table 7 that not all features are selected by backward elimination. The most-rejected feature is Lift, which is not used by Decision Tree, k-NN and ANN.

Precision and recall of date of birth from the five algorithms (with and without feature selection) are shown in Table 8.

Table 8. Recall and precision of birthday.

DOB	Recall	Precision	F1
Decision Tree	94.08	81.44	0.873
Decision Tree w/Feature	93.09	80.19	0.861
Random Forest	94.08	82	0.876
Random Forest w/Feature	94.83	83.33	0.887
SVM	99.34	71.43	0.831
SVM w/Feature	99.34	80	0.886
k-NN	91.78	74.49	0.822
k-NN w/Feature	91.12	76.32	0.830
ANN	92.76	78.64	0.851
ANN w/Feature	91.45	77.59	0.839

As seen in Table 8, the highest F1 score is Random forest with feature selection at 0.887.

Table 9. Features selected by backward elimination process for height.

Name of features (Height)	DT	RF	SVM	k-NN	ANN
ActorSup	No	No	No	Yes	No
SupportOcc	Yes	Yes	Yes	Yes	Yes
SupportCooc	Yes	No	Yes	No	Yes
SupportAct NOTCan	Yes	Yes	Yes	Yes	Yes
Confidence	Yes	No	Yes	Yes	Yes
Lift	Yes	Yes	Yes	Yes	Yes
Conviction	Yes	Yes	Yes	No	Yes

It can be seen in Table 9 that SupportOcc, SupportActNotCan and Lift are used by all five algorithms.

Table 10. Recall and precision of height.

Height	Recall	Precision	F1
Decision Tree Original	92.14	75.77	0.832
Decision Tree w/Feature	97.33	77.66	0.864
Random Forest	96.25	75.72	0.848
Random Forest w/feature	90.67	78.16	0.840
SVM	100	76.44	0.866
SVM w/Feature	100	76.53	0.867
k-NN	96.07	75.67	0.847
k-NN w/Feature	98.67	76.29	0.860
ANN	100	76.44	0.866
ANN w/Feature	100	76.53	0.867

In case of height as stated in Table 10, SVM and ANN algorithms have the highest F1 score. This is due to recall being 100%.

Table 11. Features selected by backward elimination process for heritage.

Name of features (heritage)	DT	RF	SVM	k-NN	ANN
ActorSup	No	No	Yes	Yes	No
SupportOcc	Yes	Yes	Yes	Yes	No
SupportCooc	Yes	Yes	Yes	Yes	No
SupportAct NOTCan	Yes	No	Yes	No	Yes
Confidence	No	No	No	Yes	Yes
Lift	Yes	Yes	Yes	No	No
Conviction	Yes	Yes	Yes	Yes	Yes

Table 11 shows that only conviction is selected by all algorithms.

Table 12. Recall and precision of heritage.

Heritage	Recall	Precision	F1
Decision Tree	66.33	67.21	0.668
Decision Tree w/Feature	66.67	71.79	0.691
Random Forest	70.83	74.88	0.728
Random Forest w/Feature	66.67	71.79	0.691
SVM	49.83	80.5	0.616
SVM w/Feature	59.52	83.33	0.694
k-NN	58.67	53.88	0.562
k-NN w/Feature	69.05	74.36	0.716
ANN	64.67	72.33	0.683
ANN w/Feature	61.9	83.87	0.712

According to Table 12, Random Forest gives the highest F1 score for heritage. It can be seen that with feature selection, k-NN and ANN significantly improve.

Table 13. Features selected by backward elimination process for Instagram.

Name of features (Instagram)	DT	RF	SVM	k-NN	ANN
ActorSup	Yes	Yes	Yes	No	Yes
SupportOcc	Yes	Yes	Yes	Yes	Yes
SupportCooc	Yes	Yes	Yes	Yes	Yes
SupportAct NOTCan	No	Yes	Yes	Yes	No
Confidence	No	Yes	No	No	Yes
Lift	No	Yes	Yes	Yes	Yes
Conviction	No	No	Yes	No	Yes

According to Table 13, SupportOcc and SupportCooc are selected by all the algorithms.

Table 14. Recall and precision of Instagram.

Instagram	Recall	Precision	F1
Decision Tree	79.15	79.67	0.794
Decision Tree w/Feature	85.71	80.73	0.831
Random Forest	91.48	79.09	0.848
Random Forest w/Feature	85.71	80.43	0.830
SVM	69.11	70.73	0.699
SVM w/Feature	73.75	73.46	0.736
k-NN	75.28	73.01	0.741
k-NN w/Feature	78.38	74.09	0.762
ANN	74.14	74.84	0.745
ANN w/Feature	71.04	77.31	0.740

Table 14 shows that Random Forest without feature selection is the best algorithm for Instagram prediction.

Table 15. Features selected by backward elimination process for Twitter.

Name of features (Twitter)	DT	RF	SVM	k-NN	ANN
ActorSup	Yes	No	No	Yes	No
SupportOcc	Yes	Yes	Yes	No	Yes
SupportCooc	Yes	Yes	Yes	Yes	Yes
SupportAct NOTCan	No	Yes	Yes	Yes	Yes
Confidence	No	Yes	Yes	No	Yes
Lift	Yes	Yes	No	Yes	Yes
Conviction	Yes	Yes	Yes	Yes	Yes

In Table 15 SupportCooc and Conviction are accepted by all the algorithms.

Table 16. Recall and precision for Twitter.

Twitter	Recall	Precision	F1
Decision Tree Original	53.33	59.26	0.561
Decision Tree w/Feature	53.33	80	0.640
Random Forest Original	53.33	69.17	0.602
Random Forest w/Feature	56.67	65.38	0.607
SVM Original	21.67	63.64	0.323
SVM w/Feature	26.67	80	0.400
k-NN Original	50	65.22	0.566
k-NN w/Feature	53.33	69.57	0.604
ANN Original	49.17	68.18	0.571
ANN w/Feature	43.22	72.22	0.541

For Twitter prediction, all algorithms show higher precision than recall. Decision Tree with feature selection likewise has higher F1 score than others.

In case of film names, our preliminary experiment showed that although 44,501 candidates are obtained for 315 celebrities, numbers of correct and incorrect candidates were extremely imbalanced and thus filtering was necessary. In this study, average co-occurrence between celebrity and film name candidates is approximately 12 and thus candidates with co-occurrence less than 12 will be filtered out. For comparison, results without filtering will be shown alongside results with filter value of 6 and 18 (50% and 150% of 12) in this study as well.

On film prediction, Table 17 shows that despite filtering efforts, the ratio between correct and incorrect candidates remains low and the ratio actually decreases with higher F-value.

Table 17. Comparison between correct and incorrect film candidates at different Filter values.

Filter Value	F=0	F=6	F=12	F=18
Total	44,501	15,316	9,020	6,073
Correct	2,294	274	104	40
Incorrect	42,207	15,042	8,916	5,969
Ratio	5.435%	1.821%	1.166%	0.670%

As with the previous candidate groups, five algorithms are used to get recall and precision, results of which would be compared to determine the best algorithm at different F-value.

Table 18. Recall and precision of all algorithms at F=0.

F=0	DF	RF	SVM	k-NN	ANN
Recall (%)	21.01	10.55	18.83	65.26	1.35
Precision (%)	88.12	96.41	100.0	71.97	31.96
F1	0.33	0.19	0.31	0.68	0.02

At F=0 k-NN has the highest recall while SVM has the highest precision. But k-NN has the highest F1 score due to high recall (65.26%) compared to SVM (18.83%).

Table 19. Recall and precision of all algorithms at F=6.

F=6	DT	RF	SVM	k-NN	ANN
Recall (%)	77.37	77.63	85.77	78.47	12.41
Precision (%)	75.71	86.13	99.16	78.47	49.28
F1	0.76	0.81	0.91	0.78	0.19

According to Table 19, SVM has the highest recall, precision and F1 value, signifying the best performance. SVM has the highest recall, precision and F1 value, signifying the best performance.

Table 20. Recall and precision of all algorithms at F=12.

F=12	DT	RF	SVM	k-NN	ANN
Recall (%)	0.0	89.42	94.23	76.92	15.38
Precision (%)	0.0	100.00	100.00	81.63	53.33
F1	0.00	0.94	0.97	0.79	0.23

In Table 20, SVM has the highest recall while having 10% precision, resulting in the highest F1 score. On the other hand, Decision Tree is unable to determine the result.

Table 21. Recall and precision of all algorithms at F=18.

F=18	DT	RF	SVM	k-NN	ANN
Recall (%)	87.50	92.50	0.0	72.50	2.50
Precision (%)	100.00	100.00	0.0	78.38	50.00
F1	0.93	0.96	0.00	0.75	0.04

At F=18, Random Forest algorithm has the highest F1 score while the ANN has the lowest F1 score.

4.2 Comparison with existing methods and discussions

4.2.1 Web crawling

Although the limit for crawling is 100, 76 celebrities had 50 or less snippets attributed to them due to their obscurity, or being older actors/actresses who have not acted for some time.

We compared our method to the method employed by Chainapaporn and Netisopakul [18] that used Lexto to tokenize Thai text from the web page before applying pattern matching. We found that Lexto was able to get date of birth and some country names (because they are already separated by whitespace characters). However, it was unable to tokenize out-of-vocabulary texts like abbreviations, personal names, film names or social network names, as seen in Fig. 6 This meant Lexto would not be useful for candidate extraction unless there is further tuning.

4.2.2 Pattern matching

After the experiment, we found that the largest group of candidates is film name (44,501) followed by Instagram (479) date of birth (458), height (98), heritage (78) and Twitter (71). Due to low candidate count in some groups, many celebrities have only one candidate although it does not guarantee correctness. Typos likewise affected candidate extraction, as missing or misplaced tonal marks or vowels are somewhat common, especially on personal blogs or social media posts that do not have proofreading.

<table>
<tr><td>สุนิตา ลีติกุล จรรยาธนากร (ชื่อเล่น:โบ) เกิดเมื่อวันที่ 16 สิงหาคม พ.ศ. 2518 เป็นนักร้องเพลงไทย โบเป็นลูกครึ่งไทยกับอัฟกานิสถาน คุณพ่อเป็นคนอัฟกานิสถาน และคุณแม่เป็น คนไทย และยังมีเชื้อสายจีน, มอญ และพม่า ด้วย หลังจากโบสอบเทียบจนจบชั้นมัธยมศึกษาปีที่ 6 ได้ตัดสินใจเดินถือเทปคาสเซ็ท ที่อัดเสียงร้องของเธอ ...</td><td>สุ | นิ | ตา | | ลี | ติ | กุล | | จรรยา | ธนา | กร | | (| ชื่อเล่น | : | โบ |) | | เกิด | เมื่อ | วันที่ | | 16 | | สิงหาคม | | พ.ศ. | | 2518 | | เป็น | นัก | | ร้องเพลง | ไทย | | โบ | เป็น | ลูกครึ่ง | ไทย | กับ | อัฟกานิสถาน | | คุณพ่อ | เป็น | คน | อัฟกานิสถาน | | และ | คุณแม่ | เป็น | | คนไทย | | และ | ยังมี | เชื้อสาย | จีน | , | | มอญ | | และ | พม่า | | ด้วย | | หลังจาก | โบ | สอบ | เทียบ | จน | จบ | ชั้น | มัธยมศึกษา | | ปี | ที่ | | 6 | | ได้ | ตัดสินใจ | เดิน | ถือ | เทป | คา | ส | เซ็ท | | ที่ | อัดเสียง | ร้อง | ของ | เธอ | | . | . | . |</td></tr>
<tr><td>ศรราม เทพพิทักษ์ ชื่อเล่น หนุ่ม เกิดเมื่อวันที่ 22 สิงหาคม พ.ศ. 2516 ที่กรุงเทพมหานคร เป็น นักแสดง นักร้อง ชาวไทย เป็น บุตรของ ชุมพร เทพพิทักษ์ นักแสดงอาวุโส โดยชื่อ ศรราม ตั้งมาจากชื่อของตัวเอกใน ภาพยนตร์เรื่อง หนึ่งนุช ...</td><td>ศร | ราม | | เทพ | พิทักษ์ | | ชื่อเล่น | | หนุ่ม | | เกิด | เมื่อ | วันที่ | | 22 | | สิงหาคม | | พ.ศ. | | 2516 | | ที่ | กรุงเทพมหานคร | | เป็น | นักแสดง | | นักร้อง | | ชาวไทย | | เป็น | บุตร | ของ | | ชุมพร | | เทพ | พิทักษ์ | | นักแสดง | อาวุโส | | โดย | ชื่อ | | ศร | ราม | | ตั้ง | มาจาก | ชื่อ | ของ | ตัวเอก | ใน | ภาพ | ยนตร์ | | เรื่อง | | หนึ่ง | นุ | ช | . | . | . |</td></tr>
</table>

Fig. 6. Example of Lexto tokenization.

On date of birth, most of the incorrect results were from misattribution as some celebrities did not have birthday information but due to the presence of other celebrities' birthdays (for example as a list of celebrities with their birthdays), they are attributed incorrect birthdays. Misattribution may happen when a celebrity has a name similar to another celebrity, or has a more famous sibling.

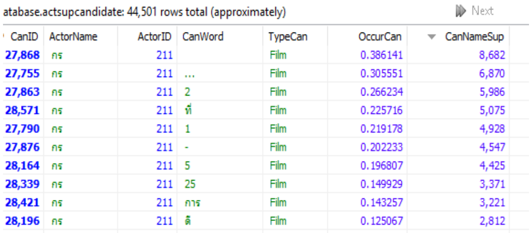
On height, misattribution occurs when a list of celebrities and their height appear on a single page; in addition, some celebrities have changed their names at some point in their careers and available height information is attributed to their old names.

On Instagram, generally pattern matching could accurately get the correct result, but sometimes fan Instagrams were selected due to celebrities' name being present in the snippet or title.

Twitter has high number of "ufo" accounts. Although official verification is available, only two Thai Twitter accounts were verified. Out of 144 Twitter accounts found, 31 were role-playing accounts. The word "ufo" came from a Korean website where fans could chat with Korean singers. Later, this role-playing trend spreads to Thai K-pop webboards and then Twitter.

Excluding noise, the highest support value of celebrity name to candidate film name usually signifies the best-known film associated with the celebrity in question.

Support of candidate film names is high for recent films and series, while older series have less than 10 occurrences. For example, "ตำรวจเหล็ก" a series from 2010 has only three occurrences in the 22,484 snippets compared to "ฮอร์โมน" that has 331. Some films such as "เพื่อนสนิท" (2005) have 94 occurrences due to it being synonymous to a Thai common word for "close friend".



atabase.actsupcandidate: 44,501 rows total (approximately)

CanID	ActorName	ActorID	CanWord	TypeCan	OccurCan	CanNameSup
27,868	กร	211		Film	0.386141	8,682
27,755	กร	211	...	Film	0.305551	6,870
27,863	กร	211	2	Film	0.266234	5,986
28,571	กร	211	ที่	Film	0.225716	5,075
27,790	กร	211	1	Film	0.219178	4,928
27,876	กร	211	-	Film	0.202233	4,547
28,164	กร	211	5	Film	0.196807	4,425
28,339	กร	211	25	Film	0.149929	3,371
28,421	กร	211	การ	Film	0.143257	3,221
28,196	กร	211	ดี	Film	0.125067	2,812

Fig. 7. Example of occurrence and support rates.

Similarly, cursory examination showed that the celebrity's name/candidate word pairs with low co-occurrence were whole phrases or sentences, while those with extremely high co-occurrence tended to be partial words, prepositions, or common words instead of specific names. Although for some long-running shows such as "ฮอร์โมน" co-occurrence tends to be high as it was one of the first, if not the first, series for the teenage cast and thus the cast were usually called by the nickname and "ฮอร์โมน" like "ก้อย ฮอร์โมน", much like the idol group BNK48.

On the other hand, valid titles' occurrence generally did not exceed 100, except some films that are synonymous with common words such as "นางร้าย" with 185 occurrences (appeared as a TV series in 2001 and 2018 on Channel 7). On the other hand, this word means the role of a female antagonist/rival many Thai actresses are associated with or best-known for, or "ฮอร์โมน" (509 occurrences), a long-running series from 2013-2015, which is synonymous with a common Thai borrowed word for hormone.

As in the previous step, we compared our method to the method employed by Chainapaporn and Netisopakul [18]. This method used keywords and regular expressions to extract information and is similar to our own method.

4.2.3 Selection of candidates by machine learning

It is found from the experiment that basic information such as date of birth and height largely have F1 score more than 0.8 due to high recall and precision. High recall and precision of such information are likely because of predictable writing patterns.

Recall and precision of Instagram extraction range between 0.74 to 0.84, more than Twitter for which F1 score of all five algorithms (with and without feature selection) never exceed 0.65. This could be attributed to more widespread use of Instagram.

Heritage extraction has F1 score between 0.5-0.7 as some celebrities are misattributed heritage. In some cases, the only candidate attributed to the celebrity is deemed wrong due to typos (including the use of adjective instead of noun like "อิตาเลี่ยน" instead of "อิตาลี") or inclusion of irrelevant words after nationality words. Furthermore, heritage information can be written in more than two ethnicities which resulted in no candidate being extracted. Feature selection does improve the F1 score somewhat, except in the case of Random Forest.

On film extraction, we found that as we applied more stringent filters, precision and recall percentages increase except when F=18, which resulted in lowered precision except for Decision Tree and Random Forest. This could be attributed to the lower number of available entries for the input. On the other hand, at F=18, all machine learning techniques showed improved F1 score except SVM, k-NN and ANN.

The most balanced filter value between number of filtered candidate words and recall/precision was 6, although at this value, real film names that have low co-occurrence with the celebrities such as "เพชรตัดเพชร", which has 16 total occurrences but has only one co-occurrence with each of its celebrities, are left out. Only one notable snippet related to the 2016 version was acquired and not the earlier 1966, 1984, and 2001 versions.

4.2.4 Comparison with existing methods and MThai.com

The following part is a comparison of our information extraction method to manually collected information on MThai.com, and the method used by Chainapaporn and Netisopakul [18]. Performance measure is whether the approach can get the same information as a set of manually collected information provided by the master student we hired. In this case we accept 95% correctness as correct, such as abbreviated form of the month in birthday, but otherwise one missing character is enough for the candidate to be classified as incorrect.

When compared to the method used by Chainapaporn and Netisopakul [18], our method treats extracted words only as "candidates" that need further vetting by machine learning process, while the method used by Chainapaporn and Netisopakul compared the extracted words directly.

We compared our approach's extraction capabilities with MThai.com's manual approach and Chainapaporn and Netisopakul's approach. It is found that our method is superior to both methods as shown in Fig. 8. It is possible that use of candidate approach and machine learning helps with extraction.

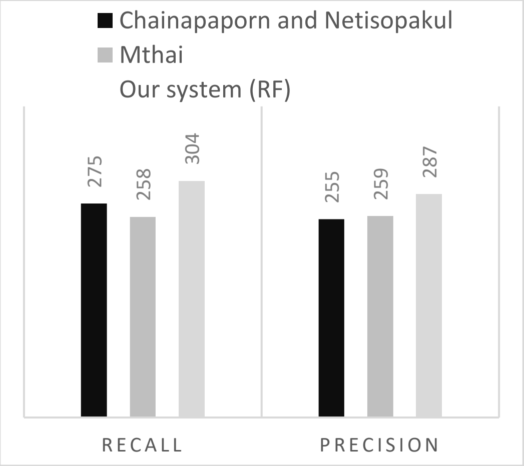


Fig. 8. Birthday recall and precision of our approach, compared to MThai.com and Chainapaporn and Netisopakul's method.

Regarding height information extraction, Fig. 9 shows performance of our approach compared with the method used by Chainapaporn and Netisopakul and MThai. It's found that our approach performs much better.

Regarding heritage, it is found that our pattern matching mostly extracted correct information except in case of popularlymistaken heritage (Praiya Suandorkmai and Nadech Kugimiya) as mentioned.

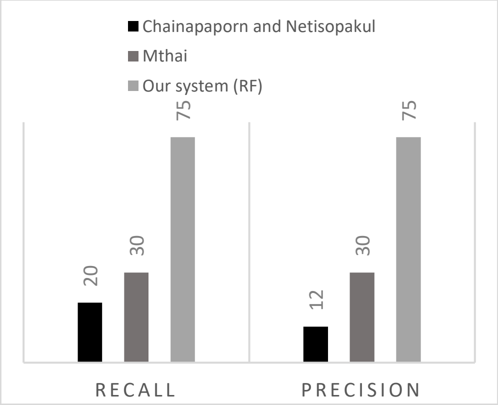


Fig. 9. Height recall and precision of our approach, compared to MThai.com and Chainapaporn and Netisopakul's method.

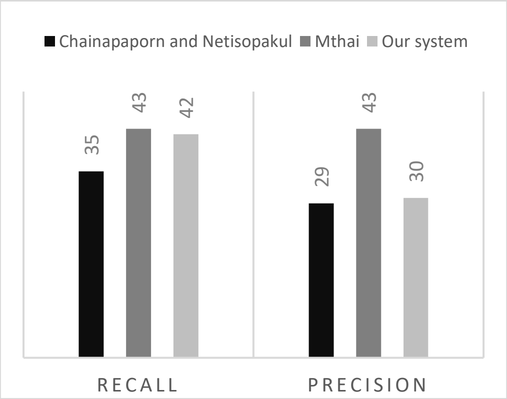


Fig. 10. Heritage recall and precision of our approach, compared to MThai.com and Chainapaporn and Netisopakul's method.

Regarding social media information, MThai is superior to our approach as shown in Fig. 11. It is possible that manually-collected information allows better cross-validation than a machine learning-based approach, especially against ufo/fan accounts. Still, use of machine learning algorithms apparently improves performance of the information extraction process.

As MThai.com collects information using its staff, it has comparable precision to our approach. However, it is possible that sheer number of celebrities means not all celebrities in MThai.com have all information.

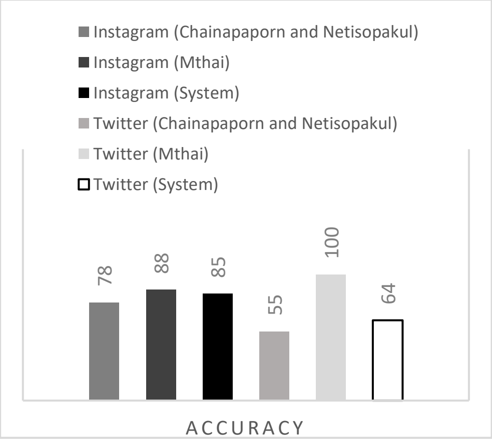


Fig. 11. Comparison of social media information extraction accuracy.

5. Conclusion

Our personal information extraction approach is able to extract information from structured and unstructured web snippets. Some information such as birthday, heritage and height can be readily extracted due to their limited, predictable forms that appear in web snippets. Any conflicted information in this part could be partially solved by machine learning. Our proposed approach has higher recall compared to a major website like MThai.com in most areas except heritage. Furthermore, when compared to other state-of-the-art methods, our approach consisting of pattern matching method and machine learning method has higher recall and precision compared to the tokenizer and rules-based approach used in previous works, as in approximately 10% better in date of birth, over twice as effective in height, 20% better in heritage, 8% in Instagram and 16% in Twitter.

However, some issues still need to be resolved. Social media account extraction is difficult not from inability to extract the account name, but verification and attribution of the account to the correct celebrity. As Thai celebrity accounts have low verification rate, impersonating or "for fun" similarly-named accounts are in significant numbers and difficult to actually validate.

Experiments showed that hand-crafting regular expression rules might not be adequate for a relatively large text and some forms of machine learning-based rule generation must be introduced.

Film name extraction results showed huge discrepancy in recall and precision as our extracted information contained a significant degree of noise, although if the information was excessively filtered, much of valid information with low co-occurrence would be excluded.

In future works, we plan to find other information such as relationship (i.e. spouse, sibling, parent, child) between celebrities, films and crew to better connect them together in the future.

Acknowledgments

The first and second authors contribute 50% equally to this work. In addition, the authors would like to express our thanks to Dr. Nattakarn Phaphoom for English grammar guidance.

References

[1] Alexa. mthai.com Competitive Analysis, Marketing Mix and Traffic - Alexa [Internet]. 2014 [cited 2020 12 June]. Available from: https://www.alexa.com/siteinfo/mthai.com.

[2] Alexa. thairath.com Competitive Analysis, Marketing Mix and Traffic - Alexa [Internet]. 2020 [cited 2020 15 May]. Available from: https://www.alexa.com/siteinfo/thairath.co.th.

[3] Noordin MF, Othman R, editors. An information retrieval system for Quranic texts: a proposed system design. 2006 2nd International Conference on Information & Communication Technologies; 2006: IEEE.

[4] Kopparapu SK, editor Automatic extraction of usable information from unstructured resumes to aid search. 2010 IEEE International Conference on Progress in Informatics and Computing; 2010: IEEE.

[5] Chen J, Gao L, Tang Z. Information extraction from resume documents in pdf format. Electronic Imaging. 2016;2016(17):1-8.

[6] Chen J, Zhang C, Niu Z. A Two-Step Resume Information Extraction Algorithm. Mathematical Problems in Engineering. 2018;2018:1-8.

[7] Liu X, Chen H, editors. AZDrugMiner: an information extraction system for mining patient-reported adverse drug events in online patient forums. International conference on smart health; 2013: Springer.

[8] Elsebai A, Meziane F, Belkredim FZ. A rule based persons names Arabic extraction system. Communications of the IBIMA. 2009;11(6):53-9.

[9] Sharma SS, De Choudhury M. Measuring and Characterizing Nutritional Information of Food and Ingestion Content in Instagram. Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion2015. p. 115-6.

[10] Imran M, Elbassuoni S, Castillo C, Diaz F, Meier P, editors. Practical extraction of disaster-relevant information from social media. Proceedings of the 22nd International Conference on World Wide Web; 2013.

[11] Chen Y, Lee SYM, Huang C-R. A robust web personal name information extraction system. Expert Systems with Applications. 2012;39(3):2690-9.

[12] Cheng TT, Cua JL, Tan MD, Yao KG, Roxas RE, editors. Information extraction from legal documents. 2009 Eighth International Symposium on

Natural Language Processing; 2009: IEEE.

[13] Freitag D, McCallum A, editors. Information extraction with HMMs and shrinkage. Proceedings of the AAAI-99 workshop on machine learning for information extraction; 1999: Orlando, Florida.

[14] Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos CD, Stamatopoulos P. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. arXiv preprint cs/0009009. 2000.

[15] Emami H, Shirazi H, Abdollahzadeh A, Hourali M. A pattern-matching method for extracting personal information in farsi content. UPB Sci Bull, Ser C. 2016;78(1):125-38.

[16] Aboaoga M, Ab Aziz MJ. Arabic person names recognition by using a rule based approach. Journal of Computer Science. 2013;9(7):922.

[17] Qu J, Lu Y, editors. Automatic identification and multi-translatable translation of vocabulary terms with a combined approach. 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI); 2016: IEEE.

[18] Chainapaporn P, Netisopakul P, editors. Thai herb information extraction from multiple websites. Knowledge and Smart Technology (KST); 2012: IEEE.

[19] Wanichayapong N, Pruthipunyaskul W, Pattara-Atikom W, Chaovalit P, editors. Social-based traffic information extraction and classification. 2011 11th International Conference on ITS Telecommunications; 2011: IEEE.

[20] Gossen G, Demidova E, Risse T, editors. iCrawl: Improving the freshness of web collections by integrating social web and focused web crawling. Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries; 2015.

[21] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of machine learning research. 2003;3(Mar):1157-82.

[22] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning: Springer; 2013.

[23] Qu J, Theeramunkong T, Le Ming N, Shimazu A, Nattee C, Aimmanee P. A Flexible Rules-based Approach to Learn Medical English-Chinese OOV Term Translations from the Web. International Journal of Computer Processing Of Languages. 2012;24(02):207-36.

[24] Qu J, Nguyen LM, Shimazu A. Cross-language information extraction and auto evaluation for OOV term translations. China Communications. 2016;13(12):277-96.

[25] Qu J, Theeramunkong T, Nattee C, editors. A novel candidate generation technique for web based English-Chinese medical OOV term translation. Proceedings of the International Conference on Knowledge, Information and Creativity Support Systems; 2009.